NTNU – Trondheim
Norwegian University of
Science and Technology

Membrane processes relevant for the polymer electrolyte fuel cell

Aleksander Kolstad

Chemical Engineering and Biotechnology
Submission date: June 2013
Supervisor: Signe Kjelstrup, IKJ

Norwegian University of Science and Technology
Department of Chemistry

Preface

This master thesis has been carried out at The Department of Chemistry at The Norwegian University of Science and Technology. The important aspects concerning the Polymer Electrolyte Membrane Fuel Cell, more commonly known as Proton Exchange Membrane Fuel Cell (PEMFC), have been studied in two separate parts.

Part 1 of the thesis is an introduction about PEMCF and hydrogen technology, as well as how this can be used in the human society, which is given before the in depth study begins. Each part covers essential study that can further help understanding and enhance the fuel cell technology. A simulation model has been established for each part, where part 2 is a combination of experimental results and validation of the simulation model. Part 3 is more in depth study of the Proton Exchange Membrane Fuel Cell by use of the simulation model.

As far as I know this is the first study with finding the absorption enthalpy between liquid water and Sigracet layers, along with establishing a more robust model for simulating the Proton Exchange Membrane Fuel Cell, based on non-equilibrium thermodynamics, compared to previously simulation models.

I would like to thank my supervisor Signe Kjelstrup for outstanding help and encouragement in the process of writing this thesis. I would like to thank Odne S. Burheim for giving valuable advices and general help, along with being helpful enough to read through an earlier edition and help sorting out errors along the way. I would also like to thank Bjørnar Holsetstuen and Oskari Oksanen for proofreading my thesis. Thank you for your feedback and support.

Abstract

Research in the Proton Exchange Membrane Fuel Cell (PEMFC) is important to get an efficient fuel cell that can be used as an energy carrier, for example in the transport sector. Understanding the different phenomena and variations in temperature, heat and other quantities is critical. Non-equilibrium thermodynamics is used to establish a 1-dimensional model for transport processes in a Nafion membrane system consisting of heat and mass transport, and for a PEM fuel cell with heat and transport of mass and charge.

The Nafion membrane in part 2 is coated with a Sigracet layer of either GDL10AA without Teflon or with GDL10BA with 5 % Teflon. Outside of these layers is liquid water. The absorption enthalpy between liquid water and the Sigracet layer has been found by combining experimental data with the established simulation model. For GDL10AA without Teflon this absorption enthalpy ranges from -460 J/mol to -3380 J/mol for mean temperatures of 30 ^{o}C and 75 ^{o}C respectively. For GDL10BA with Teflon this absorption enthalpy ranges from 1150 J/mol to 7850 J/mol for mean temperatures of 30 ^{o}C and 75 ^{o}C respectively. The heat capacity value of water, c_p^s, for Sigracet GDL10AA and GDL10BA was found to be 10 J/K mol and 223 J/K mol respectively. The effect on the absorption enthalpy and the sign and value of the water flux by changing the temperature and material properties is studied. This study has found that the heat conductivities play a minor role when it comes to transport of water compared to the diffusion constant of the Nafion membrane and the Sigracet layers.

A simulation model is established for the PEM Fuel Cell in part 3. Only variations in quantities along one dimension is considered. Non-equilibrium thermodynamics is used to properly describe heat and transport of mass and charge. The system has a Nafion membrane coated with a Sigracet layer of GDL10AA without Teflon at both ends. Outside of these layers are water vapor with hydrogen at one side and oxygen at the other. Case studies such as the reversible limit is studied in detail to confirm the accuracy and validity of the simulation model. Profiles of temperature, chemical composition, water content, measurable heat flux, electrical potential and entropy production are found by use of the simulation model for various current densities. A polarization curve by plotting the cell potential for different current densities is found. Additionally study and a sensitivity analysis for the PEM fuel cell are carried out to fully understand transport processes and the effects from material properties.

Sammendrag

Forskning innenfor Polymer Elektrolytt Membrane brenselcelle (PEMFC) er viktig for å få en effektiv brenselcelle som kan brukes som en energibærer, som for eksempel i transportsektoren. Forsåelse for de ulike fenomenene og variasjonene i temperatur, varme og andre egenskaper er kritisk. Ikke-likevekts termodynamikk er brukt for å etablere en 1-dimensjonal modell for transportprosesser i et Nafionmembran system bestående av varme og massetransport, og for en PEM brenselcelle med varme og transport av masse og ladning.

Nafionmembranen i del 2 er dekket med et Sigracetlag på begge sider av typen GDL10AA uten Teflon eller GDL10BA med 5 % Teflon. Vann i væskefase finnes på begge sider utenfor disse lagene. Absorbsjonsentalpien mellom flytende vann og Sigracetlagene har blitt funnet ved hjelp eksperimentelle data og den etablerte simuleringsmodellen. For Sigracet GDL10AA uten Teflon så ble denne absorbsjonstentalpien funnet til å ligge fra - 460 J/mol til -3380 J/mol for middeltemperaturer 30 ^{o}C and 75 ^{o}C respektivt. For Sigracet GDL10BA med Teflon så ble denne absorbsjonstentalpien funnet til å ligge fra 1150 J/mol til 7850 J/mol for middeltemperaturer 30 ^{o}C and 75 ^{o}C respektivt. Effekten på absorbsjonsentalpien og fortegn og verdi for vannfluksen ved forandring i temperatur og materielle egenskaper er studert i detalje.

En simuleringsmodell for PEMFC i del 3 er etablert. Kun variasjoner i en dimensjon er tatt med. Ikke-likevekts termodyamikk er brukt for å beskrive varme og transport av masse og ladning. Dette systemet har en Nafionmembran dekket med Sigracetlag av typen GDL10AA uten Teflon på begge sider. Utenfor disse lagene er det vanndamp sammen med hydrogen på den ene siden og oksygen på den andre siden. Det reversible tilfellet når strømtettheten går mot null er studert i detalje for å sammenligne med kjente verdier og erfaring for en brenselcelle ved reversible betingelser. Dette styrker modellen og kan si noe om hvor bra treffsikkerhet resultatene har. Temperatur, kjemisk sammensetning, vanninnhold, målbar varmefluks, elektrisk potensial og entropiproduksjon er funnet ved hjelp av simuleringsmodellen for varierende strømtettheter. En polarisasjonskurve ved å plotte cellepotensialet for ulike strømtettherer er funnet. I tillegg er en sensitivitetstest gjort for PEMFC for å videre forstå transportprosessene og effekter fra materielle egenskaper.

Contents

1 Part 1 - PEM Fuel Cell: The energetic, environmental and sustainable society and the challenges it represents

Alternative energy sources are sought to replace the traditional energy sources in the society today. Not only because of environmental aspects, but also due to todays energy sources limit. Oil, coal and natural gas are some of the main energy sources that we are using today. According to a study of Beretta[1] 77.8 % of the primary energy consumption came from fossil fuel in 2004 (32.8 % oil , 21.1 % natural gas, 24.1 % coal). The study further emphasize how the global energy consumption has doubled the last three decades, and how the demand for energy is still increasing rapidly. This can be viewed in Figure 1.0.1 which shows the worlds energy consumption from 1820 up to 2012.

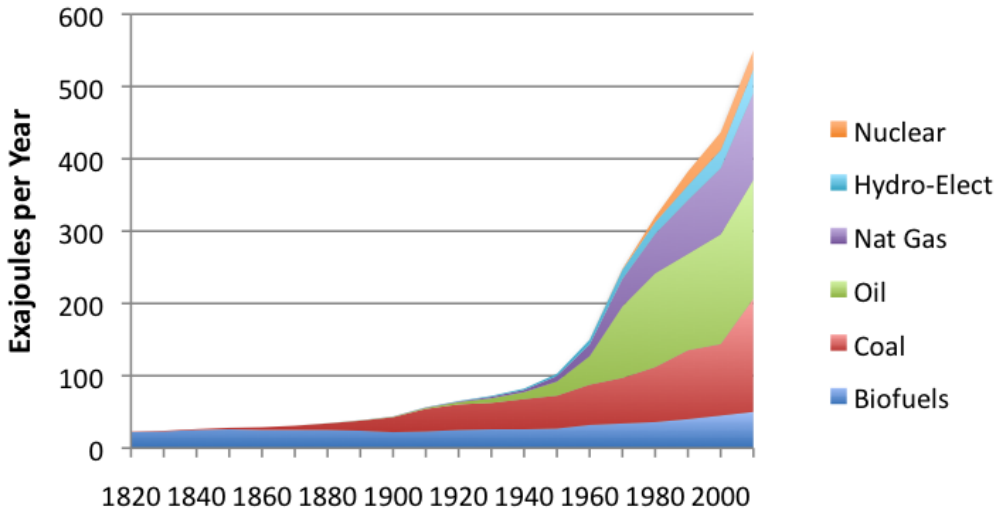

Figure 1.0.1: *Worlds energy consumption from 1820 up to 2012[2].*

Fossil fuel sources will not always be available for the human society, as the reservoirs will be depleted at some point or made inaccessible. It is necessary to develop new ways of replacing these energy sources to get an energetic, environmental and sustainable society. Continuing using fossil fuel can some day destroy the society as we it know today, if the trend of increased energy consumption of fossil fuel is followed. A study by

Dincer[3] shows how the fuel cell technology and hydrogen can be one possible solution to these issues. His work highlights the importance of the topic and show how fuel cell technology can help achieve a better environment and sustainability. However, Dincer emphasize an important aspect, namely that hydrogen cannot simply be produced from a mine or a well. Considerable energy is consumed in the extraction process, which means that hydrogen should properly be considered an energy carrier rather than an energy source according to Dincer. An interesting solution to this is to use more renewable energy sources, such as solar, wind, hydro, tides, waves and geothermal heat. The sun is the source of many of these renewable energy sources and the sun alone produce 10 000 times more energy[4] than what the society needs today. In other words, we do not have an energy crisis, but rather a logistic problem with respect to having the right energy form in right content and time. The renewable energy sources have to be considered unstable or highly dependent on the surroundings when compared to energy sources such as fossil fuel. For instance, the power delivered from a solar power plant varies in an uncontrollable fashion during day time and does not work during night time. This means that the society can not depend on these energy sources at all times due to these limitations. However, if renewable energy is combined with the technology of hydrogen and fuel cells, then we reach an interesting solution. Renewable energy sources can be used in the extraction process of hydrogen, where hydrogen can be stored and considered as the energy carrier in respect to the study of Dincer[3]. Stored hydrogen can be considered as a fuel in the same way fossil fuel is, and it has many possible applications, for example in the transport sector. 50 % of the oil today is used in the transport sector[4], and hydrogen technology should be considered as a good alternative.

Figure 1.0.2: *Picture of a wind farm. Wind is one of the many renewable energy sources that is a possibility of being used in the extraction process of hydrogen. Illustration taken from Green Living , Alternative Energy: Wind Electricity[5].*

A fuel cell is a device that converts the chemical energy from fuel into electricity through a chemical reaction with oxygen or another oxidizing agent. The concept of the fuel cell has been known since early 19th century[6]. However, it is in the recent years that the fuel cell has received more attention with respect to research and the important aspect with regards to alternative energy sources. Among the types of fuel cells the Polymer Electrolye Membrane Fuel Cell, more commonly known as Proton Exchange Membrane Fuel Cell (PEMFC) or PEM Fuel Cell, is receiving most attention for automotive and small stationary applications. PEM fuel cells convert the chemical energy from hydrogen into electricity through a chemical reaction with oxygen. The processes of mass and heat transfer, electrochemical reactions, and ionic and electronic transport are all important to get a full understanding of the fuel cell. Knowledge of these transport phenomena and processes are essential for getting a high efficient working fuel cell with the correct operating temperatures, fuel amount, efficient heat exchange and other relevant system settings or properties.

Research on new types of the PEM fuel cell is currently in the spotlight. An example is the newly developed High Temperature PEM Fuel Cell by Global Energy Innovation (GBI)[7], see Figure 1.0.3. This PEM fuel cell has an operating temperature of 160 ^{o}C

to 180 ^{o}C, which is higher than the regular fuel cell that traditionally operates at 70 ^{o}C to 80 ^{o}C. GBI claims that high temperature PEM fuel cells are more cost-effective, efficient and reliable than conventional low temperature fuel cell systems. Simulation modeling is one option to see what possibilities and limits that exists for the PEM fuel cell, along with how to obtain the efficiency that is required for the fuel cell to be viable as an energy carrier.

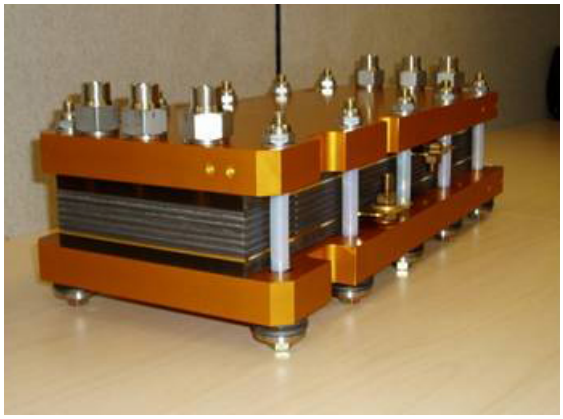

Figure 1.0.3: *Example of a High Temperature PEM Fuel Cell developed by Global Energy Innovation, the illustration is taken from the open website of GBI[7].*

Several papers have been published with simulation models for the PEM fuel cell. Weber and Newman[8] explain various methods for simulation and modeling of the PEM fuel cell along with the different types of transport. Kjelstrup and Røsjorde[9] and Vie et al.[10] show how non-equilibrium can be used to efficiently simulate the fuel cell, and extract essential information about temperature, mole fraction, heat flux and electrical potential variations in the cell. It is important that processes such as the surface reactions taking place in the electrodes and water transport through a membrane are well understood, so that the model is robust and able to fully reflect a working fuel cell. Knowledge about how to construct an efficient fuel cell can be withdrawn from good simulation models with accurate material properties, as well as in depth knowledge about the behavior of water, oxygen and hydrogen fluxes and other transport phenomena in the fuel cell. In depth studies and research regarding the PEM fuel cell, ranging from the basic electrochemical reactions to the more complex surface adsorption processes

and similar phenomena, are the key essences when it comes to the future of an efficient PEM fuel cell. The work of Bautista-Rodríuez et al.[11] highlights the limits for reaching a high efficient PEM fuel cell as mass transport and kinetic phenomena. According to them these are: 1) electrochemical kinetics, where the oxygen reduction reaction is the main problem, 2) mass transport phenomena such as transport (conduction) and diffusion of chemical species, species dissolution and conductivity of ionic species and electron transport, 3) as well as the product management to the outside of the cell. It is especially emphasized that water management gets difficult at higher current densities and it is highlighted as the main cause of diminished functionality in the PEM fuel cell at these conditions. This originates from transport barriers, which causes the liquid water trapped in the pores of electrodes. These transport barriers cause dead spots on the active area of the electrodes by reducing the effective reaction area, thus lowering the efficiency.

Temperature control, material aging and water management are some of the main topics that represent the challenges in todays fuel cell society. Scientific work on these parts would prove useful to further enhance existing and upcoming simulation models. Several papers[12,13,14] have studied and highlighted the important aspect about non-isothermal PEM fuel cells. Similar study with regards to the non-isothermal model is done with respect to heat effects and the contribution from these effects[15,16]. Knowledge about the temperature distribution and effects in the PEM fuel cell is both important for the performance due to kinetics and due to degradation rates which are exacerbated by high temperatures. The durability of the PEM fuel cell is related to efficiency and economical costs, and material aging is therefore an important aspect that needs considerations. The study from Burheim et al.[17] show the importance and effects material aging represents. Water management is widely known to be essential for cell performance and efficiency. The work of Bautista-Rodríuez et al.[11], as mentioned earlier, claims this to be one of the most important aspects that needs attention in todays scientific work. Other studies[18,19] agree on the importance of water management and the effects it has on the PEM fuel cell and the efficiency. They address the key challenges as "dry-out" and "flooding". "Dryout" is caused by membrane dehydration and occurs mainly on the anode side and causes low proton conductivity. "Flooding" happens when there is excessive water generated by the electrode reaction, and it is rapidly condensed in

the cathode electrode. When this occurs, the open pores in the catalyst layer and gas diffusion layer are filled with liquid water, and oxygen cannot be supplied to the reaction sites. These two phenomena are critical barriers for high efficiency and power density.

The proton exchange membrane fuel cell is still in the early years and there are much work to be done before we can hope to replace fossil fuel with renewable energy sources and hydrogen as an energy carrier. As Dincer[3] highlighted in his study, we face another challenge when it comes to the extraction process of hydrogen, which is an expensive procedure with regards to both economy and energy considerations. The infrastructures in the world would also have to adapt to using hydrogen as a fuel instead of relying on oil and gas, which will take time. All these features are expensive and it is therefore important that knowledge about the PEM fuel cell is studied in depth to get cheaper and more efficient devices. A hope for the future is to achieve these goals to obtain an energetic, environmental and sustainable society based on scientific work behind simulation models and experimental results. One possible use of the PEM fuel cell is in a car, which replaces the traditional gasoline motor, see Figure 1.0.4.

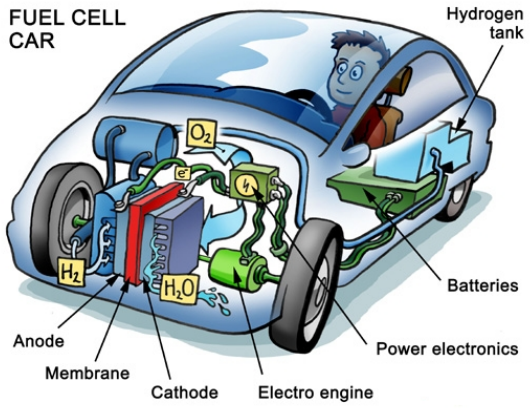

Figure 1.0.4: *A car is one of many possible applications for the proton exchange membrane fuel cell. A hydrogen tank is required as the fuel instead of traditional gasoline. The benefits of using hydrogen instead of gasoline are many, but the most obvious one is perhaps the lack of climate gases and other gases that affect the environment. Water is the only waste product being produced by the PEM fuel cell. The illustration is taken from Wikipedias page on fuel cells[20]*

This paper is dedicated to focus on some of the main topics that represents the challenges concerning the proton exchange membrane (PEM) fuel cell. This work has been divided into two separate parts, where each part deals with different aspects with regards to the PEM fuel cell. The first part studies water transport through the Nafion membrane and how the material properties of Sigracet GDL layers, which are porous transport layers, are affecting the water flux. This is done in a simulation model where variations is limited to exist in one dimension and transport of charge is not included. This work is done to find out how the Sigracet layers can affect the water transport as well as to establish the difference in enthalpy between liquid water and Sigracet. This is important for understanding how transport of water in the PEM fuel cell can be modeled with respect to Sigracet layers and how the water flux can be affected. The second part is about establishing a model for the PEM fuel cell for variations in one dimension, such as temperature, mole fraction, heat flux, electrical potential and entropy production. Sigracet layers of GDL10AA type are used as the anode and cathode electrodes, and in depth study with respect to the properties of these layers and the Nafion membrane is done. This work is important because of the previously mentioned aspects with regards to temperature control, water management and details with regards to heat fluxes.

2 Part 2 - Membrane system with heat and transport of mass

2.1 Introduction

Theory of irreversible thermodynamics is used to properly describe the fluxes and forces in a proton exchange membrane fuel cell system, as the system is not in equilibrium. In the PEM fuel cell there are three fluxes: heat, mass and charge. This results in a system with three main driving forces and three coupled forces. It is consequently useful to first study a simplified system with only heat and mass flux. The system is also reduced to variations in one dimension to further simplify the equations and the calculations.

To optimize the efficiency of a fuel cell, it is important to be have knowledge about water management and transport of water through the membrane and porous transport layers. This part focuses on water transport through a Nafion membrane coated with Sigracet layers, and how material properties of these layers can affect the transport of water. Experiments have been done with such a membrane system, consisting of a Nafion membrane surrounded by Sigracet layers and a water bath at each side of the layers. This work is done by Akyalcin[21], and personal communication with him has been done to provide the data and information about the material properties and experimental values. The water flux in this system has been measured with a temperature difference as the only driving force. The experiments have been done with a temperature difference of 3 ^{o}C , 5 ^{o}C and 10 ^{o}C and with a mean temperature in the system ranging from 30 ^{o}C to 75 ^{o}C. The results from the experiments show a shift in the sign of the water flux when different Sigracet layers were used. Sigracet GDL10AA without teflon gave a negative water flux, while Sigracet GDL10BA with 5 % Teflon gave a positive water flux.

Water is being transported as a result of a temperature gradient, where the sign of the flux appears to be dependent on the Sigracet layers material properties, which can be observed from the experimental results done by Akyalcin[21]. The experiments have a positive water flux for Sigracet GDL10BA and a negative water flux for Sigracet GDL10AA. It is therefore of interest to further study the difference in material properties of these Sigracet layers with respect to the water flux. The aim of this study is to establish a model that can describe these experimental observations based on a combi-

nation of equations from irreversible thermodynamics and data from the experimental results. Based on the work of Glavatskiy et al.[22,23] this behavior can be simulated using non-equilibrium thermodynamics. The final aim of this paper is to use the model to find the unknown absorption enthalpies going from liquid water into the Sigracet layer with varying mean temperatures inside the membrane cell system and type of Sigracet layer.

Some of the material properties of the Nafion membrane, and both the types of Sigracet layers, are known from earlier research done by Burheim et al. [24,25]. The heat conductivity is the main result from these papers, and these results are used in the simulations. The mass diffusion constant for water inside the Nafion membrane is known from Villaluenga et al.[26], while the diffusivity constant for water inside Sigracet is estimated by use of the diffusion constant for water.

This part of the thesis will initially give a theoretical description of the fluxes in a membrane system, with equations for fluxes and forces from theory of irreversible thermodynamics. The equations required to describe an inhomogeneous system were recently developed by Kjelstup et. al.[27,28] and give more complicated, but also more robust, expressions for the heat and mass fluxes.

2.2 Theory

The theoretical approach in this section is built on the non-equilibrium thermodynamics of Kjelstrup et al.[27,28] and the procedure of Glaviatsky et al.[22,23] unless stated otherwise.

The transport of water with concentration c, in mol/m^3 dry membrane, through a homogeneous membrane can be described by the total heat flux J_q and the mass flux $J_m = cv$ where v is the velocity of water. The energy fluxes satisfy the conservation equations $\partial c/\partial t = -\nabla J_m$ and $\partial ce/\partial t = -\nabla J_q$ where e is the total energy per mol. Since the fluxes in the system are very small, there is little kinetic energy and the total energy is approximately equal to the internal energy u. For the stationary state, J_m and J_q are constant throughout the system. In the case of experiments, the total heat flux J_q is not measured directly. Instead the measurable heat flux is being used, $J'_q = J_q - HJ_m$, where H is the partial molar enthalpy of the moving content.

The complete non-equilibrium thermodynamic description requires the expression for the entropy produced in a non-equilibrium system per unit of time in a given volume, i.e. the entropy production σ. The entropy production is given by Equation (2.1) and the linear flux-force relations for the measurable heat flux and the mass flux are given for the membrane in Equations (2.2) and (2.3). This entropy production refers to an experimental setup in a system where the measurable heat flux, J'_q , is used. However, as seen later in the calculations part, the total heat flux is used in an equivalent representation.

$$\sigma = J'_q \frac{\partial}{\partial x}\left(\frac{1}{T}\right) + J_m\left(-\frac{1}{T}\frac{\partial \mu_{m,T}}{\partial x}\right) \tag{2.1}$$

$$J'_q = l_{qq}\frac{\partial}{\partial x}\left(\frac{1}{T}\right) + l_{q\mu}\left(-\frac{1}{T}\frac{\partial}{\partial x}\mu_T\right) \tag{2.2}$$

$$J_m = l_{\mu q}\frac{\partial}{\partial x}\left(\frac{1}{T}\right) + l_{\mu\mu}\left(-\frac{1}{T}\frac{\partial}{\partial x}\mu_T\right) \tag{2.3}$$

A heterogeneous system, as shown in Figure 2.2.1, consists of one Nafion membrane with two Sigracet layers. The a and c part of the system is a water bath where the temperature is constant equal to T_a and T_c. The temperature in part a is always higher

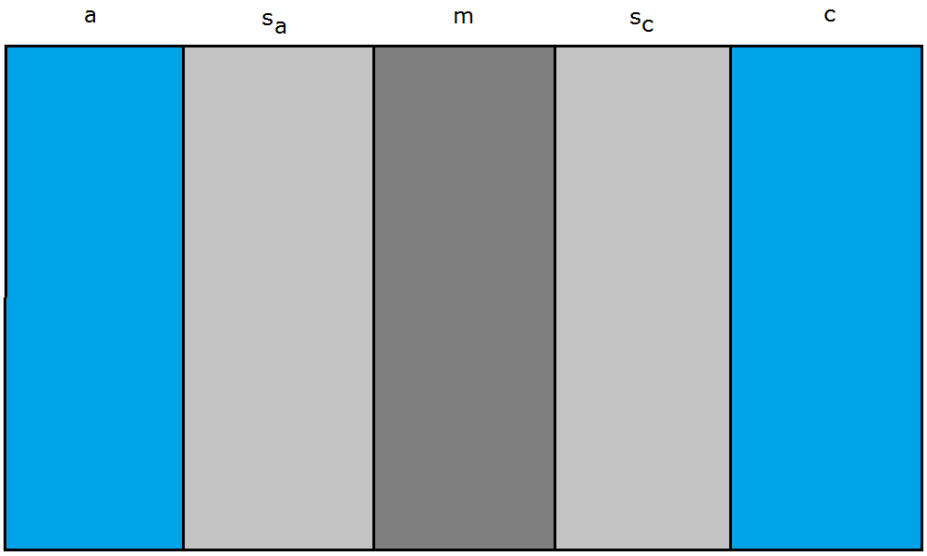

Figure 2.2.1: *A sketch of the system where the different regions in the system are divided into a, s_a, m, s_c and c. The two liquid water bulk phases are indicated with a and c, the two Sigracet layers are indicated with s_a and s_c and the Nafion membrane is indicated with m.*

than the temperature in part c, ranging from 3 K to 10 K in difference. s_a and s_c are the two Sigracet layers and the m indicates the Nafion membrane. In the heterogeneous system considered there exists a jump in the temperature and the chemical potential across these layers of Sigracet and over the membrane.

According to the second law of thermodynamics, σ is always positive, thus the relations between J_m, J_q, T and μ can be found. The general flux equations for two different fluxes and two different driving forces (X_1 and X_2) are given in Equations (2.4) and (2.5).

$$J_1 = l_{11}X_1 + l_{12}X_2 \tag{2.4}$$

$$J_2 = l_{21}X_1 + l_{22}X_2 \tag{2.5}$$

A combination of Equations (2.4) and (2.5) gives the expression in Equation (2.6).

$$J_1 = \left(l_{11} - \frac{l_{12}l_{21}}{l_{22}}\right) X_1 + \frac{l_{12}}{l_{22}} J_2 \tag{2.6}$$

By inserting $J_q^{'}$ and J_m instead of J_1 and J_2, the expression in Equation (2.7) is obtained. The driving force X_1 is given in Equation (2.2). The forces are assumed to be linear across the distance d.

$$J_q^{'} = \frac{1}{T^2}\left(l_{qq} - \frac{l_{mq}^2}{l_{mm}}\right)\frac{\Delta T}{d} + \frac{l_{mq}}{l_{mm}}J_m \tag{2.7}$$

By doing a similar operation for the mass flux, an equivalent equation is obtained for the J_m in Equation (2.8).

$$J_m = \frac{1}{T}\left(l_{mm} - \frac{l_{mq}^2}{l_{qq}}\right)\frac{\Delta \mu}{d} + \frac{l_{mq}}{l_{qq}}J'_q \tag{2.8}$$

The *conductivity* coefficients l_{mm}, l_{mq} and l_{qq} are given from Equations (2.9), (2.10) and (2.11).

$$\lambda_\mu = -\left(\frac{J_q^{'}}{\partial T/\partial x}\right)_{\partial \mu/\partial x=0} = \frac{1}{T^2}l_{qq} \tag{2.9}$$

$$q^* = \left(\frac{J_q^{'}}{J_1}\right)_{dT=0} = \frac{l_{qm}}{l_{mm}} \tag{2.10}$$

$$J_1 = -l_{mm}\frac{1}{T}\frac{\partial \mu_{1,T}}{\partial x} = -l_{mm}\frac{1}{T}\frac{\partial \mu_{1,T}}{\partial c_1}\frac{\partial c_1}{\partial x} = -D_{1,2}\frac{\partial c_1}{\partial x} \tag{2.11}$$

Where λ, q^* and D are the measurable coefficients in a experimental setting.

Inserting Equations (2.9), (2.10) and (2.11) into Equation (2.7) we obtain Equation (2.12). In this equation all the *conductivity* coefficients are replaced by the material properties D, λ and $\partial \mu_T/\partial c$.

$$J_{\mathrm{q}}^{'} = -\lambda\frac{\Delta T}{d} - \frac{(q^*)^2 D}{T\partial \mu/\partial c_T}\frac{\Delta T}{d} - \frac{q^* D}{\partial \mu_T/\partial c}\frac{\Delta \mu_T}{d} \tag{2.12}$$

The same operation for the mass flux by inserting Equations (2.9), (2.10) and (2.11) into Equation (2.8) equals Equation (2.13).

$$J_m = -\frac{q^* D}{T \partial \mu / \partial c_T} \frac{\Delta T}{d} - \frac{D}{\partial \mu_T / \partial c} \frac{\Delta \mu_T}{d} \tag{2.13}$$

d is the thickness of the layer, λ is the thermal conductivity of the layer, D is the diffusion coefficient and q^* is the heat of transfer. The heat of transfer is defined as the ratio between the heat flux and the molar flux when the temperature gradient is zero. This coefficient expresses the coupling of heat and mass transfer.

Similar equations are valid for the jumps across the membrane and Sigracet layers. The conductances characterize the corresponding region (s_a, m and s_c) and may depend on the temperature and other local properties. Although these quantities may vary across the membrane and the Sigracet layers, one can assume that they are constant and equal to their average values, because we are interested in magnitude of jumps. In the further analytical description the variation of all the properties across the membrane and Sigracet layers is neglected.

The Sigracet layer can be viewed as a surface with a thickness δ. In order to compare surface and bulk resistivities, one can divide their ratio by the thickness of the surface, as shown in Equation (2.14).

$$r^{s,i} = k^i \delta r^i \tag{2.14}$$

In this equation $r^{s,i}$ is the resistivity in the surface i, while r^i is the resistivity for the bulk phase i. k^i is a coefficient between these two quantities, and it is assumed to be equal to 1. The parameters used for the Sigracet layer in this model is the parameters for the Sigracet surface with the surface thickness δ taken into consideration.

The term $\partial \mu_T^m / \partial c$ for the Nafion membrane can be related to the water content χ through the expression in Equation (2.15):

$$\frac{\partial \mu_T^m}{\partial c} = \frac{RTM_m}{\rho_m a \partial \chi / \partial a M_{H_2O}} \tag{2.15}$$

The water content χ can be found using Equation (2.16)[28]:

$$\begin{aligned} \chi &= 0.043 + 17.81a - 39.85a^2 + 36.0a^3, && 0 < a < 1 \\ \chi &= 14 + 1.4(a-1), && 1 \leq a < 3 \end{aligned} \tag{2.16}$$

The term $\partial \mu_T^s / \partial c$ for the Sigracet layer is given in Equation (2.17):

$$\frac{\partial \mu_T^s}{\partial c} = \frac{RT}{c^s} \tag{2.17}$$

It can be shown [28] that the heat of transfer is proportional to the enthalpy of absorption between two layers, ΔH_{abs}, as an approximation. This relation is given in Equation (2.18). This heat of transfer is the heat going through the surface as defined in Equation (2.10), see Appendix B for figure.

$$q^* = -k \Delta H_{abs} \tag{2.18}$$

The k value is a given constant for the membrane and layers in use. The k value can vary from -1 to 1. Appendix B shows how these k-values can be approximated and shows the calculations for the k-values used in this simulation.

The partial molar enthalpy, H, depends on the water interactions with the surroundings. In the bulk phases the water enthalpy can be given as $H^i = c_p^l T$, where c_p^l is the heat capacity for the liquid phase of water. In the membrane area the water is subjected to strong interactions with the membrane. $\Delta_f H_{H_2O,l}$ for liquid water is used at $T^0 = 298$ K as a reference for the enthalpy of water in all regions of the system. The enthalpy of water in the bulk phases a and c is then be given by Equation (2.19). The enthalpy of water for the Sigracet layers is given in Equation (2.20)

$$H^i = \Delta_f H_{H_2O,l} + c_p^l (T^i - T^o) \tag{2.19}$$

$$H^s = \Delta_f H_{H_2O,l} + c_p^l(T^s - T^o) + \Delta H_{abs}(T^s) \tag{2.20}$$

ΔH_{abs} is the difference in enthalpy between water in bulk phase and in a Sigracet layer, as shown in Equation (2.21).

$$\Delta H_{abs} = H^s - H^l \tag{2.21}$$

The index i indicates bulk phase a or c of liquid water. T^i is the temperature in this region and T^o is the reference temperature of 298 K. The index s indicates either the s_a or the s_c Sigracet layer. The absorption enthalpy going from Sigracet to the Nafion membrane is approximated to be equal to the absorption enthalpy from water to Sigracet, but with the opposite sign. In other words the enthalpy of the membrane is equal to the water enthalpy in the bulk phases if one does not take the small temperature difference into consideration, $H^a = H^m = H^c$. This assumption is based on the work of Reucroft[29]. His work states that the enthalpy of the Nafion membrane is equal to that of liquid water, as long as the water content inside the membrane is high. As the water activity at both sides, a and c, for the liquid water phase is equal to 1, then it is assumed that the water content inside the membrane is high enough for this approximation. The enthalpy of the Sigracet layer H^s is expected to differ for the two different types of Sigracet as they have different material properties and structure. The absorption enthalpy will therefore be different with respect to different Sigracet type. A diagram of the enthalpy change in the system is given for a positive and negative absorption enthalpy in Figure 2.2.2 and Figure 2.2.3 respectively.

When the absorption enthalpy is found, it is then possible to find the heat capacity for water inside the Sigracet layer, c_p^s. A Δc_p value can be found by using Equation (2.22). This Δc_p can then be related to the heat capacity of water in Sigracet by use of Equation (2.23).

$$\Delta c_p = \frac{\Delta \Delta H_{abs}}{\Delta T} \tag{2.22}$$

$$c_p^s = \Delta c_p + c_p^l \tag{2.23}$$

ΔT is the temperature difference of the mean temperature (T^m) in the system, $\Delta\Delta H_{abs}$ is the difference in absorption enthalpy for the mean temperature difference and c_p^l is the heat capacity of liquid water (75 J/K mol)[6].

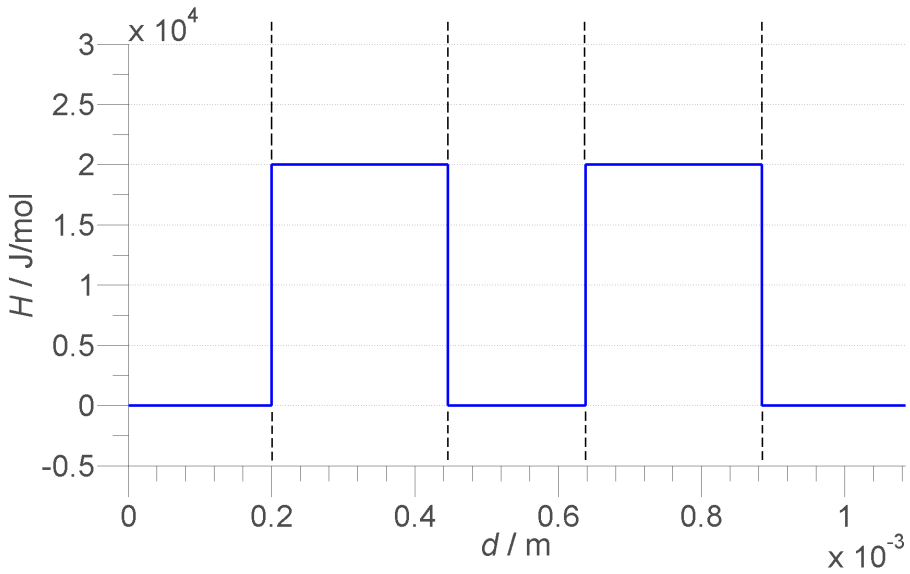

Figure 2.2.2: *A sketch of the partial enthalpy of water in the different layers of the system going from bulk phase a from the left to the bulk phase c to the right. A positive absorption enthalpy is shown for an average temperature in one experiment. The jump from zero up to the upper value is equal to the absorption enthalpy between liquid water and Sigracet layer* ΔH_{abs}*. This sketch has used the assumption of* $H^a = H^m = H^c$*.*

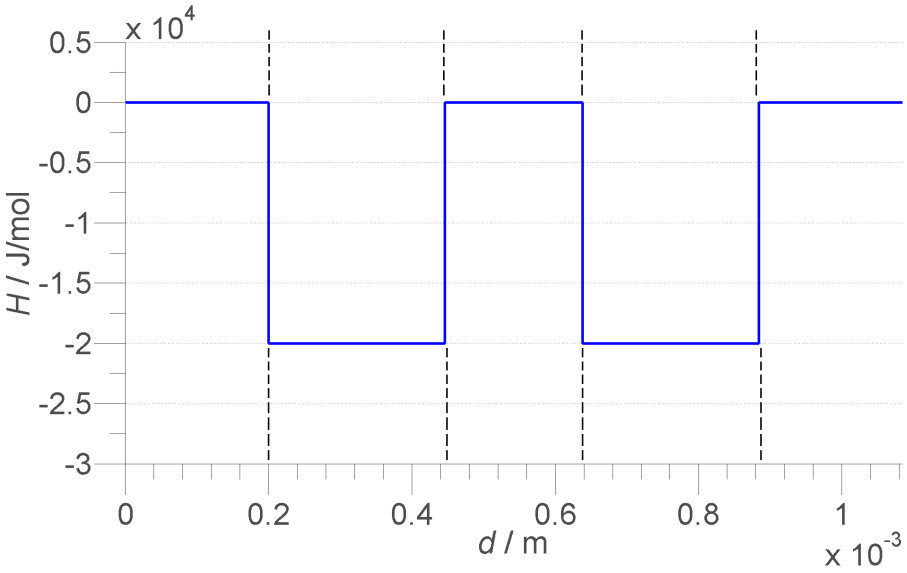

Figure 2.2.3: *A sketch of the partial enthalpy of water in the different layers of the system going from bulk phase a from the left to the bulk phase c to the right. A negative absorption enthalpy is shown for an average temperature in one experiment. The jump from zero down to the lower value is equal to the absorption enthalpy between liquid water and Sigracet layer* ΔH_{abs}*. This sketch has used the assumption of* $H^a = H^m = H^c$*.*

2.3 System properties

2.3.1 Experimental setup and results

The experimental setup is described after personal communication with Akyalcin[21], which is the person who has performed the experiments. The experiments on the water flux is done in a Nafion membrane system with two types of Sigracet layers attached at each surface of the membrane. Outside of each Sigracet layer there is a water bath with constant temperature. The water bath on the a side holds a higher temperature than the temperature in the c side water bath, $T^a > T^c$. The temperatures in these water baths are set in such a way that the mean temperature in the system, T^m, has a temperature of 30 ^{o}C, 45 ^{o}C, 60 ^{o}C and 75 ^{o}C. Apart from this, the temperature difference between a and c is set to be either 3 ^{o}C, 5 ^{o}C or 10 ^{o}C, which makes a total of 12 experiments for each Sigracet type. For each experiment the water flux from water bath a to water bath c is measured. The results of this is given in the table in Appendix A. It is of interest to find the properties of the materials for a given temperature. The measured water flux value for each mean temperature is plotted against the temperature difference in the experiment. This gives four curves with three points each for Sigracet GDL10AA and the same amount for Sigracet GDL10BA. The slope of this curve is then found by using regression of the three points for each mean temperature. See the figure in Appendix A for the these regression curves and points. The slope of these curves corresponds to the mean temperatures used in the respective experiments, and will be used along with equations to make a model to find the unknown material property of the absorption enthalpy for the Sigracet types.

2.3.2 Material properties and general constant

All material properties are taken from Non-Equillibium Thermodynamics by Kjelstrup and Bedaux[28] unless stated otherwise. In these calculations the following constants have been used: The universal gas constant R = 8.314 J/K mol and the molar mass of water M_w = 0.018 kg/mol. For the Nafion membrane the following constants have been used: Thickness of the Nafion membrane, d_m = 192 μm, which is taken from personal communication with Akyalcin[21] after his experiments. Molecular weight of the membrane M_m = 1.1 kg/mol, density of the membrane ρ_m = 1640 kg/m^3 (dry membrane is used), the thickness used for the membranes d_m=192 μm, thermal conductivity

λ_m=0.229 W/m K which is taken from the work of Burheim et al.[24] and diffusion constant D_m=8.05 $\cdot$ 10^{-9} m^2/s which is taken from the work of Villaluenga[26]. As mentioned in the theory, k-coefficients for calculating the heat of transfer with absorption enthalpy can be approximated by use of the materials heat conductivities. This has been done in Appendix B. This approximation gives three different k-coefficients for each surface in the membrane system for each Sigracet type. The following k-coefficients have been used for GDL10AA: $k_{l,s}$= 0.58 for the surface between liquid water and Sigracet layer s_a, $k_{s,m}$ = 0.65 for the surface between Sigracet layer s_a and the membrane, $k_{m,s}$ = 0.35 for the surface between the membrane and Sigracet layer s_c. The following k-coefficients have been used for GDL10BA: $k_{l,s}$ = 0.64 for the surface between liquid water and Sigracet layer s_a, $k_{s,m}$ = 0.59 for the surface between Sigracet layer s_a and the membrane, $k_{m,s}$ = 0.41 for the surface between the membrane and Sigracet layer s_c.

For the Sigracet layer the following constants have been used: The thickness of the layer, d_s=248 μm, which is taken from personal communication with Akyalcin[21] after his experiments. The diffusion constant has been estimated to be 10 times the size of water self diffusion[30], D_s= 2.3$\cdot$ 10^{-8} m^2/s. The water concentration inside the Sigracet layers are approximated to be 0.4 times the concentration of pure water. The porosity of the Sigracet material is found to be around 80 % [25], but it is assumed that the resistance for water is higher such that the water concentration is lower than what 80 % porosity would indicate. The thermal conductivity for GDL10AA and GDL10BA sigracet layer are λ_s=0.42 W/m K and λ_s=0.33 W/m K respectively[25]. As mentioned in the theory section the properties of the Sigracet can be viewed as properties of the Sigracet as a surface multiplied with the surface thickness. The parameters for the surface Sigracet would then be λ_s^s=1700 W/m^2 K for GDL10AA and λ_s^s=1300 W/m^2 K for GDL10BA. The diffusion constant for the surface would be equal to D_s^s= 9.3$\cdot$ 10^{-5} m/s.

Material properties and general constant used in the simulation are given in Table 2.3.1. All references that have been used for the properties are given, except for the calculated and estimated properties.

Table 2.3.1: *System properties used in the simulation.*

System property	Value
d_m	192 μm[21]
d_s	248 μm[21]
M_m	1.1 kg/mol[27]
M_w	0.018 kg/mol[27]
ρ_m	1940 kg/m^3[27]
λ_m	0.229 W/m K [24]
$\lambda_s(GDL10AA)$	0.42 W/m K [25]
$\lambda_s(GDL10BA)$	0.33 W/m K [25]
$\lambda_s^s(GDL10AA)$	1700 W/m^2 K
$\lambda_s^s(GDL10BA)$	1300 W/m^2 K
D_m	$8.05 \cdot 10^{-9}$ m^2/s[26]
D_s	$2.3 \cdot 10^{-8}$ m^2/s
D_s^s	$9.3 \cdot 10^{-5}$ m/s
$k_{l,s}(GDL10AA)$	0.58
$k_{s,m}(GDL10AA)$	0.65
$k_{m,s}(GDL10AA)$	0.35
$k_{l,s}(GDL10BA)$	0.64
$k_{s,m}(GDL10BA)$	0.59
$k_{m,s}(GDL10BA)$	0.41
R	8.314 J/K mol[27]
c_p^l	75 J/K mol[27]
$\Delta_f H_{H_2O}$	-285 kJ/mol[27]

2.4 Solution procedure

2.4.1 Case studies

The activity of liquid water in the a and c bulk phase are both set to 1 and the temperatures in these two regions are different, $T^a \neq T^c$. The difference in temperature between a and c, ΔT, is set as 3 K, 5 K or 10 K. The temperature difference between these two bulk phases will produce a mass transport in the system. As previously mentioned, the heat flux and the mass flux are calculated due to a assumed linear temperature drop in system. The jumps of the temperature inside each layer are calculated with the use of Equations (2.25) and Equation (2.26) using constant mass flux and heat flux. The temperature drop inside the membrane system is 10 K at maximum, and since this drop is relatively small then a mean temperature, T^m=(T^a+T^c)/2, is used in the system. The simulation has been done with four mean temperatures $T^m = 30^o$C, $T^m = 45^o$C, $T^m = 60^o$C and $T^m = 75^o$C. The parameters used in the simulation are considered unaffected by the temperature change, except for the enthalpies which is the unknown quantities in this simulation. There will be one unknown value for each mean temperature and for each type of Sigracet layer.

2.4.2 Equations and model

The entropy production used for this system involves the total heat flux instead of the measurable heat flux that was shown in the theory. This entropy production is shown in Equation (2.24). All the flux equations are derived from this entropy production.

$$\sigma = J_q \Delta \left(\frac{1}{T}\right) - J_m \Delta \left(\frac{\mu}{T}\right) \tag{2.24}$$

The advantage of using the total heat flux and the mass flux as variables in the solution procedure, is that analytical solutions can be found for the profiles of the driving forces across the membrane. As mentioned in theory the measurable heat flux $J_q^{'}$ is connected to the total heat flux J_q by the relation of $J_q' = J_q - HJ_m$. Furthermore, resistance of the whole layer will be used rather than specific local conductivities. The advantage of using resistances instead of specific local conductivities is that all layers are placed in series, which makes the resistance of the whole system simply a sum of resistances of separate layers. In other words this equals a sum of resistances of Sigracet layes s_a and

s_c, in addition to the Nafion membrane m.

Using this choice of variables, the flux Equations (2.12) and (2.13) take the form of Equations (2.25) and (2.26) for each layer in the system.

$$\Delta\frac{1}{T} = r_{qq}J_q + r_{qm}J_m \tag{2.25}$$

$$-\Delta(\frac{\mu}{T}) = r_{mq}J_q + r_{mm}J_m \tag{2.26}$$

The resistances r_{ij} are related to constants and the measurable quantities as shown in Equations (2.27), (2.28) and (2.29). The procedure of finding these resistances results from the flux Equations (2.12) and (2.13). According to the Onsager reciprocal relations the coupling coefficients r_{mq} and r_{mq} are equal, and r_{mq} will from now on be used for both of them.

$$r_{qq} = \frac{d}{\chi T^2} \tag{2.27}$$

$$r_{mq} = -(q^* + H)\frac{d}{\chi T^2} \tag{2.28}$$

$$r_{mm} = (q^* + H)^2\frac{d}{\chi T^2} + \frac{d}{TD}\frac{\partial \mu_T}{\partial c} \tag{2.29}$$

The parameters of these resistances are the same as before, d is the thickness of the layer, λ is the thermal conductivity, T is the temperature, q^* is the heat of transfer, H is the partial specific enthalpy of water in the layer and D is the diffusion constant.

Each part of the system, Nafion membrane and Sigracet layer, has a set of resistances r^s, and r^m. Since the fluxes are equal over the whole system then the jumps of $\frac{1}{T}$ and $\frac{\mu}{T}$,

given by Equations (2.25) and (2.26), can be added to get Equations (2.30) and (2.31) for the jumps of $\frac{1}{T}$ and $\frac{\mu}{T}$ over the whole system.

$$\frac{1}{T^c} - \frac{1}{T^a} = R_{qq}J_q + R_{mq}J_m \tag{2.30}$$

$$-\frac{\mu^c}{T^c} + \frac{\mu^a}{T^a} = R_{mq}J_q + R_{mm}J_m \tag{2.31}$$

Where R_{ij} is the sum of all resistances in the system as given in Equation (2.32).

$$R_{ij} = r_{ij}^{s1} + r_{ij}^{m} + r_{ij}^{s2} \tag{2.32}$$

$\Delta\mu_T$ is more convenient to use instead of $\Delta(\mu/T)$ since it is directly related to measurable properties such as the pressure or activity. These two quantities are related through the thermodynamic relation given in Equation (2.33)

$$\Delta\frac{\mu}{T} = \frac{\Delta\mu_T}{T} + H\Delta\frac{1}{T} \tag{2.33}$$

As the activity in both a and c bulk phase are equal to 1, then the quantity $\Delta\mu_T$ is zero, and only the $H\Delta(1/T)$ term is used in this system. By combining Equation (2.30) and (2.31), and using the expression for $\Delta(\mu/T)$ above, then the fluxes can be written as done in Equation (2.34) and (2.35).

$$J_m = -\frac{H(T^m)R_{qq} + R_{mq}}{\Re}\Delta\frac{1}{T} \tag{2.34}$$

$$J_q = \frac{H(T^m)R_{mq} + R_{mm}}{\Re}\Delta\frac{1}{T} \tag{2.35}$$

The $\Re$ is related to the resistances R_{ij} as given in Equation (3.28).

$$\Re = R_{mm}R_{qq} - R_{mq}^2 \tag{2.36}$$

2.4.3 Matlab procedure

All calculations are done in the software program Matrix Laboratory (MatLab) where a loop of different absorption enthalpy is used to give the same results with simulations as with experiments. See Appendix E for full MatLab code. As mentioned earlier, all parameters apart from the absorption enthalpy, are considered unaffected by temperature and thus set as a constant. In the experimental data, four mean temperatures, T^m = 30 ^{o}C, 45 ^{o}C, 60 ^{o}C and 75 ^{o}C were used for the two different Sigracet layer types. Each of this mean temperature is an independent simulation, which gives a corresponding absorption enthalpy as a result. Inside the loop with different absorption enthalpies the program starts with calculating the resistivity, $R_{i,j}$, for the mean temperature as input parameter. The program then proceeds to calculate the mass and heat flux based on the resistivities and the temperature difference. The results from the experiments is the slope $J_w/\Delta T$. The value of the temperature difference used in the simulations is therefore not too important, as it is assumed that the slope is a constant value. The experiments have used 3 ^{o}C, 5 ^{o}C and 10 ^{o}C as the temperature difference, and found the average of these results. The simulation model however needs just one temperature difference to find the mass and heat flux, so 3 ^{o}C is chosen for all the simulation calculations. When the simulation loop finds the matching result for the $J_w/\Delta T$ result from the experiments, then the corresponding absorption enthalpy is found for this mean temperature. This procedure is repeated for all 4 mean temperatures for both Sigracet types. The graphical solution of this is given in the results section.

2.5 Results

The absorption enthalpy of water when going from liquid state into the Sigracet layer was found by a graphical solution. The simulated water flux value divided by the temperature difference is plotted for various absorption enthalpies. The constant experimental value of water flux divided by temperature difference is plotted in the same figure. It is the intersection between these two graphs that gives the absorption enthalpy in the x-axis for the corresponding mean temperature. A graphical representation for the different mean temperatures of the results is given in Figure 2.5.1, Figure 2.5.2, Figure 2.5.3 and Figure 2.5.4 for Sigracet GDL10AA without Teflon. A similar graphical representation for the different mean temperatures for Sigracet GDL10BA layers is given in Figure 2.5.5, Figure 2.5.6, Figure 2.5.7 and Figure 2.5.8. The resulting values for the absorption enthalpy, after the intersection in the figures, are given in Table 2.5.1 and Table 2.5.2 for GDL10AA and GDL10BA respectively. An example of the temperature profile inside the membrane and the Sigracet layers is shown for GDL10AA, T^m=30 ^{o}C in Figure 2.5.9. The c_p^s values for each Sigracet layer have been given in Table 2.5.3.

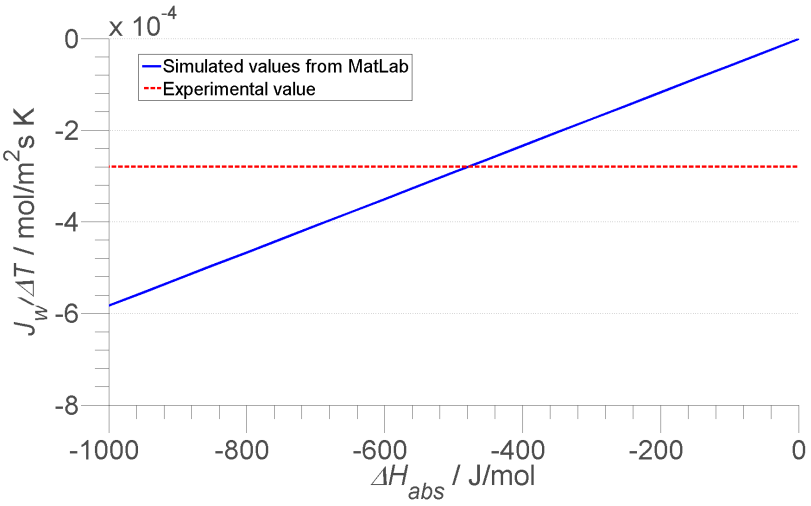

Figure 2.5.1: *Experimental value of $J_w/\Delta T$ plotted against simulated values for different ΔH_{abs} for T^m= 30 ^{o}C for Sigracet GDL10AA layers. The intersection between the experimental line and the simulated line gives the absorption enthalpy for this mean temperature equal to -460 J/mol.*

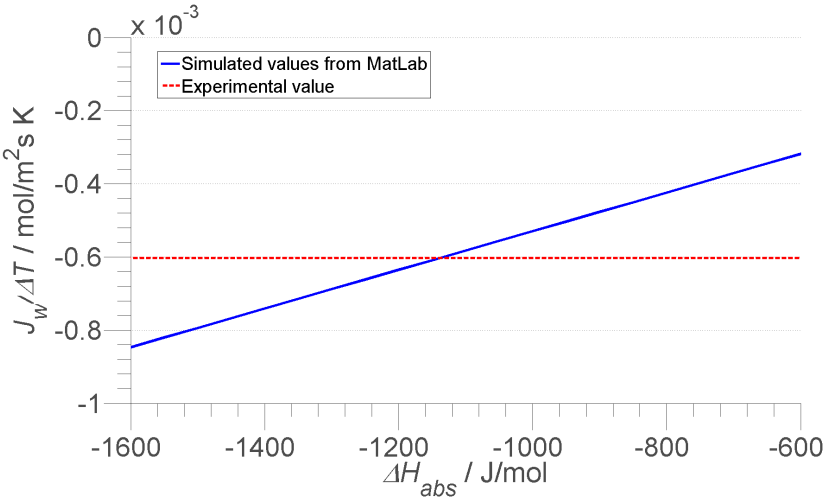

Figure 2.5.2: *Experimental value of $J_w/\Delta T$ plotted against simulated values for different ΔH_{abs} for T^m= 45 °C for Sigracet GDL10AA layers. The intersection between the experimental line and the simulated line gives the absorption enthalpy for this mean temperature equal to -1140 J/mol.*

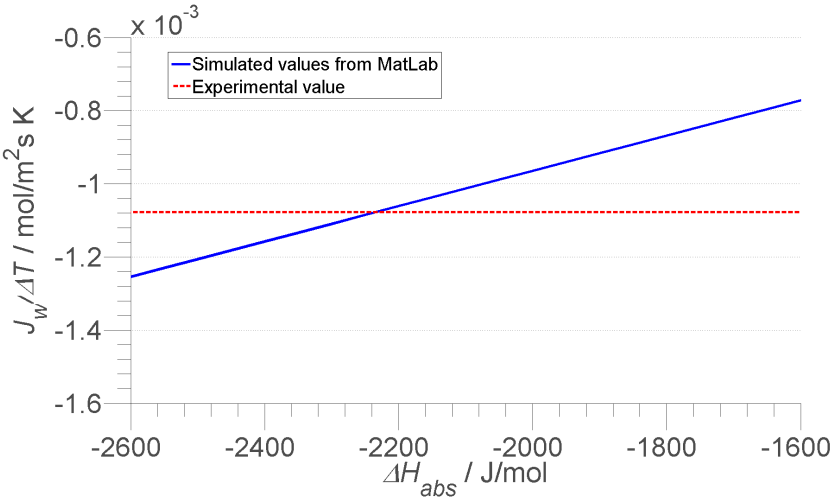

Figure 2.5.3: *Experimental value of $J_w/\Delta T$ plotted against simulated values for different ΔH_{abs} for T^m= 60 °C for Sigracet GDL10AA layers. The intersection between the experimental line and the simulated line gives the absorption enthalpy for this mean temperature equal to -2230 J/mol.*

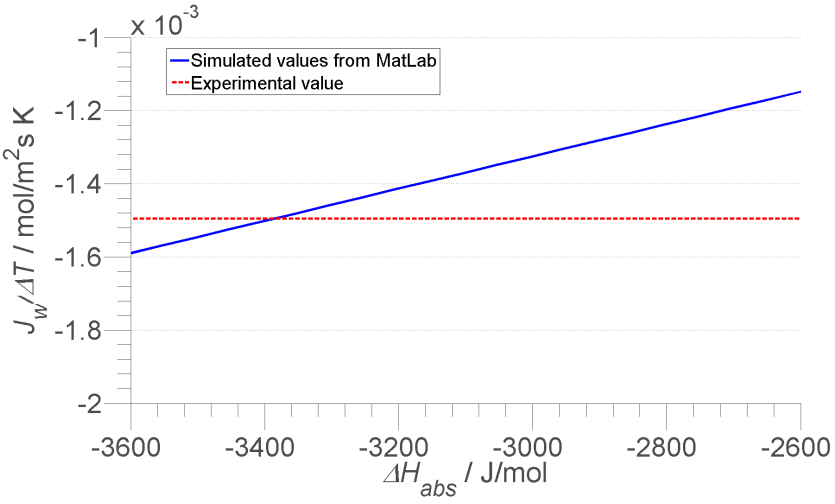

Figure 2.5.4: *Experimental value of $J_w/\Delta T$ plotted against simulated values for different ΔH_{abs} for $T^m = 75\ ^oC$ for Sigracet GDL10AA layers. The intersection between the experimental line and the simulated line gives the absorption enthalpy for this mean temperature equal to -3380 J/mol.*

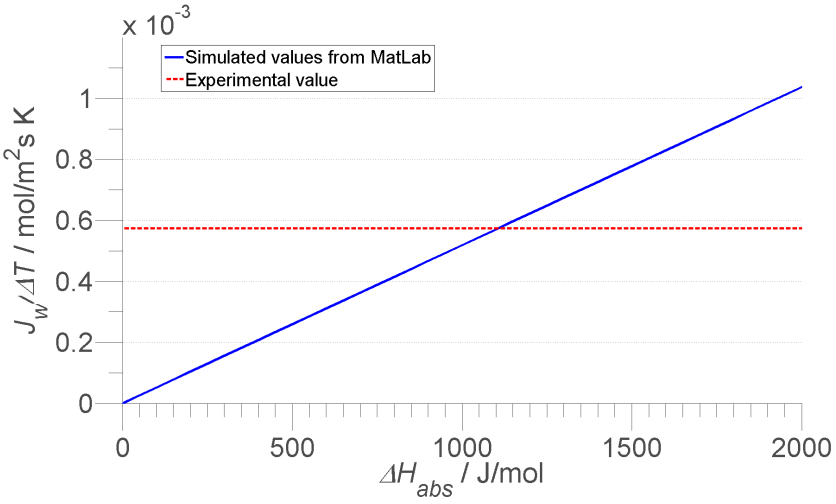

Figure 2.5.5: *Experimental value of $J_w/\Delta T$ plotted against simulated values for different ΔH_{abs} for $T^m = 30\ ^oC$ for Sigracet GDL10BA layers. The intersection between the experimental line and the simulated line gives the absorption enthalpy for this mean temperature equal to 1120 J/mol.*

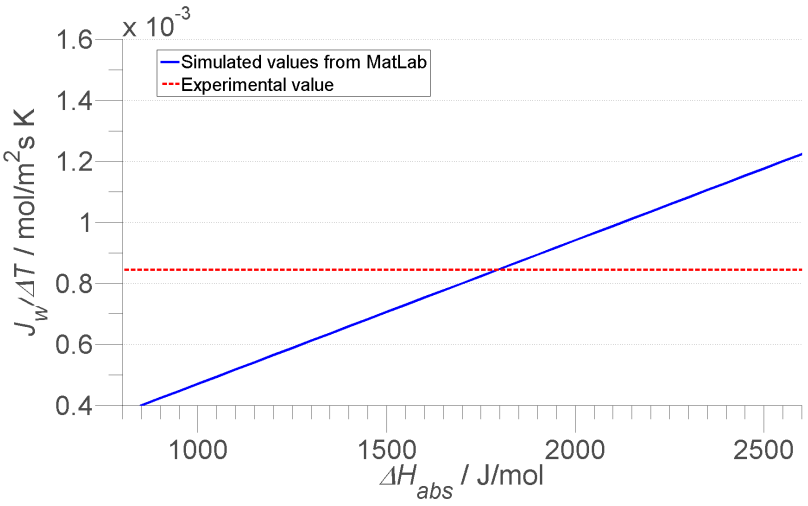

Figure 2.5.6: *Experimental value of $J_w/\Delta T$ plotted against simulated values for different ΔH_{abs} for $T^m = 45\ ^oC$ for Sigracet GDL10BA layers. The intersection between the experimental line and the simulated line gives the absorption enthalpy for this mean temperature equal to 1800 J/mol.*

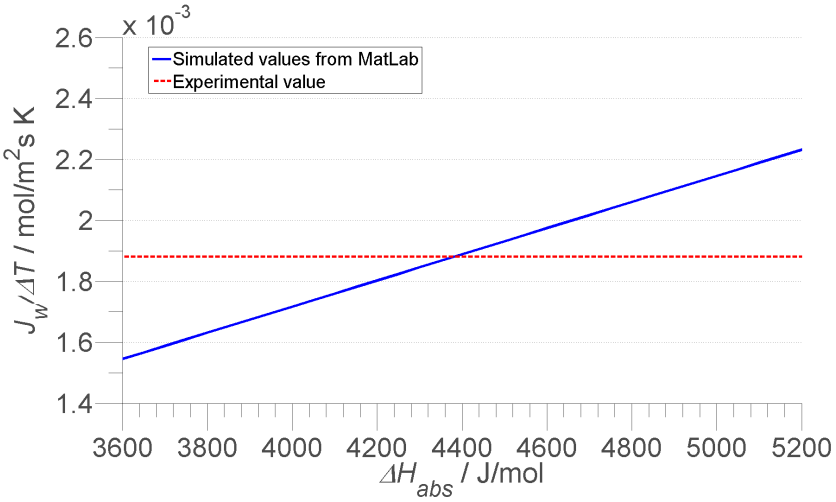

Figure 2.5.7: *Experimental value of $J_w/\Delta T$ plotted against simulated values for different ΔH_{abs} for $T^m = 60\ ^oC$ for Sigracet GDL10BA layers. The intersection between the experimental line and the simulated line gives the absorption enthalpy for this mean temperature equal to 4380 J/mol.*

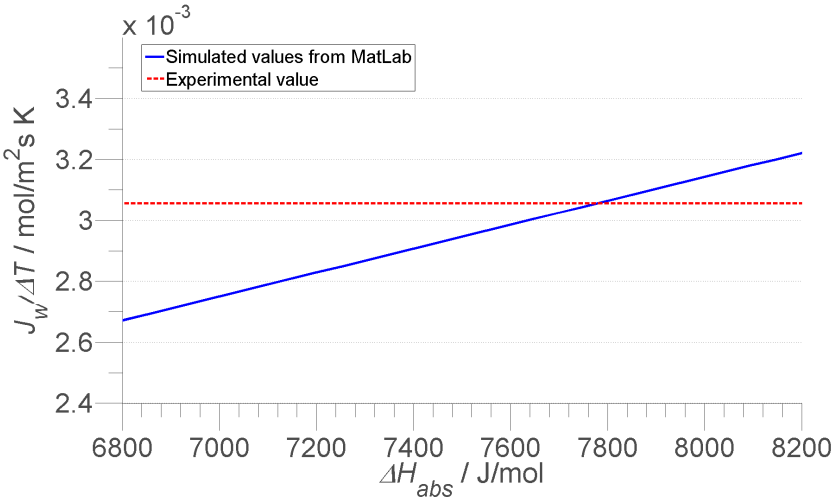

Figure 2.5.8: *Experimental value of $J_w/\Delta T$ plotted against simulated values for different ΔH_{abs} for $T^m = 75\ ^oC$ for Sigracet GDL10BA layers. The intersection between the experimental line and the simulated line gives the absorption enthalpy for this mean temperature equal to 7780 J/mol.*

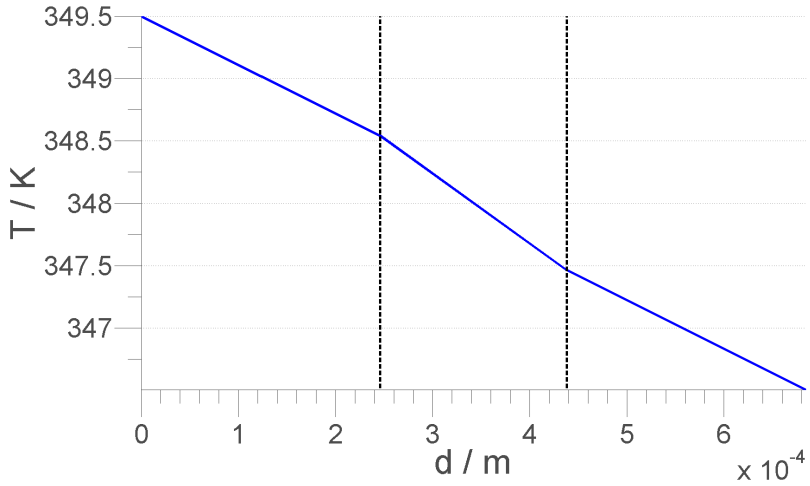

Figure 2.5.9: *The temperature jumps in the system for $T^m = 30\ ^oC$ for GDL10AA, starting from water bath a at the left side and going through the sigracet layers and membrane to the water bath c on the right side. The dashed vertical lines indicates where the Sigracet layers and the membrane start and stop.*

Table 2.5.1: *Experimental flux values for GDL10AA Sigracet layer without Teflon and the corresponding simulated absorption enthalpy value for the different mean temperatures.*

$T^m(^oC)$	Experimental value, $J_m/\Delta T$ / mol/m^2 s K	ΔH_{abs} / J/mol
30	-2.8·10^{-4}	-460
45	-6.0·10^{-4}	-1140
60	-1.1·10^{-3}	-2230
75	-1.5·10^{-3}	-3380

Table 2.5.2: *Experimental flux values for GDL10BA Sigracet layer with 5 % Teflon and the corresponding simulated absorption enthalpy value for the different mean temperatures.*

$T^m(^oC)$	Experimental value, $J_m/\Delta T$ / mol/m^2 s K	ΔH_{abs} / J/mol
30	5.7·10^{-4}	1120
45	8.4·10^{-4}	1800
60	1.9·10^{-3}	4380
75	3.1·10^{-3}	7780

Table 2.5.3: *Heat capacity values for water inside Sigracet GDL10AA and GDL10BA, c_p^s.*

Sigracet type	c_p^s / J/K mol
GDL10AA	10
GDL10BA	223

2.6 Discussion

2.6.1 Absorption enthalpy for Sigracet layer with and without teflon

The experiments have shown that it is possible to change the sign and value of the water flux by using Sigracet layers with different material properties. It now seems that we have a model to explain the observations. The GDL10AA Sigracet layer without Teflon had a negative absorption enthalpy, ΔH_{abs}, ranging from -460 J/mol to -3380 J/mol. The difference between absorption enthalpy for each mean temperature is increasing when the mean temperature is increasing. The largest difference in absorption enthalpy value is for the difference between mean temperatures of 60 ^{o}C and 75 ^{o}C with a difference value -1150 J/mol. The lowest difference in absorption enthalpy value is for the difference between mean temperatures of 30 ^{o}C and 45 ^{o}C with a difference value -680 J/mol. The difference is almost twice as large for the higher mean temperatures than for the lower ones. This trend of increase in the difference could be explained by temperature effects in the materials which so far have been unaccounted for. Another possible solution to the variations is the effect temperature has on water. The vapor pressure of water is increasing and the viscosity is decreasing[31] when the temperature is increasing. Including these effects may explain the variations as the mean temperature in the system is increasing. The variation in difference in absorption enthalpies for varying mean temperature is neglected, which results in a c_p value for water inside the Sigracet layer equal to 10 J/K mol. The resulting value is positive and the validity of this value will discussed in a later subsection.

The GDL10BA Sigracet layer with Teflon had a positive absorption enthalpy, ΔH_{abs}, ranging from 1120 J/mol to 7780 J/mol. The effect with increasing difference between absorption enthalpies for higher mean temperatures is also present for this type of Sigracet layer. The largest difference in absorption enthalpy value is for the difference between mean temperatures of 60 ^{o}C and 75 ^{o}C with a difference value 3400 J/mol. The lowest difference in absorption enthalpy value is for the difference between mean temperatures of 30 ^{o}C and 45 ^{o}C with a difference value 680 J/mol. The difference is five times larger for the higher temperatures than for the lower ones. The absorption enthalpy increases greatly for the last two mean temperatures, compared to the difference for the smaller temperatures. This could perhaps be explained in the same manner as for the Sigracet layer without teflon with respect to temperature effects. Temperature effects for the

material properties of both Sigracet and Nafion membrane are neglected and this could provide the increase in absorption enthalpy which is present in the experimental results. Temperature effects in the water can also be one explanation for causing the variations, as explained for the Sigracet GDL10AA type. Other temperature effects not mentioned may also be present. This can affect the absorption enthalpy between liquid water and Sigracet, and research on these effects would prove useful for the model. The variation in difference in absorption enthalpies for varying mean temperature is neglected, which results in a c_p value for water inside the Sigracet layer equal to 223 J/K mol. The resulting value is positive and the validity of this value will discussed in a later subsection.

It is apparent that the temperature effect is greater for GDL10BA with Teflon then for GDL10AA without Teflon by looking at the absorption enthalpy differences between the mean temperatures. The numeric value of the difference between absorption enthalpy between the mean temperatures is larger for Sigracet GDL10AA than for Sigracet GDL10BA. The experimental values increase greatly for Sigracet GDL10BA after the mean temperature reaches 60 ^{o}C. This trend is not happening for Sigracet GDL10AA, which could indicate the Sigracet GDL10BA type to have some other temperature effects that is not present in the Sigracet type GDL10AA. One possibility is that the Teflon enhances some of the material properties, which obviously would not be present in the Sigracet GDL10AA type without Teflon. There may also be other changes in the Nafion membrane and the Sigracet layers caused by the temperature, such as alterations in heat conductivity and in diffusion constant. The reason for suspecting this comes from the fact that increase in mean temperature indeed has an effect on the absorption enthalpy increase in the experiments, even for the Sigracet type without Teflon. It is essential for future experiments to try to establish these temperature effects on the material properties, such as heat conductivity and diffusion constant, to get more accurate results.

The study of Burheim et al.[17] shows how aging affects porous transport layers containing PTFE (Teflon). They found that aging of the porous transport layers resulted in lower contact angles, in other words became less hydrophobic, and the PTFE content was found to decrease. In response to this effect, the thermal conductivity of the layers containing water increased with ageing time. This is an interesting result that can be compared to the study done in this part. A decrease in PTFE and increase in conduc-

tivity would decrease the enthalpy values in the membrane system. This scientific result added with the water transport results for the different Sigracet layers in our simulation model makes this an important aspect when trying to build an efficient fuel cell. Further study on the effect from ageing on the enthalpy values would provide essential information and knowledge to the simulation model and fuel cells in general.

2.6.2 Sensitivity analysis

Different values for the diffusion constant and heat conductivity for the Sigracet layer and the Nafion membrane have been used in the simulation to see how a change in them affects the flux value. The motivation for this came from the apparent temperature effect on the mass flux for the Sigracet layers for higher mean temperatures. It became apparent from the simulation that a change in the heat conductivity had little, if any, effect on the mass flux value. Note that this is from changing the heat conductivity alone and holding everything else constant. The k-coefficients used to find the heat of transfer are strongly dependent on the heat conductivities and changing them would give different results. However, if the k-coefficients are set as constant, while still changing the heat conductivities, then the heat conductivity had to be increased with a factor of 100 or higher to even see an effect. This result was the same for either of the Sigracet type heat conductivities and the membrane heat conductivity. If the k-coefficients used to calculate the heat of transfer is changed along with the heat conductivities then this has a larger impact on the result. Changing the heat conductivities with a factor of 100 would give approximately a 10 % change in the results. The diffusion constant however, had a rather large impact on the flux value. A factor of 1.5 or 2 gave a noticeable change. In the simulation the impact caused by the change in the diffusion constant of the Sigracet was dominant, and in comparison the effect the change in diffusion constant of the membrane had to simulation outcome was only minimal. However, even this had more significant effect in simulation than any of the changes made in the heat conductivities. This means that the Sigracet layer must have a greater resistance when it comes to mass diffusion when compared to the membrane in order to explain the data. It also means that the value of the diffusion constant plays a greater role in deciding the mass flux compared to the heat conductivities. In other words, the water flux through the system is heavily diffusion controlled and changes in the diffusion constant have a large impact on the result. Having a correct diffusion constant for this model is there-

fore important for more accurate results; especially the Sigracet diffusion constant is of the utmost importance for the model whenever the mass flux results have problematic fluctuations, and a precise result is needed. The heat conductivities plays a minor role, and are not too important for the result of this model.

2.6.3 k-value approximation

The assumption used to calculate the heat of transfer k-coefficients may give a different result from using real values, but a sensitive analysis on this matter shows that changing these k-coefficients have small impact on the result of the absorption enthalpy. Initially an assumption of 0.5 for all k-values was used instead of the current used values. Using a value of 0.5 for all k-coefficients gave results that are 5 % to 10 % different from the results reported in this paper. The difference is not large, and one can therefore conclude that the approximation used in this report is good enough for this model as the effect from the heat of transfer coefficient is not too important for the final result. The sign of the k-coefficients are all positive, which means that the heat of transfer going through the surfaces between the layers are all with the opposite sign of the absorption enthalpy between the layer. The heat of transfer will therefore be dependent on the sign of the absorption enthalpy, which depends on the type of the Sigracet layer and in which surface it goes through.

2.6.4 Reasonability of the simulated values

The c_p^s values for both GDL10AA and GDL10BA are positive, and are thus physically possible. Initially the results gave a negative heat capacity value for GDL10AA when using a smaller diffusion constant for the Sigracet layer than approximately $8 \cdot 10^{-9}$ m^2/s. This diffusion constant was originally set as the self diffusion constant for water[30], which is $2.3 \cdot 10^{-9}$ m^2/s. However, this gave a negative value for the heat capacity, which could imply that the diffusion constant was a not a correct estimation or some of the other parameters were badly estimated or wrong. The diffusion constant for water in the Sigracet layer is one of the biggest uncertainties in this simulation as there are no

experiments to back up the estimated value. Experimental results and study on the heat capacity of water and absorption enthalpies when going into different types of Sigracet layers would prove useful for the validity of this simulation model. It would also provide better knowledge about how water transport is behaving inside the Sigracet layers.

2.7 Conclusion

Theory of irreversible thermodynamics is used to properly describe the fluxes and forces in a Nafion membrane system consisting of outer Sigracet layers. A simulation model for heat and transport of water through the system is established, where variations is restricted to be in one dimension. Knowledge about how water transport and heat behave in these porous transport layers is essential for creating an efficient proton exchange membrane fuel cell. The absorption enthalpy going from liquid water into the Sigracet layer is found for two types of Sigracet, GDL10AA without Teflon and GDL10BA with 5 % Teflon. Flux value data from experiments in this membrane cell system has been used to find these enthalpy values corresponding to different mean temperatures in the cell. For GDL10AA without Teflon this absorption enthalpy ranges from -550 J/mol to -3750 J/mol for mean temperatures of 30 ^{o}C and 75 ^{o}C respectively. For GDL10BA with Teflon this absorption enthalpy ranges from 1150 J/mol to 7850 J/mol for mean temperatures of 30 ^{o}C and 75 ^{o}C respectively. The heat capacity value of water, c_p^s, for Sigracet GDL10AA and GDL10BA were found to be 10 J/K mol and 223 J/K mol respectively.

Through the simulations it has been found that the heat conductivities play a minor role for the mass flux for this system and changing values for either the membrane or the Sigracet layer gave an insignificant change in the mass flux. The transport of water is much more dependent on the mass diffusion constant and changing this value for either the membrane or the Sigracet layer makes a relatively huge change in the mass flux. Further study about the diffusion constant of water in Sigracet and Nafion would be essential for the simulation model and its accuracy.

2.8 Nomenclature

Symbol	Units	Description
a	$-$	Activity
c	mol/m^3	Concentration
c_p^l	J/K mol	Heat capacity for liquid water
c_p^s	J/K mol	Heat capacity for water in a Sigracet layer
ΔH_{abs}	J/mol	absorption enthalpy of liquid water into Sigracet
$\Delta_f H_{H_2O,l}$	kJ/mol	Enthalpy of formation of liquid water
D	m^2/s	Mass diffusion coefficient
d	m	Thickness
e	J/mol	Total energy
H	J/mol	Partial molar enthalpy
J_m	mol/m^2 s	Mass flux of water
J_q	J/m^2 s	Total heat flux
$J_q^{'}$	J/m^2 s	Measurable heat flux
k	$-$	Ratio used between heat of transfer and absorption enthalpy
λ	W/m K	Thermal conductivity
l_{qq}	J K/m s	Heat coefficient
$l_{\mu q}$	mol K/m s	Mass and heat coupling coefficient
$l_{\mu\mu}$	mol^2 K/J m s	Mass coefficient
M_m	kg/mol	Molar mass of membrane
M_w	kg/mol	Molar mass of water
μ	J/mol	Chemical potential
$\Re$	m^4 s^2/K^2 mol^2	Ratio of resistivities
ρ_m	kg/m^3	Density of membrane
R	J/K mol	Universal gas constant
R_{qq}	m^2 s/K	Sum of heat resistivity over all layers
R_{mq}	m^2 s/K mol	Sum of heat and mass coupling resistivity over all layers
R_{mm}	m^2 s J/ K mol^2	Sum of mass resistivity over all layers
r_{qq}	m^2 s/K J	Heat resistivity
r_{mq}	m^2 s/K mol	Heat and mass coupling resistivity
r_{mm}	m^2 s J/ K mol^2	mass resistivity
σ	J/K mol	Entropy production

T	K	Temperature
T^o	K	Temperature of standard state
t	s	Time
u	J/mol	Internal energy
q^*	J/mol	Heat of transfer
χ	kg H_2O/kg membrane	Water content
X_i	varies	Driving force

2.9 References

References

1. G. P. Beretta, World energy consumption and resources: an outlook for the rest of the century.

2. G. Tverberg, http://ourfiniteworld.com/2012/03/12/world-energy-consumption-since-1820-in-charts/

3. I. Dincer, Hydrogen and Fuel Cell Technologies for Sustainable Future, Jordan Journal of Mechanical and Industrial Engineering Volume 2, Number 1, Mar. 2008.

4. O. Burheim, Personal communication and use of his presentation: Energiutfordringer og løsninger.

5. Alternative Energy: Wind Electricity, http://greenliving.nationalgeographic.com/alternative-energy-wind-electricity-2191.html.

6. FuelCellToday, http://www.fuelcelltoday.com/about-fuel-cells/history

7. Global Energy Innovations, http://www.geifuelcells.com/innovation/ht-pem

8. A. Z. Weber and J. Newman, Modeling Transport in Polymer-Electrolyte Fuel Cells.

9. S. Kjelstrup, A. Røsjorde, Local and Total Entropy Production and Heat and Water Fluxes in a One-Dimensional Polymer Electrolyte Fuel Cell, J. Phys. Chem. B 2005, 109, 9020 - 9033.

10. P. J.S. Vie, S. Kjelstrup, Thermal conductivities from temperature profiles in the polymer electrolyte fuel cell, Electrochimica Acta 49 (2004) 1069 - 1077. A. Z. Weber and J. Newman, Modeling Transport in Polymer-Electrolyte Fuel Cells.

11. C. M. Bautista-Rodríguez, M. G. A. Rosas-Paleta, J. A. Rivera-Marquez, N. T. Ochoa, Effects on the PEMFC Performance by Combination of Gas Distribution Plates and Gas Diffusion Medias, Int. J. Electrochem. Sci., 6 (2011)256 - 269.

12. J.G. Pharoah, O.S. Burheim, P.J.S Vie, S. Kjelstrup, THERMAL EFFECTS IN PEMFCS.

13. A. Thomas, G. Maranzana, S. Didierjean, J. Dillet and O. Lottin, Measurements of Electrode Temperatures, Heat and Water Fluxes in PEMFCs: Conclusions about Transfer Mechanisms, J. Electrochem. Soc. 2013, Volume 160, Issue 2, Pages F191-F204.

14. A. Thomas, G. Maranzana, S. Didierjean, J. Dillet, O. Lottin, Thermal Effect on Water Transport in Proton Exchange Membrane Fuel Cell, (2011).

15. C. J. Bapat, S. T. Thynell , Anisotropic Heat Conduction Effects in Proton-Exchange Membrane Fuel Cells, J. Heat Transfer 129(9), 1109-1118 (Jul 26, 2006) (10 pages).

16. S. Kjelstrup, P.J.S. Vie, L. Akyalcin, P. Zefaniyaa, J.G. Pharoahe, O.S. Burheim, The Seebeck coefficient and the Peltier effect in a polymer electrolyte membrane cell with two hydrogen electrodes, Electrochimica Acta Volume 99, 1 June 2013, Pages 166-175.

17. O.S. Burheim, G. Ellila, J.D. Fairweather, A. Labouriau, S. Kjelstrup, J.G. Pharoah, Ageing and thermal conductivity of Porous Transport Layers used for PEM Fuel Cells.

18. Y.S Chen, H. Peng, Studying the Water Transport in a Proton Exchange Membrane Fuel Cell by Neutron Radiography and Relative Humidity Sensors.

19. K. Nishida, S. Tsushima, S. Hira, Water Management and ExperimentalDiagnostics in Polymer Electrolyte Fuel Cell .

20. Wikipedias page on fuel cells, http://en.wikipedia.org/wiki/Fuel_cell

21. L. Akyalcin, personal communication about the experiments performed on water transport in the membrane system.

22. Glavatskiy, K., Pharoah, J. and Kjelstrup, S. (NTNU) Membrane boundary conditions for transport of water and heat.

23. Glavatskiy, K., Pharoah, J, Kjelstrup, S, Thermal phenomena associated with water transport across a fuel cell membrane: Soret and Dufour effects.

24. O. S. Burheim, P.J.S. Vie, J.G. Pharoaha, S. Kjelstrup, Ex situ measurements of through-plane thermal conductivities in a polymer electrolyte fuel cell, Journal of Power Sources 195 (2010) 249-256.

25. O. S. Burheim, J.G. Pharoah, H. Lampert, P.J.S. Vie, S. Kjelstrup, Through-Plane Thermal Conductivity of PEMFC Porous Transport Layers.

26. J.P.G Villaluenga, B. Seoana, V.M. BarragÃąn, C. Ruiz-BauzÃą, Thermo-osmosis of mixtures of water and methanol through a Nafion membrane.

27. S. Kjelstrup, D. Bedaux, E. Johannessen, J. Gross, Non-Equilibrium Thermodynamics for Engineers, 2010.

28. Kjelstrup, S., Bedeaux, D. (2008) Non-equilibrium thermodynamics for heterogeneous systems, Series on Statistical mechanics, Vol 16, World Scientific, Singapore.

29. P.J. Reucroft, D. Rivin, N.S. Schneider, Thermodynamics of Nafion - vapor interactions. I. Water vapor, 2002.

30. M. Spyrou, The Diffusion Coefficient of Water: A Neutron Scattering Study using Molecular Dynamics Simulations.

31. J. Kestlin, M. Sokolov, W. A. Wakeham, Viscosity of Liquid Water in the Range -8 ^{o} C to 150 ^{o} C.

32. Engineeringtoolbox, http://www.engineeringtoolbox.com/thermal-conductivity-d_429.html

3 Part 3 - Proton Exchange Membrane Fuel Cell with heat and transport of mass and charge

3.1 Introduction

Knowledge about temperature control, transport processes and similar phenomena associated with the Proton Exchange Membrane(PEM) fuel cell is important for the developing of new and better fuel cells with high efficiency. This part is a study on the variations in temperature, heat and other important quantities for the PEM fuel cell as well as describing the various phenomenas occurring in the different segments and surfaces. In depth study on the material properties and how they affect the fuel cell is an important study that is needed to properly create efficient fuel cells, and it is also included in this part.

The non-isothermal polymer electrolyte fuel cell has been studied by Kjelstrup and Røsjorde [1] and Vie et al.[2]. Similar studies have shown how the heat production is a large quantity and varies with the electric current[3]. Other studies have shown how the temperature is varying across the fuel cell as a result of the heat flux[4]. It is of interest to establish a model that can explain these variations and how they are affected by material properties.

Theory of irreversible thermodynamics is used to properly describe the transport processes taking place in a Proton Exchange Membrane fuel cell. In this fuel cell there are three fluxes: heat, mass and charge. This results in a system with three main driving forces and three coupled forces. The system is reduced to have variations in only one dimension to further simplify the equations and the calculations. It is important to know how the temperature, electrical potential, heat flux and chemical composition vary through the fuel cell to get a complete understanding of the system. This knowledge can be used to build more efficient fuel cells. For instance, knowledge about the heat flux will contribute to improve the efficiency of the heat exchanger. Another example is knowledge about temperature variations and temperature control, which can help to determine the correct operating temperature for the fuel cell. An example of a typical PEM fuel cell is given in Figure 3.1.1.

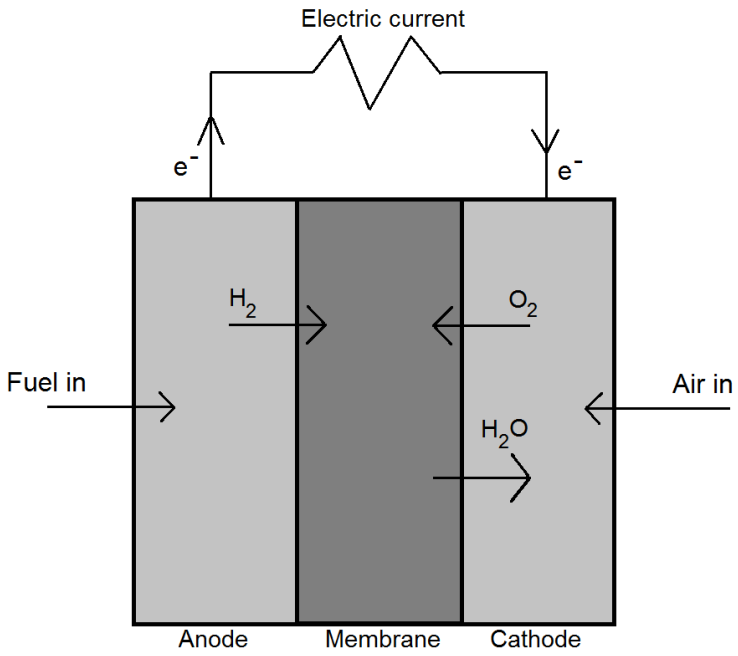

Figure 3.1.1: *Simple schematic of the PEM fuel cell with mass transport of hydrogen, oxygen and water given with arrows. Electrons are transferred from the anode to the cathode along the shown line.*

This work is based on the paper on local and total entropy production and heat and water fluxes in a 1-dimensional polymer electrolyte fuel cell by S. Kjelstrup and A. Røsjorde[1] and the theoretical approach on non-equilibrium thermodynamics by S. Kjelstrup and D. Bedaux[5].

To establish a model for such a fuel cell, the cell is divided into five subsections, where each part has a set of equations derived from irreversible thermodynamics. These subsections are the anode backing and cathode backing, made of Sigracet GDL10AA. In the middle of these two backings is a Nafion membrane. Additionally, it is convenient to add surfaces between the Nafion membrane and the anode and cathode backing. The surfaces are regarded as independent segments of the fuel cell. It is in these two surfaces the electrochemical reactions take place, and it is important to include these two surfaces in order to satisfy both the total energy and entropy balance.

The model of this fuel cell has a finite set of independent differential equations, energy-

and entropy balances, and therefore a set of boundary conditions is needed to fully explain the behavior inside the cell. In this paper, these conditions are all given at the beginning of the anode backing. These boundary conditions are the temperature, mole fraction of water, heat flux and electrical potential. These boundary conditions can be changed accordingly to what is more appropriate for the user of the model. Here they are, however, set as $T^0 = 330$ K, $x_w = p_w^*/p_{tot}$, $J_q^0 = 1000$ W/m^2 and $\phi^0 = 0$ V. Both the mole fraction of water and the electrical potential are as per definition, while the temperature is a parameter that can be changed to what one prefer. The initial heat flux value is just a guess and will be corrected in the program by the use of the total energy balance in the cell. The unknown profiles in the cell are found by integrating the differential equations and energy balance for each segment in the cell.

3.2 Theory

The theory is taken from Non-Equilibrium Thermodynamics for Heterogeneous Systems by S. Kjelstrup and D. Bedeaux[5] and Non-Equilibrium Thermodynamics for Engineers by S. Kjelstrup et al.[6] unless stated otherwise in the text. A Proton Exchange Membrane (PEM) fuel cell converts hydrogen and oxygen to water and produce electric work. The anode and cathode surface reactions are given in Equations (3.1) and (3.2).

$$\frac{1}{2}H_2(g) \rightarrow H^+ + e^- \tag{3.1}$$

$$\frac{1}{4}O_2(g) + H^+ + e^- \rightarrow \frac{1}{2}H_2O(l) \tag{3.2}$$

Since the gases are humid then t_w^m moles of water is transferred through the membrane per mole of protons transferred. The overall cell reaction is given in Equation (3.3).

$$\frac{1}{2}H_2(g) + \frac{1}{4}O_2(g) + t_w^m H_2O(l) \rightarrow \left(t_w^m + \frac{1}{2}\right) H_2O(l) \tag{3.3}$$

When there is no current density, in other words when j $\rightarrow$ 0 A/m^2, then the fuel cell goes to the limit of being a reversible electrochemical cell. In this special case the maximum cell potential, which is the potential difference between the anode and cathode, will be equal to the standard cell potential E_{cell}^0. This quantity can be calculated from using an oxidation-reduction table, or it can be calculated from the Gibbs free energy, $\Delta_n G$ through the relation given in Equation (3.4).

$$E_{cell}^0 = -\frac{\Delta_n G}{nF} \tag{3.4}$$

In this equation n is the mole of electrons transfered, which in the case of Equation (3.3) is equal to one, and F is Faraday's constant. Gibb's free energy is related to enthalpy and entropy through the relation given in Equation (3.5).

$$\Delta_n G = \Delta_n H - T\Delta_n S \tag{3.5}$$

$\Delta_n H$ is the enthalpy change for Equation (3.3), $\Delta_n S$ is the entropy change for Equation (3.3) and T is the temperature of the system. Using these relations makes it possible to calculate the cell potential of an electrochemical cell under reversible conditions as long as the thermodynamic properties are known. It is the main method for calculating the cell potential for an electrochemical cell at reversible conditions. This is however not the case when dealing with a typical fuel cell, where there is transport of heat, mass and charge under irreversible conditions. In order to get a full understanding of this fuel cell under irreversible conditions it is necessary to apply non-equilibrium thermodynamics in combination with transport equations and energy balances.This will from now on the be the main scope of the theoretical approach.

The maximum work done by the PEM fuel cell, W_{ideal}, can be calculated from knowing the thermodynamic data of the process at temperature T and pressure p. The maximum work is given in Equation (3.6).

$$-W_{ideal} = \Delta_n H - T_0 \Delta_n S \tag{3.6}$$

$\Delta_n H$ is the enthalpy change for the electrochemical reaction at temperature T and pressure p. $\Delta_n S$ is the entropy change for the electrochemical reaction at temperature T and pressure p. T_0 is the temperature of the surroundings. The PEM fuel cell dissipates much of the potential work as heat when a current is drawn from the cell. This lost work is the entropy production times the surrounding temperature T_0, and is given in Equation (3.7).

$$\frac{dW_{lost}}{dt} = T_0 \frac{dS_{irr}}{dt} = T_0 \frac{d}{dt}(\Delta S_0 + \Delta_n S) \tag{3.7}$$

dS_{irr}/dt is the total entropy production in the fuel cell and ΔS_0 is the entropy change in the surroundings. The lost work is zero in the reversible limit, as the entropy production is zero. The lost work varies with the electric current density that is drawn from the cell. Even at moderate current densities, the cell's potential is halved and the power reduced accordingly.

The cell can be divided into five subsystems: anode and cathode electrode surfaces, anode and cathode backing and membrane. A schematic of this system is given in Figure 3.2.1.

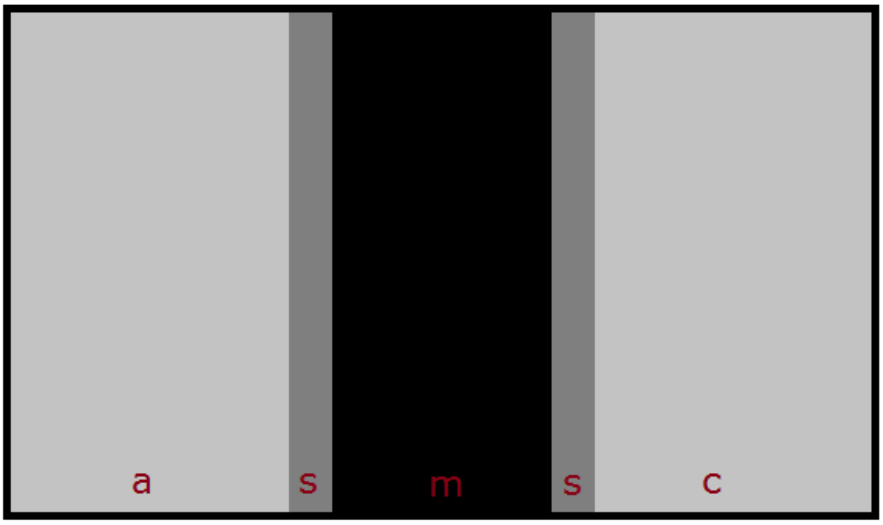

Figure 3.2.1: *Schematic of the fuel cell where a is the Sigracet anode backing, c is the Sigracet cathode backing, m is the Nafion membrane and s are the surfaces. The electrochemical reactions takes place at the surfaces, while the anode and cathode backing have a transport of hydrogen and oxygen to the surface. Protons are transported through the Nafion membrane.*

The total entropy production in the fuel cell can be divided into local entropy production in the three bulk phases and the two surfaces. This local entropy production is given in Equations (3.8), (3.11) and (3.10) for anode, cathode and membrane respectively.

$$\sigma^a = J_q'^a \frac{d}{dx}\left(\frac{1}{T}\right) - J_D^a \frac{1}{T}\frac{d\mu_{w,T}}{dx} - j\frac{1}{T}\frac{d\phi}{dx} \tag{3.8}$$

Where the J_D^a in this equation is a inter-diffusion flux of water and hydrogen, given in Equation (3.9).

$$J_D^a = \left(\frac{J_w}{x_w} - \frac{J_{H_2}}{x_{H_2}}\right) x_w \tag{3.9}$$

There is only a water flux through the membrane:

$$\sigma^m = J_m^{'m} \frac{d}{dx}\left(\frac{1}{T}\right) - J_w^m \frac{1}{T}\frac{d\mu_{w,T}}{dx} - j\frac{1}{T}\frac{d\phi}{dx} \tag{3.10}$$

$$\sigma^c = J_q^{'c} \frac{d}{dx}\left(\frac{1}{T}\right) - J_w^c \frac{1}{T}\frac{d\mu_{w,T}}{dx} - J_{O_2}^c \frac{1}{T}\frac{d\mu_{O_2,T}}{dx} - J_{N_2}^c \frac{1}{T}\frac{d\mu_{N_2,T}}{dx} - j\frac{1}{T}\frac{d\phi}{dx} \tag{3.11}$$

The flux of nitrogen in the cathode backing is zero and the chemical potential gradient of water can be assumed to be zero. The only mass flux that is left is the flux of O_2. The entropy production in the cathode backing for this system can then be expressed as done in Equation (3.12).

$$\sigma^m = J_q^{'} \frac{d}{dx}\left(\frac{1}{T}\right) - J_{O_2}^c \frac{1}{T}\frac{d\mu_{O_2,T}}{dx} - j\frac{1}{T}\frac{d\phi}{dx} \tag{3.12}$$

The linear flux-force relations in the system can be given as done in Equations (3.13), (3.14) and (3.15).

$$J_q^{'} = l_{qq} \frac{d}{dx}\left(\frac{1}{T}\right) - l_{q\mu}\frac{1}{T}\frac{d\mu_T}{dx} - l_{q\phi}\frac{1}{T}\frac{d\phi}{dx} \tag{3.13}$$

$$J = l_{\mu q} \frac{d}{dx}\left(\frac{1}{T}\right) - l_{\mu\mu}\frac{1}{T}\frac{d\mu_T}{dx} - l_{\mu\phi}\frac{1}{T}\frac{d\phi}{dx} \tag{3.14}$$

$$j = l_{\phi q} \frac{d}{dx}\left(\frac{1}{T}\right) - l_{\phi\mu}\frac{1}{T}\frac{d\mu_T}{dx} - l_{\phi\phi}\frac{1}{T}\frac{d\phi}{dx} \tag{3.15}$$

Here J is the mass flux of either oxygen, hydrogen or water and μ_T is either $\mu_{O_2,T}$ or $\mu_{w,T}$. The l_{ij} are phenomenological coefficients. The set of equations can be written by eliminating the electrical potential gradient as done in Equations (3.16), (3.17) and (3.18).

$$J_q^{'} = L_{qq}\frac{d}{dx}\left(\frac{1}{T}\right) - L_{q\mu}\frac{1}{T}\frac{d\mu_T}{dx} + \frac{l_{q\phi}}{l_{\phi\phi}}j \tag{3.16}$$

$$J = L_{\mu q}\frac{d}{dx}\left(\frac{1}{T}\right) - L_{\mu\mu}\frac{1}{T}\frac{d\mu_T}{dx} + \frac{l_{q\phi}}{l_{\phi\phi}}j \tag{3.17}$$

$$j = l_{\phi q}\frac{d}{dx}\left(\frac{1}{T}\right) - l_{\phi\mu}\frac{1}{T}\frac{d\mu_T}{dx} - l_{\phi\phi}\frac{1}{T}\frac{d\phi}{dx} \tag{3.18}$$

The L_{ij} coefficients are related by Equation (3.19).

$$L_{ij} = l_{ij} - \frac{l_{\phi i}l_{j\phi}}{l_{\phi\phi}} \tag{3.19}$$

The transport transference coefficient can be given as done in Equation (3.20).

$$L_{ij} = \left(\frac{J}{j/F}\right)_{d\mu_t=0,dT=0} = F\frac{l_{\mu\phi}}{l_{\phi\phi}} \tag{3.20}$$

The Peltier coefficient is given in Equation (3.21).

$$\pi = \left(\frac{J_q^{'}}{j/F}\right)_{d\mu_t=0,dT=0} = F\frac{l_{q\phi}}{l_{\phi\phi}} \tag{3.21}$$

The measurable heat of transfer is given in Equation (3.22).

$$q^* = \left(\frac{J_q^{'}}{J}\right)_{j=0,dT=0} = \frac{L_{q\mu}}{L_{\mu\mu}} \tag{3.22}$$

By using the definitions of these coefficients into the flux equations in Equations (3.16), (3.17) and (3.18) while introducing the gradient of chemical potential $d\mu_T/dx$, into the heat flux, then these flux equations can be written as done in Equations (3.23), (3.24) and (3.25).

$$\frac{dT}{dx} = -\frac{1}{\lambda}\left(J_q^{'} - q^*\left(J - t\frac{j}{F}\right) - \pi\frac{j}{F}\right) \tag{3.23}$$

$$\frac{d\mu_T}{dx} = -\frac{q^*}{T}\frac{dT}{dx} - \frac{1}{L_{\mu\mu}}\left(J - t\frac{j}{F}\right) \tag{3.24}$$

$$\frac{d\phi}{dx} = -\frac{\pi}{TF}\frac{dT}{dx} - \frac{t}{F}\frac{d\mu_T}{dx} - rj \tag{3.25}$$

Here the thermal conductivity λ, the mass transfer coefficient $L_{\mu\mu}$ and the electrical resistivity r are given in Equations (3.26), (3.27) and (3.28).

$$\lambda = \frac{L_{qq}}{T^2} - (q^*)^2\frac{L_{\mu\mu}}{T^2} \tag{3.26}$$

$$l_{\mu\mu} = \frac{Dx}{RT} \tag{3.27}$$

$$r = \frac{T}{l_{\phi\phi}} \tag{3.28}$$

D is the diffusion constant, x is the mole fraction and R is the universal gas constant.

Each section of the PEM fuel cell is described in detail and the expressions in Equations (3.23), (3.24) and (3.25) will be given for each section. The catalyst surface of anode/membrane and membrane/cathode will be explained in detail after the theory and transport equations for the three bulk phases have been given. The surfaces in this model is treated as shown in Figure 3.2.2, where i is into the surface, and o is out of the surface. For the anode surface, the anode backing is the i index while the membrane is the o index. For the cathode surface, the membrane is the i index while the cathode backing is the o index. In the calculations the jumps between each index and surface is calculated separately, and this will be covered in the later subsection.

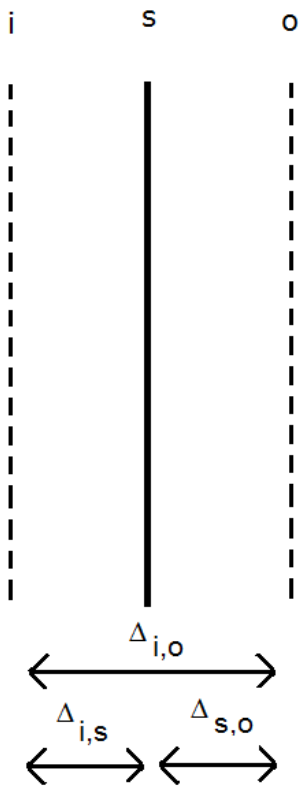

Figure 3.2.2: *The surface is divided and treated as two separated parts, i,s and s,o, where i is the index for into the surface and o is the index for out of the surface.*

3.2.1 Anode backing

The anode backing has transport of heat, water, hydrogen and charge. Hydrogen and water are transported in the pores of the backing at a constant pressure p while the heat and charge are transported in the solid material of the backing. The chemical potential of water, μ_w is given in Equation (3.29).

$$\mu_w = \mu_w^0 + RT \ln \frac{x_w}{x_w^*} \tag{3.29}$$

In this equation x_w is the mole fraction of water, x_w^* is the mole fraction of water at saturation and μ_w^0 is the chemical potential when $x_w = x_w^*$. The mole fraction of water

at saturation, x_w^* is found by p_w^*/p_{pot}. The saturation pressure of water has been found by Mench[7] which is given in Equation (3.30).

$$p_w^* = -2846.4 + 411.24(T - 273.15) - 10.554(T - 273.15)^2 + 0.16636(T - 273.15)^3 \quad (3.30)$$

By introducing Equation (3.29) into Equations (3.23), (3.24) and (3.25), they can be rewritten as shown in Equations (3.31), (3.32) and (3.33).

$$\frac{dT}{dx} = -\frac{1}{\lambda^a}\left(J_q^{'a} - q^{*,a}\left(J_D^a - t_D^a\frac{j}{F}\right) - \pi^a\frac{j}{F}\right) \quad (3.31)$$

$$\frac{dx_w}{dx} = -\frac{q^{*,a}x_w}{RT^2}\frac{dT}{dx} - \frac{1}{D_{wH}}\left(J_D^a - t_D^a\frac{j}{F}\right) \quad (3.32)$$

$$\frac{d\phi}{dx} = -\frac{\pi^a}{TF}\frac{dT}{dx} - \frac{t_D^a RT}{Fx_w}\frac{dx_w}{dx} - r^a j \quad (3.33)$$

The energy balance is given in Equation (3.34).

$$\frac{d}{dx}J_u = \frac{d}{dx}J_q^{'a} + j\frac{d\phi}{dx} + J_{H_2}\frac{dH_{H_2}}{dx} + J_w^a\frac{dH_w}{dx} = 0 \quad (3.34)$$

Here J_u is the constant energy flux through the backing. The dH_i/dx is given in Equation (3.35). It is assumed that the gas component i follow the ideal gas law.

$$\frac{dH_i}{dx} = c_{p,i}\frac{dT}{dx} \quad (3.35)$$

The Peltier coefficient, π^a is given in Equation (3.36). The heat of transfer, $q^{*,a}$, is given in Equation (3.37).

$$\pi^a = -T(0.5S_{H_2} + S_{e^-}^*) \quad (3.36)$$

$$q^{*,a} = -TS_w^a \quad (3.37)$$

3.2.2 Membrane

Inside the membrane there is transport of water, heat and charge. The transport equations over the membrane is given in Equations (3.38), (3.39) and (3.40).

$$\frac{dT}{dx} = -\frac{1}{\lambda^m}\left(J_q^{'m} - q^{*,m}\left(J_w^m - t_w^m\frac{j}{F}\right) - \pi^m\frac{j}{F}\right) \tag{3.38}$$

$$\frac{d\mu_{w,T}}{dx} = -\frac{q^{*,m}}{T}\frac{dT}{dx} - \frac{1}{L_{\mu\mu}^m}\left(J_w^m - t_w^m\frac{j}{F}\right) \tag{3.39}$$

$$\frac{d\phi}{dx} = -\frac{\pi^m}{TF}\frac{dT}{dx} - \frac{t_w^m}{F}\frac{d\mu_{w,T}}{dx} - r^m j \tag{3.40}$$

$L_{\mu\mu}^m$ is related to the concentration of water and diffusion constant through Equation (3.41).

$$L_{\mu\mu}^m = D_w^m\frac{c_w}{RT} \tag{3.41}$$

D_w^m is the diffusion constant of water inside the membrane, and the water concentration, c_w, can be related to the water content χ through Equation (3.42). The water content inside the Nafion membrane is found by Springer et al.[9] to follow Equation (3.43).

$$c_w = \frac{\chi\rho_m}{M_m} \tag{3.42}$$

$$\begin{aligned} \chi &= 0.043 + 17.81a - 39.85a^2 + 36.0a^3, && 0 < a < 1 \\ \chi &= 14 + 1.4(a-1), && 1 \leq a < 3 \\ \chi &= 16.8 && a \geq 3 \end{aligned} \tag{3.43}$$

ρ is the density of the membrane, M_m is the molar mass of the membrane and a_w is the activity of water inside the membrane. This activity is calculated from its definition, $a_w = p_w/p_w^*$. By introducing $d\chi/dx$ with $(d\chi/da_w)(da_w dx)$ into the transport equations, then these differential equations can be written as done in Equation (3.44), (3.45) and (3.46).

$$\frac{dT}{dx} = -\frac{1}{\lambda^m}\left(J_q^{'m} - q^{*,m}\left(J_w^m - t_w^m\frac{j}{F}\right) - \pi^m\frac{j}{F}\right) \tag{3.44}$$

$$\frac{da_w}{dx} = -\frac{q^{*,m}\chi}{(d\chi/da_w)RT}\frac{dT}{dx} - \frac{(J_w^m - t_w^m j/F)M_m}{(d\chi/da_w)\rho D_w^m} \tag{3.45}$$

$$\frac{d\phi}{dx} = -\frac{\pi^m}{TF}\frac{dT}{dx} - \frac{t_w^m RT}{Fa_w}\frac{da_w}{dx} - r^m j \tag{3.46}$$

The electrical resistivity r^m is found by Springer et al.[9] and is given in Equation (3.47).

$$(r^m)^{-1} = exp\left(1268\left(\frac{1}{303} - \frac{1}{T}\right)\right)(0.5139\chi - 0.326) \tag{3.47}$$

The energy conservation in the membrane is given in Equation (3.48).

$$\frac{d}{dx}J_u = \frac{d}{dx}(J_q^{'m} + j\phi^m + J_w^m H_w^m) = 0 \tag{3.48}$$

The Peltier coefficient, π^m is given in Equation (3.49). The heat of transfer, $q^{*,m}$, is given in Equation (3.50).

$$\pi^m = T(S_{H+}^* - t_w^m S_w^m) \tag{3.49}$$

$$q^{*,m} = -TS_w^m \tag{3.50}$$

3.2.3 Cathode backing

In the cathode backing there is transport of oxygen, water, heat and charge. The flux equations for the cathode backing are given in Equations (3.51), (3.52) and (3.53). The coupling between oxygen an other gas fluxes is neglected in these equations.

$$\frac{dT}{dx} = -\frac{1}{\lambda^c}\left(J_q^{'c} - \pi^c\frac{j}{F}\right) \tag{3.51}$$

$$\frac{dx_{O_2}}{dx} = \frac{j}{4FD_{ON}} \tag{3.52}$$

$$\frac{d\phi}{dx} = -\frac{\pi^c}{TF}\frac{dT}{dx} - \frac{RT}{4Fx_{O_2}}\frac{dx_{O_2}}{dx} - r^c j \tag{3.53}$$

The energy conservation in the cathode backing is given in Equation (3.54).

$$\frac{d}{dx}J_u = \frac{d}{dx}(J_q^{'c} + j\phi^c + J_w^c H_w^c + J_{O_2}^c H_{O_2}^c = 0 \tag{3.54}$$

The Peltier coefficient, π^c, and the heat of transfer, $q^{*,c}$, are given in Equations (3.55) and (3.56).

$$\pi^c = T\left(0.25S_{O_2} - S_e^* - \left(\frac{J_w^a}{j/F} + 0.5\right)S_w^c\right) \tag{3.55}$$

$$q^{*,c} = -TS_w^c \tag{3.56}$$

3.2.4 Anode surface

At the anode catalyst surface the enthalpy of hydrogen is converted into energy and heat. Additionally there is a change in the enthalpy of water as it goes from vapor state to condensed state in the membrane which releases heat. The conversion of energy across this phase boundary is given in Equation (3.57)

$$J_u = J_q^{'a} + j\phi^a + J_{H_2}H_{H_2} + J_w^a H_w^a = J_q^{'m} + j\phi^m + J_w^m H_w^m \tag{3.57}$$

The entropy production of the electrode surface between the anode phase and the membrane phase is given in Equation (3.58).

$$\sigma^{a,m} = J_q^{'a}\Delta_{a,s}\frac{1}{T} + J_q^{'m}\Delta_{s,m}\frac{1}{T} - J_w^a\frac{1}{T^s}\Delta_{a,s}\mu_{w,T} - J_w^m\Delta_{s,m}\mu_{w,T} - j\frac{1}{T^s}\left(\Delta_{a,m}\phi + \frac{\Delta_n G^s}{F} + \frac{\Delta_{g,T}}{v_g F}\right) \tag{3.58}$$

By dividing the surface into two segments, (a,s) and (s,m) which are the two segments

shown in Figure 3.2.2, then the equations can be divided into two separate parts as well. The equations of transport that derive from this entropy production is given in Equations (3.59), (3.60), (3.61), (3.62) and (3.63).

$$\Delta_{a,s}T = -\frac{J_q^{',a}}{\lambda_a^s} + \frac{q^{*,a}}{\lambda_a^s}\left(J_w - t_w\frac{j}{F}\right) + \pi^a\frac{j}{\lambda_a^s F} \tag{3.59}$$

$$\Delta_{s,m}T = -\frac{J_q^{',m}}{\lambda_m^s} + \frac{q^{*,m}}{\lambda_m^s}\left(J_w - t_w\frac{j}{F}\right) + \pi^m\frac{j}{\lambda_m^s F} \tag{3.60}$$

$$\Delta_{a,s}\mu_{w,T} = -\frac{q^{*,a}}{T^{a,m}}\Delta_{a,s}T - \frac{J_w - t_w\frac{j}{F}}{L_{\mu\mu}^s} \tag{3.61}$$

$$\Delta_{s,m}\mu_{w,T} = -\frac{q^{*,m}}{T^{m,a}}\Delta_{s,m}T - \frac{J_w - t_w\frac{j}{F}}{L_{\mu\mu}^s} \tag{3.62}$$

$$\Delta\phi_{eff} = -\frac{\pi^a}{T^{a,m}F}\Delta_{a,s}T - \frac{\pi^m}{T^{m,a}F}\Delta_{s,m}T - \frac{t_w}{F}\Delta_{a,m}\mu_{w,T} - r^s j \tag{3.63}$$

The electrochemical reaction takes place at the surface, and the reaction at stationary state is equal to jF. One can assume that there is equilibrium for adsorption of hydrogen at the surface. This means that the reaction of Gibbs energy in the surface can be expressed by thermodynamic properties. This way of expression the Gibbs energy along with the relation between $\Delta_{a,m}\phi_{eff}$ and $\Delta_{a,m}\phi$ are given in Equations (3.64), (3.65) and (3.66). The overpotential in the anode surface is neglected.

$$\Delta_{a,m}\phi_{eff} = \Delta_{a,m}\phi + \frac{\Delta_n G^{s,a}}{F} + \frac{\Delta\mu_{H_2,T}(T^{s,a})}{2F} \tag{3.64}$$

$$\Delta_{a,m}\phi_{eff} = \Delta_{a,m}\phi - \frac{1}{2F}(\mu_{H_2}^s - \Delta_{a,s}\mu_{H_2,T}(T^{s,a})) \tag{3.65}$$

$$\Delta_{a,m}\phi_{eff} = \Delta_{a,m}\phi - \frac{1}{2F}(H_{H_2} - T^{s,a}S_{H_2}) \tag{3.66}$$

3.2.5 Cathode surface

At the cathode catalyst surface, oxygen reacts with the protons and forms water. The energy conservation at this surface is given in Equation (3.67).

$$J_u = J_q^{'m} + j\phi^m + J_w^m H_w^m = J_q^{'c} + j\phi^c + J_w^c H_w^c + J_{O_2} H_{O_2} \tag{3.67}$$

The entropy production of the electrode surface between the cathode phase and the membrane phase is given in Equation (3.68).

$$\sigma^{m,c} = J_q^{'m}\Delta_{m,s}\frac{1}{T} + J_q^{'c}\Delta_{s,c}\frac{1}{T} - J_w^m\frac{1}{T^s}\Delta_{m,s}\mu_{w,T} - J_w^c\Delta_{s,c}\mu_{w,T} - j\frac{1}{T^s}\left(\Delta_{m,c}\phi + \frac{\Delta_n G^s}{F} + \frac{\Delta_{g,T}}{v_g F}\right) \tag{3.68}$$

By dividing the surface into two segments, (m,s) and (s,c) which are the two segments shown in Figure 3.2.2, then the equations can be divided into two separate parts as well. The equations of transport that derive from this entropy production is given in Equations (3.69), (3.70), (3.71), (3.72) and (3.73).

$$\Delta_{m,s}T = -\frac{J_q^{'m}}{\lambda_m^s} + \frac{q^{*,m}}{\lambda_m^s}\left(J_w^m - t_w^m\frac{j}{F}\right) + \pi^m\frac{j}{\lambda_m^s F} \tag{3.69}$$

$$\Delta_{s,c}T = -\frac{J_q^{'c}}{\lambda_c^s} + \frac{q^{*,c}}{\lambda_c^s}\left(J_w^c - t_w^c\frac{j}{F}\right) + \pi^c\frac{j}{\lambda_c^s F} \tag{3.70}$$

$$\Delta_{m,s}\mu_{w,T} = -\frac{q^{*,m}}{T^{m,c}}\Delta_{m,s}T - \frac{J_w^m - t_w^m\frac{j}{F}}{L_{\mu\mu}^m} \tag{3.71}$$

$$\Delta_{s,c}\mu_{w,T} = -\frac{q^{*,c}}{T^{c,m}}\Delta_{s,c}T - \frac{J_w^c - t_w^c\frac{j}{F}}{L_{\mu\mu}^s} \tag{3.72}$$

$$\Delta\phi_{eff} = \frac{1}{F}\left(-\frac{\pi^m}{T^{m,c}}\Delta_{m,s}T - \frac{\pi^c}{T^{c,m}}\Delta_{s,c}T - t_w^c\Delta_{m,s}\mu_{w,T} - t_w^c\Delta_{s,c}\mu_{w,T} - r^s j\right) - \eta^c \tag{3.73}$$

In the last equation there is an overpotential, η^c, that comes from the resistance of the

activated electrochemical reaction taking place at the oxygen electrode. The value of the cathode overpotential was determined by Vie and Kjelstrup [2] and is given in Equation (3.74).

$$\eta^c = \frac{2RT}{F} \ln \frac{j}{j^0} \tag{3.74}$$

The exchange current density for oxygen in air is $j^0 = 2.5 \cdot 10^{-3}$ A/m^2. It is assumed that there is equilibrium for the adsorption of oxygen at the surface. The reaction Gibbs energy in the surface can then be expressed by thermodynamics properties of the gas. This expression is given along with the relation between $\Delta_{m,c}\phi_{eff}$ and $\Delta_{m,c}\phi$ in Equations (3.75) and (3.76).

$$\Delta_{m,c}\phi_{eff} = \Delta_{m,c}\phi + \frac{Delta_n G^{s,c}}{F} + \frac{\Delta\mu_{O_2,T}(T^{s,c})}{4F} \tag{3.75}$$

$$\Delta_{m,c}\phi_{eff} = \Delta_{m,c}\phi + \frac{1}{2F}(H_w^c - T^{s,c}S_w^c) - \frac{1}{4F}(H_{O_2} - T^{s,c}S_{O_2}) \tag{3.76}$$

3.2.6 Total energy and entropy balance

The total energy balance in the system can be given as the energy coming into the anode backing and out of the cathode backing. This balance is given in Equation (3.77).

$$J_q^{'a} + J_{H_2}H_{H_2} + J_w^a H_w^a + j\phi^a = J_q^{'c} + J_{O_2}H_{O_2} + J_w^c H_w^c + j\phi^c \tag{3.77}$$

The entropy production for a unit cross-sectional area of the cell can be found by integrating over the local entropy contributions from all the five subsystems, given in Equation (3.78).

$$\frac{dS_{irr}}{dt} = \Omega \left(\int_{0,a}^{a,m} \sigma^a dx + \sigma^{s,a} + \int_{a,m}^{m,c} \sigma^m dx + \sigma^{s,c} + \int_{c,m}^{0,c} \sigma^c dx \right) \tag{3.78}$$

In this equation, Ω is the cross sectional area of the cell. The entropy production has to be larger or equal to zero, $dS_{irr}/dt \geq 0$, otherwise it will be thermodynamically impossible,

as it violates the second law of thermodynamics, rendering the model useless. The total entropy balance for the whole cell, which is an equivalent way of finding the entropy production, is given in Equation (3.79).

$$\frac{dS_{irr}}{dt} = (J_s^c - J_s^a)\Omega \tag{3.79}$$

The entropy flux J_s is given in Equation (3.80).

$$J_s = \frac{J_q^{'}}{T} + \sum_i J_i S_i \tag{3.80}$$

S_i is the partial molar entropy of component i and J_i is the mass flux of component i. By inserting these fluxes into the total entropy production gives Equation (3.81) and can be rewritten as Equation (3.82) using knowledge about the mass fluxes.

$$\frac{1}{\Omega}\frac{dS_{irr}}{dt} = \left(\frac{J_q^{',c}}{T^c} + J_w^c S_w^c + J_{O_2} S_{O_2}\right) - \left(\frac{J_q^{',a}}{T^a} + J_w^a S_w^a + J_{H_2} S_{H_2}\right) \tag{3.81}$$

$$\frac{1}{\Omega}\frac{dS_{irr}}{dt} = \frac{J_q^{',c}}{T^c} - \frac{J_q^{',a}}{T^a} + \frac{j}{F}\left(\frac{1}{2}(S_w^c - S_{H_2}) - \frac{1}{4}S_{O_2}\right) + J_w^a(S_w^c - S_w^a) \tag{3.82}$$

3.3 System properties

3.3.1 General parameters

Through the simulation a set of general constants is used. These constants are the universal gas constant, R, which is equal to 8.314 J/K mol[5] and Faradays constant, F, equal to 96500 C/mol[5]. These parameters are given in Table 3.3.1.

Table 3.3.1: *General constants used in the simulation.*

Property	Value
R	8.314 J/K mol[5]
F	96500 C/mol[5]

3.3.2 Properties of anode and cathode backing

The thickness of the anode and cathode backing in this simulation is 246 μm, to reflect the similar system size for the membrane system in part 2. The pressure of hydrogen, p_{H_2} in the anode backing is set to 1 bar in the anode backing, while the pressure of oxygen, p_{O_2} is set to 0.21 bar in the cathode backing. Similar, the mole fraction of oxygen in the end of the cathode backing, x_{O_2} is set to 0.21, which is the mole fraction of oxygen in air. The temperature at the start of the anode backing, T^a, is set to be 330 K. The material properties of the anode and cathode backing is that of Sigracet GDL10AA. The heat conductivity for this Sigracet type is $\lambda^a = \lambda^c = 0.42$ W/m K[10]. The effective binary diffusion constants used is $D_{wH} = D_{ON}$, is set as $5{\cdot}10^{-5}$ m^2/s.[5]. The electrical resistance, $r^a = r^c$ is set as 10^{-4} ohm m[8]. The combined water and hydrogen transference number t_D^a is set as 0. These parameters are given in Table 3.3.2.

Table 3.3.2: *Material properties of the anode and cathode backing.*

Property	Value
d_a	246 μm
d_c	246 μm
$\lambda^a(GDL10AA)$	0.42 W/mK[10]
$\lambda^c(GDL10AA)$	0.42 W/mK[10]
D_{wH}	$5{\cdot}10^{-5}$ m^2/s[5]
D_{ON}	$5{\cdot}10^{-5}$ m^2/s[5]
r^a	10^{-4} ohm m[8]
r^c	10^{-4} ohm m[8]
t_D^a	0

3.3.3 Properties of anode and cathode surface.

The heat conductivity in both surfaces, $\lambda_a^s = \lambda_c^s$, are calculated by dividing the heat conductivity for the anode and cathode backing by the thickness. This is equal to $\lambda_a^s = \lambda_c^s$ = 1700 W/m^2 K. Vie[12] determined the thermal conductivity of the catalyst surface to be equal to 1000 W/K m^2 for a similar model, but without coupling terms. This experimental value is in close agreement with the estimated value used in this simulation and provide some validity to the thermal conductivity used for the surface. The electrical resistance, $r_s^a = r_s^c$, is set as $7.2{\cdot}10^{-6}$ ohm m^2[5]. It is assumed that the temperature difference from the anode surface and into the membrane, $\Delta_{s,m}T$ is negligible. The temperature difference between the membrane and the cathode surface, $\Delta_{m,s}$T is also neglected. It is also assumed that there is equilibrium of water over the surfaces, which results in a neglect of the difference in chemical potential over the surfaces. This simplifies the calculation and the need of estimating properties at the surface. All parameters for the surfaces are given in Table 3.3.3.

Table 3.3.3: *Material properties of the anode and cathode surface.*

Property	Value
$\lambda_s^a(GDL10AA)$	1700 W/m^2 K
$\lambda_s^c(GDL10AA)$	1700 W/m^2 K
r_s^a	$7.2 \cdot 10^{-6}$ ohm m^2[5]
r_s^c	$7.2 \cdot 10^{-6}$ ohm m^2[5]
$\Delta_{s,m}T$	0 K
$\Delta_{m,s}T$	0 K
$\Delta_{a,m}\mu_w$	0 J/mol
$\Delta_{m,c}\mu_w$	0 J/mol

3.3.4 Properties of Nafion membrane

The properties of the membrane is that of the Nafion membrane. The thickness of the membrane in this simulation is 192 μm, to reflect the similar system size for the membrane system in part 2. The thermal conductivity, λ_m, is found to be varying with the water content inside the Nafion membrane by Burheim et al. [11]. They found the thermal conductivity to vary as $\lambda_m = (0.177 + 3.7 \cdot 10^{-3}\chi)$ W/m K, where χ is the water content per sulphonic group. The diffusion of water inside the Nafion membrane is found by Villaluenga[13], D_m, is 8.05 $\cdot 10^{-8}$ m^2/s. The density of the dry membrane, ρ_m is 1640 kg/m^3[5]. The molar mass of the membrane, M_m, and water, M_w, is 1.1 kg/mol[5] and 0.018 kg/mol respectively[5]. These parameters are given in Table 3.3.4.

Table 3.3.4: *Material properties of the Nafion membrane.*

Property	Value
d_m	192 μm
λ_m	$0.177 + 3.7 \cdot 10^{-3}\chi$[11]
D_m	$8.05 \cdot 10^{-8}$ m^2/s[13]
ρ_m	1640 kg/m^3[5]
M_m	1.1 kg/mol[5]
M_w	0.018 kg/mol[5]
t_w^m	1.2[5]

3.3.5 Enthalpy and entropy values

The entropy values used in this simulation is calculated using Equation (3.83), while the entropy is calculated from using Equation (3.84). It is assumed that the components follow the ideal gas law.

$$H_i^j = \Delta_f H_i^0 + c_{p_i}(T^j - T^0) \tag{3.83}$$

$$S_i^j = S_i^0 + c_{p_i} \ln \frac{T^j}{T^0} - R \ln \frac{p}{p^0} \tag{3.84}$$

$\Delta_f H_i^0$ is the standard formation enthalpy of component i, H_i^j is the partial molar enthalpy of component i in position j, S_i^0 is the standard entropy of component i, S_i^j is the partial molar entropy of component i in position j and c_{p_i} is the heat capacity of component i. T^j is the temperature at position j and T^0 is the standard temperature, equal to 298 K, p is the pressure and p^0 is the standard pressure of 1 bar. These are the standard enthalpies used for the calculation[5]: $\Delta_f H_{H_2}^0$= 0 kJ/mol, $\Delta_f H_{O_2}^0$= 0 kJ/mol, $\Delta_f H_{H_2O(l)}^0$= -285 kJ/mol and $\Delta_f H_{H_2O(g)}^0$= -242 kJ/mol. These are the standard entropies used for the calculation: $S_{H_2}^0$= 131 J/K mol, $S_{O_2}^0$= 205 J/K mol, $S_{H_2O(l)}^0$= 70 J/K mol, $S_{H_2O(g)}^0$= 189 J/K mol, S_{H^+}= 192 J/K mol[14]. The transported entropy of the electrons in carbon, S_{e^-}, has been found by Hansen et al.[15] to be close to -2 J/K mol for the temperature used in this simulation. It is assumed that the heat capacity

values for each component are independent of the temperature. These are the heat capacities used for the calculation: $c_{p_{H_2}}$= 29 J/K mol, $c_{p_{O_2}}$= 29 J/K mol, $c^l_{p_{H_2O}}$= 75 J/K mol, $c^g_{p_{H_2O}}$= 34 J/K mol and $c_{p_{H^+}}$= 21 J/K mol. All the values are given in Table 3.3.5.

Table 3.3.5: *Standard enthalpy, entropy and heat capacity values for T=298 K.*

Component i	H_i / kJ/mol	S_i / J/Kmol	c_{p_i} / J/Kmol
$H_2O(g)$	-242	189	34
$H_2O(l)$	-285	70	75
H_2	0	131	29
O_2	0	205	29
H^+	-	192	21
e^-	-	-2	-

3.3.6 Fluxes

The current density, j, is a constant value through the system. This value can be changed accordingly to the users wish, and it is set as 200 A/m^2, 500 A/m^2, 1000 A/m^2 and 5000 A/m^2 in this experiment. The measurable heat flux, $J_q^{'}$ is changing through out the whole system, and is dependable on the system settings overall. The hydrogen flux and oxygen flux are constant in the anode and cathode backing, and originate from the current density at stationary state. These fluxes can be calculated by the use of Equations (3.85) and (3.86) for hydrogen and oxygen respectively.

$$J_{H_2} = \frac{j}{2F} \tag{3.85}$$

$$J_{O_2} = -\frac{j}{4F} \tag{3.86}$$

The water flux in the anode backing and the membrane is assumed to be equal, $J^a_w = J^m_w$, while the water flux in the cathode is given from the total water balance in the cell in Equation (3.87). It is assumed that these water fluxes are constant in the fuel cell.

$$J_w^c = J_w^a + \frac{j}{2F} \tag{3.87}$$

The oxygen and hydrogen flux are given in Table 3.3.6 for the corresponding current densities used in the simulation. The water fluxes are dependent on the system as a whole, and can not be calculated directly. A section will show how this is done in the simulation.

Table 3.3.6: *Fluxes of hydrogen and oxygen for various current density.*

J_i	j / A/m^2	Flux value / mol/m^2 s
J_{H_2}	200	0.0010
J_{H_2}	500	0.0026
J_{H_2}	1000	0.0052
J_{H_2}	5000	0.026
J_{O_2}	200	- 0.00052
J_{O_2}	500	- 0.0013
J_{O_2}	1000	- 0.0026
J_{O_2}	5000	- 0.013

3.3.7 Boundary conditions

The differential equations requires a set of boundary conditions for them to be solvable. These boundary conditions are all at the start of the anode backing, and these are the temperature, mole fraction of water, measurable heat flux, electrical potential and entropy production. The temperature is a choice that can be changed accordingly to what is more appropriate for the real system being simulated. For this simulation a temperature of 330 K is used. The mole fraction of water is defined as the ratio of saturated water pressure and the total pressure, where the total pressure in the anode backing is set as 1 bar. The saturated water pressure is determined by the temperature, so the choice of the temperature also affects the boundary condition of mole fraction of water. The electrical potential is set as 0 V after definition of the electrochemical reaction at the anode surface. Since this is the start of the system then there is no accumulated entropy production and this is thus set as 0 W/K m^2. The boundary condition

of measurable heat flux is determined by the total energy balance, but there is an initial guess of 1000 W/m^2. The correct boundary condition for the measurable heat flux is then given after the total energy balance has been calculated. The boundary conditions for the different current densities are given in Table 3.3.7, where the boundary condition for the measurable heat flux is taken from the MatLab simulation after iterating over the total energy balance in the cell.

Table 3.3.7: *Boundary conditions at the anode for various current densities.*

Boundary condition quantity	j / A/m^2	Value
T	200	330 K
T	500	330 K
T	1000	330 K
T	5000	330 K
x_w	200	0.168
x_w	500	0.168
x_w	1000	0.168
x_w	5000	0.168
$J_q^{'}$	200	703.2 W/m^2
$J_q^{'}$	500	1532 W/m^2
$J_q^{'}$	1000	2532 W/m^2
$J_q^{'}$	5000	5026 W/m^2
ϕ	200	0 V
ϕ	500	0 V
ϕ	1000	0 V
ϕ	5000	0 V
σ	200	0 W/K m^2
σ	500	0 W/K m^2
σ	1000	0 W/K m^2
σ	5000	0 W/K m^2

3.4 Solution procedure

3.4.1 Transport equations, boundary conditions and initial settings

All calculations are done in the software program Matrix Laboratory (MatLab), see Appendix F for the complete code of the simulation. The subroutines Jw.m and activity.m, and the method to find the water activity at the start of the Nafion membrane, are made with permission from the work of Kjelstrup and Røsjorde [1,14]. The main program master3.m stores the constants which are used in the calculation along with calling on the subroutine that is doing the main calculation and feeding it with input parameters. The subroutine that is doing the main calculation is called Transport.m.

Each segment; anode backing, membrane and cathode backing, has three differential equations and one energy balance. This makes it possible to have four independent unknowns in the system, which in this case is temperature, mole fraction or chemical potential, measurable heat flux and electrical potential. The entropy production depends on the differential equations of the temperature, mole fraction and electrical potential, and can be found after they have been solved.

Each segment is an independent subroutine which is called on by the subroutine transport.m to calculate these four independent unknowns as well as the entropy production through the whole segment. The differential equations and the energy balance give the derivatives of these five quantities in one direction, x-axis of the cell, and it is therefore needed to use a differential solver to get the values at each point on the x-axis. This is done in MatLab, which has a in built function called ode15s. ode15s is an ordinary differential equation subroutine that is designed specifically to deal with stiff differential systems of equations. This subroutine requires an input, one for each differential equation that is being solved. This works as the boundary conditions in the segment that is being calculated. For instance, the input parameters of the ode solver for the anode backing is the start parameters which are defined for this given system. Each of the four quantities need one boundary condition. For the anode backing this is $T = 330$ K, $x_w = p_w^*/p_{tot}$, $J_q^{'a} = J_q^{'a0}$ and $\phi^a = 0$ V. The measurable heat flux is not known when the simulation starts, so this value has to be estimated in the beginning. $J_q^{'a0} = 1000$ W/m^2 is used as an initial estimation. The correct boundary value of the measurable heat flux is found by iterating the whole calculation with the total energy balance over

the cell. The entropy production requires a boundary condition as well, and this is 0 W/m^2 K as it is in the start of the system.

3.4.2 Finding the correct water flux

The simulation model starts with finding the water flux in the system. To find this water flux an estimation of the value of the water flux is needed along with the measurable heat flux in the anode backing. First the heat flux equation through the anode backing is integrated to find the temperature at the end of the anode backing. Then the saturation pressure of water gas is calculated for this temperature using Equation (3.30). Then the mole fraction of water at the end of the anode backing is found by integrating the Stefan-Maxwell equation through the anode backing. With this mole fraction the activity of water is calculated at the end of the anode backing. Using this water activity gives the water content from Equation (3.43). The water activity at the end of the membrane is known to be $a_w = 1$, which gives a water content of 14 molecules of H_2O per sulphonic group.[9]. It is then possible to calculate the water flux from the difference in water content. This routine now ends with a new value for the water flux, and the whole process can repeated. The correct water flux is found by using the embedded fsolve function in MatLab, which iterates the subroutine until the difference between the previous water flux result and the newest water flux result is close to zero. When this iteration is done then the correct water flux, corresponding to the input parameter of the measurable heat flux, is found. The effect from the temperature difference is neglected while finding the correct water flux. This will effect the results somewhat, especially for higher current densities, such that the water content inside the membrane can be slightly lower than the real value for higher current densities. It is assumed that this assumption is valid enough to get satisfactorily accurate results none the less.

3.4.3 Local and total entropy production

When the correct measurable heat flux is found by iterating the program, and the corresponding water flux is found, then all subroutines of the segments are called to calculate

the temperature, mole fraction or chemical potential, measurable heat flux and electrical potential profiles. With the correct profiles through the cell then the local entropy production can be found in each segment by using the equations in the theory section. The local entropy production in the surfaces is found by using the equations for the respective surface directly as these are not differential equations. An integration of the local entropy production over the anode backing, membrane and cathode backing is needed to find the accumulated entropy production. To do this the subroutine ode15s is once again used to solve these ordinary differential equations and give the local entropy production at each point along the x-axis of the segment. A summation of the integrated local entropy productions from the segments along with the local entropy productions in the surfaces should be equal to the calculated entropy production from the total entropy production in the cell. This is a good test for the validity of the model and the simulation. The profiles for the local entropy productions are calculated from the differential equations using the ode15s solver, while the sum of the local entropy productions given in the results is calculated by using trapezoidal numerical integration. This method is considered more accurate when trying to find the sum of differentials, while the ode15s solver may give less accuracy due to having less numerical points. However, the trapezoidal numerical integration is not able to find profiles from a differential equation, which is why the ode15s solver is used. It is expected to be a small variation in numerical value for the endpoint in the cathode backing and the numerical result from the trapezoidal numerical integration.

3.5 Results

The results are divided into each specific case and given in their own subsection. For a complete set of results for the four main cases with different current densities, see Appendix D for figures.

3.5.1 Water flux

The water flux is found by using the procedure described in the section for solution procedure. The water fluxes of the anode and the cathode backing are given in Table 3.5.1 for the different current densities. The effective water transport number in the anode and the cathode backing is given in Figures 3.5.1 and 3.5.2 respectively.

Table 3.5.1: *Water flux of the anode and cathode backing for various current densities.*

Current density / A/m^2	J_w^a / mol/m^2 s	J_w^c / mol/m^2 s
200	$7.7 \cdot 10^{-4}$	0.0018
500	0.0017	0.0043
1000	0.0027	0.0079
5000	0.0059	0.032

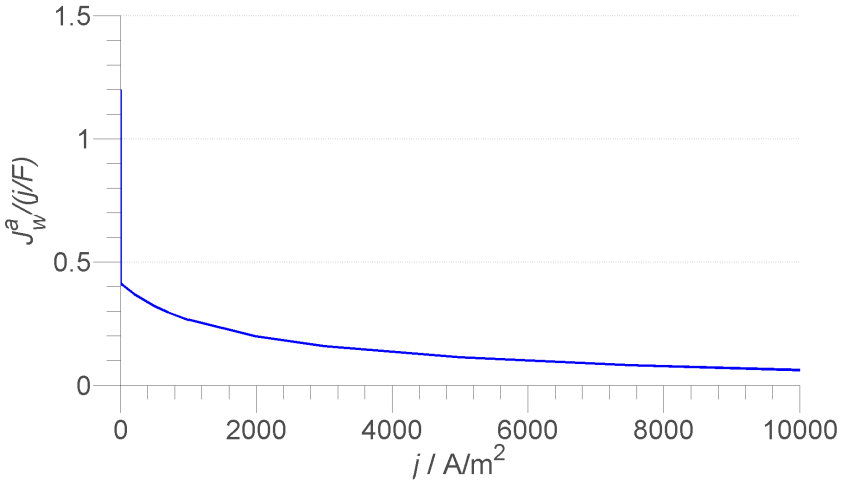

Figure 3.5.1: *Effective water transport number in the anode backing $J_w^a/(j/F)$ plotted for various current densities.*

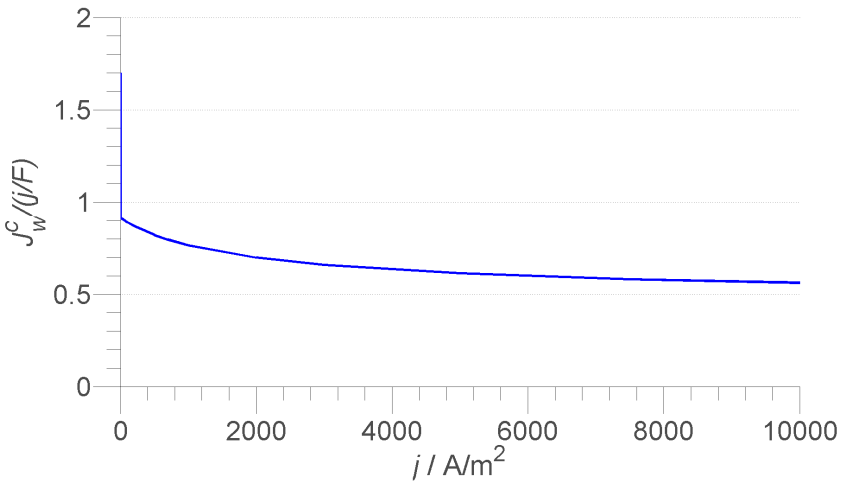

Figure 3.5.2: *Effective water transport number in the cathode backing $J_w^c/(j/F)$ plotted for various current densities.*

3.5.2 Heat flux

As a validity check of the model the measurable heat flux divided by j/F is calculated for all varying current densities. This is done for the measurable heat flux at the start of the anode backing and at the end of the cathode backing. These results are given in Figures 3.5.3 and 3.5.4 for the anode backing and cathode backing respectively.

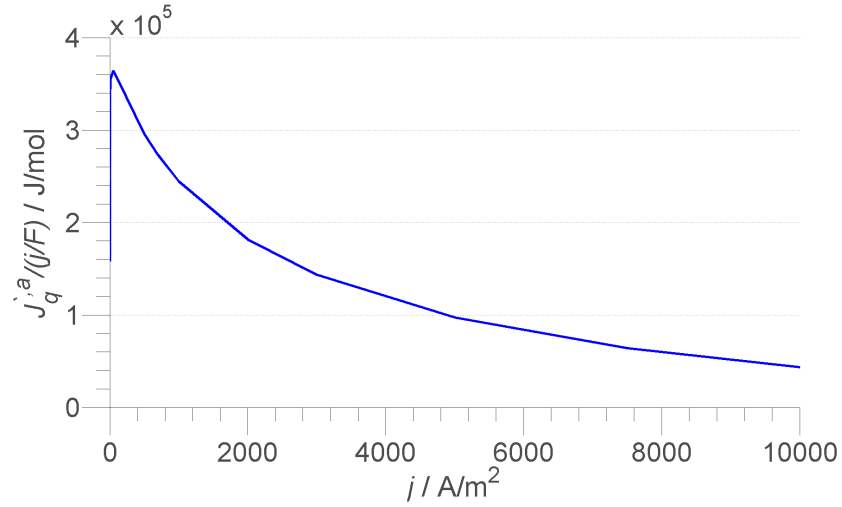

Figure 3.5.3: *Measurable heat flux at the start of the anode backing dividing by j/F for various current densities.*

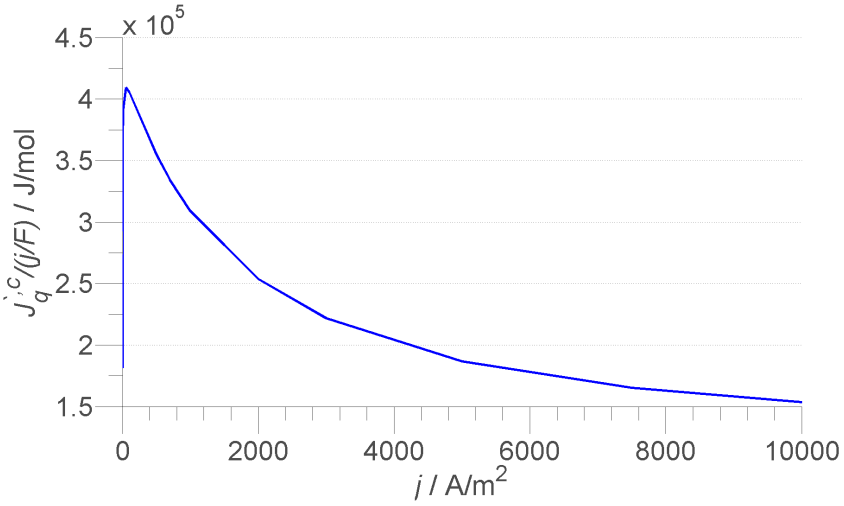

Figure 3.5.4: *Measurable heat flux at the end of the cathode backing dividing by j/F for various current densities.*

3.5.3 Limit when the current density goes to 0 A/m^2 - reversible case

A special case when j $\rightarrow$ 0 A/m^2 was done to see if the results match results for a reversible electrochemical cell. This is a validity check for the model, and matching results is a good result which makes the model more trustworthy. The difference in enthalpy, entropy and Gibbs free energy for the reaction in Equation (3.3) along with the standard cell potential are given in Table 3.5.2. Thermodynamic data of the components, along with Equations (3.83) and (3.84), are used to calculate the enthalpy and entropy differences in this cell reaction. Equations (3.5) and (3.4) are used to calculate the Gibbs free energy and standard cell potential respectively. The temperature difference in the cell in this special case is neglected, as can be seen in Figure 3.5.5, and the temperature is thus set as 330 K for all calculations. The electrical potential profile for this special case is given in Figure 3.5.6. The measurable heat flux profile is given in Figure 3.5.7. The accumulated entropy production is given in Figure 3.5.8.

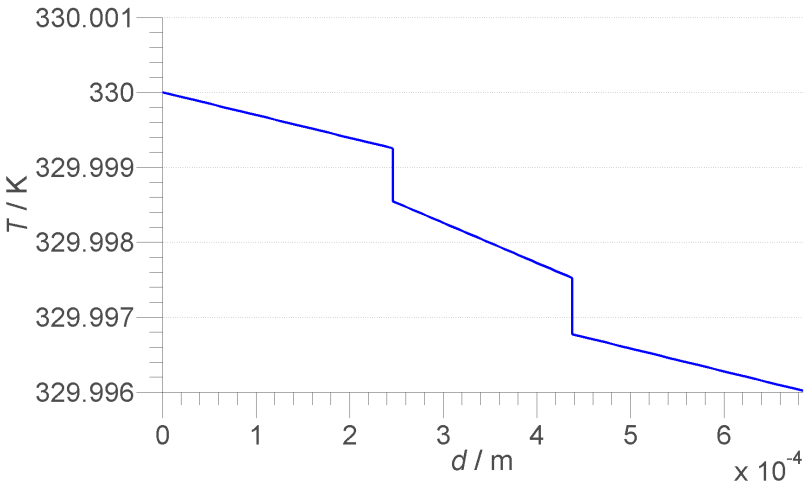

Figure 3.5.5: *Temperature profile in the cell when* $j \to 0$ $\mathrm{A/m^2}$.

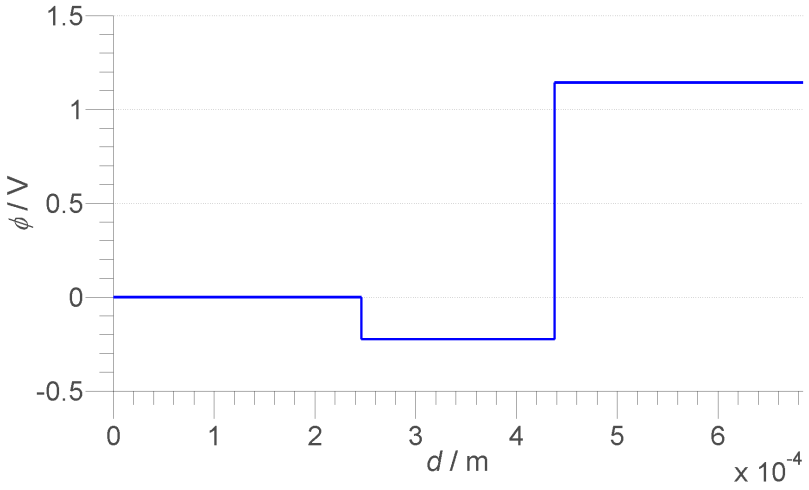

Figure 3.5.6: *Electrical potential profile in the cell when* $j \to 0$ $\mathrm{A/m^2}$.

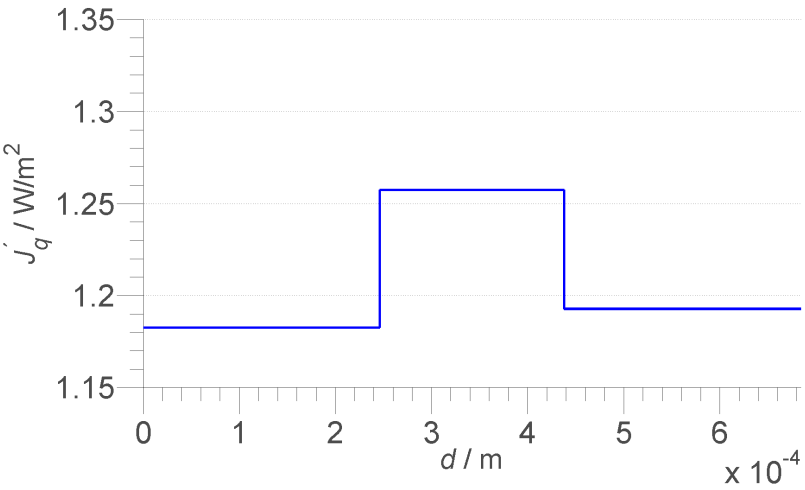

Figure 3.5.7: *Measurable heat flux profile in the cell when* $j \to 0$ A/m^2. *Heat is going into the anode backing and out of the cathode backing.*

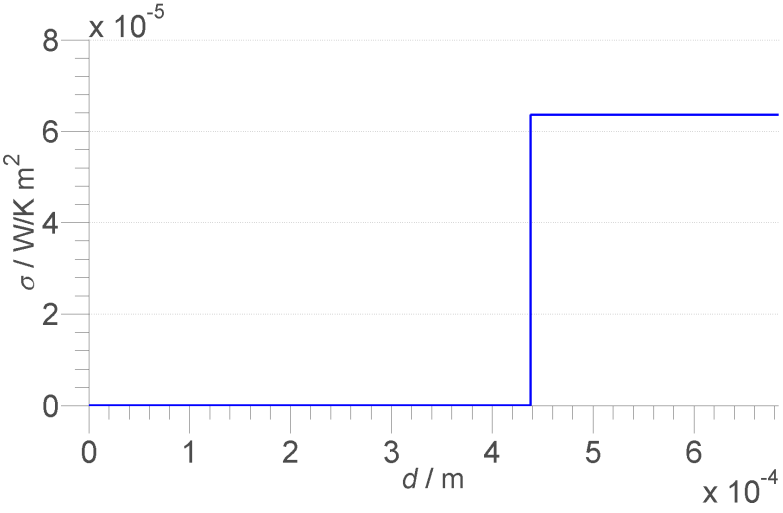

Figure 3.5.8: *Accumulated entropy profile in the cell when* $j \to 0$ A/m^2.

Table 3.5.2: *$\Delta_n H$, $\Delta_n S$, $\Delta_n G$ and ΔV for T=330 K for the reversible case when the current density goes to zero.*

Thermodynamic quantity	Value
$\Delta_n H$	-241 kJ/mol
$\Delta_n S$	-26 J/K mol
$\Delta_n G$	-113 kJ/mol
E^0_{cell}	1.17 V

3.5.4 Results for the fuel cell for j = 200 A/m^2

The profiles in each segment of the fuel cell for temperature, composition, electrical potential and heat flux are given in Figures 3.5.9, 3.5.10, 3.5.11 and 3.5.12 respectively for j = 200 A/m^2. The water content inside the membrane is given in Figure 3.5.13. The accumulated entropy production profile through the cell is given in Figure 3.5.14.

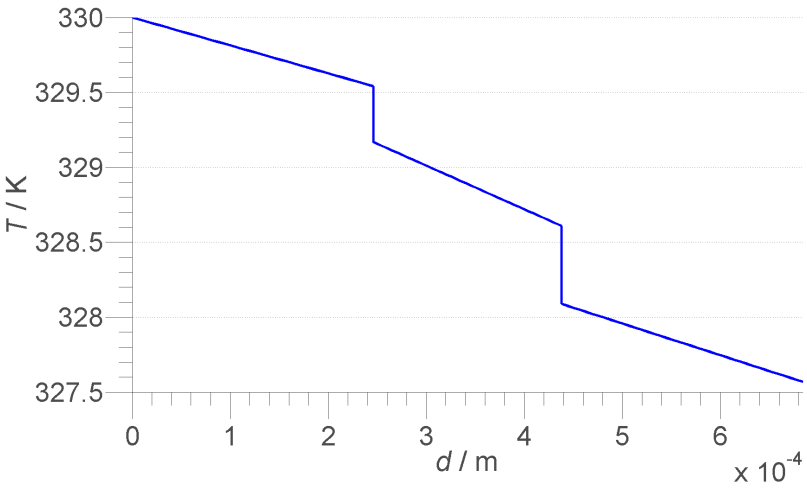

Figure 3.5.9: *Temperature profile in the cell for j = 200* A/m^2*, where T=330 K at the start of the anode backing is set.*

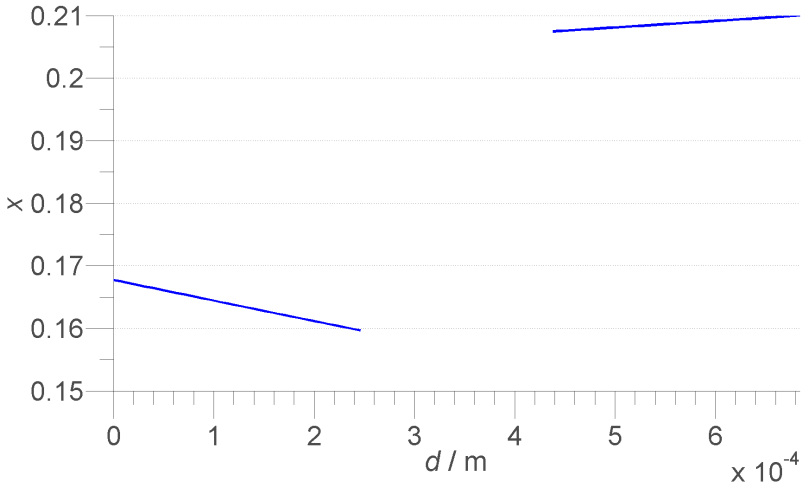

Figure 3.5.10: *Mole fraction profile in the cell for j = 200* A/m^2*, for water at the left side and oxygen at the right side.*

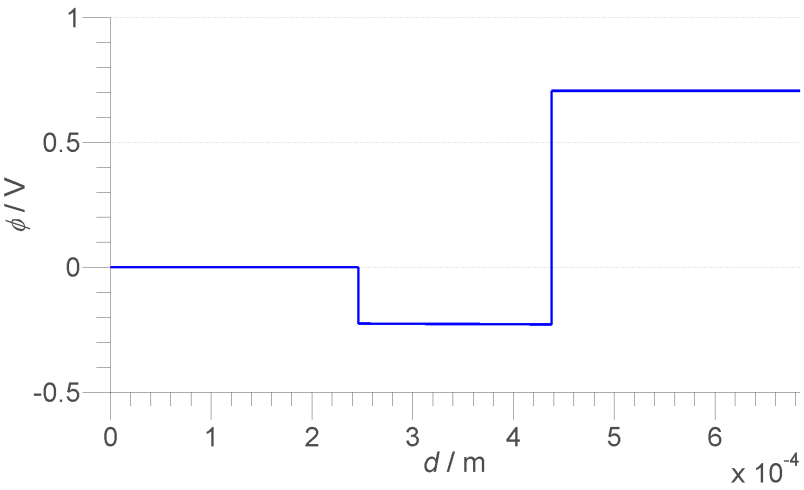

Figure 3.5.11: *Electrical potential profile in the cell for* $j = 200$ $\mathrm{A/m^2}$, $\phi^a = 0$ V *at the start of the anode backing due to definition.*

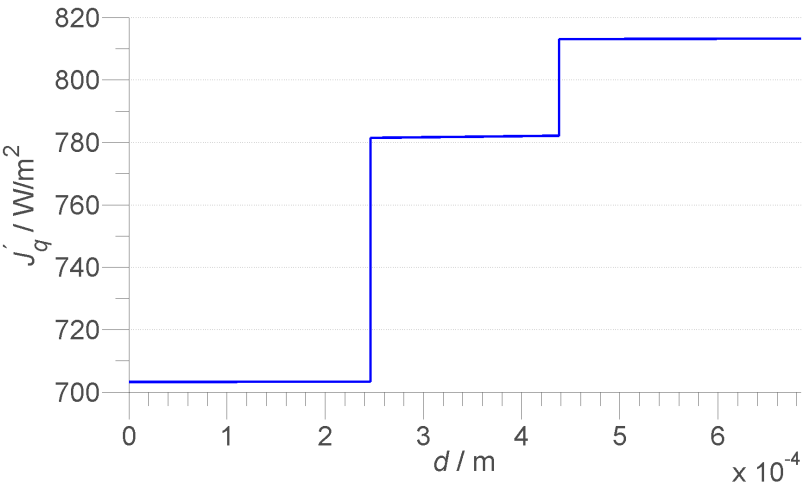

Figure 3.5.12: *Measurable heat flux profile in the cell for* $j = 200$ $\mathrm{A/m^2}$. *Heat is going into the anode backing and out of the cathode backing.*

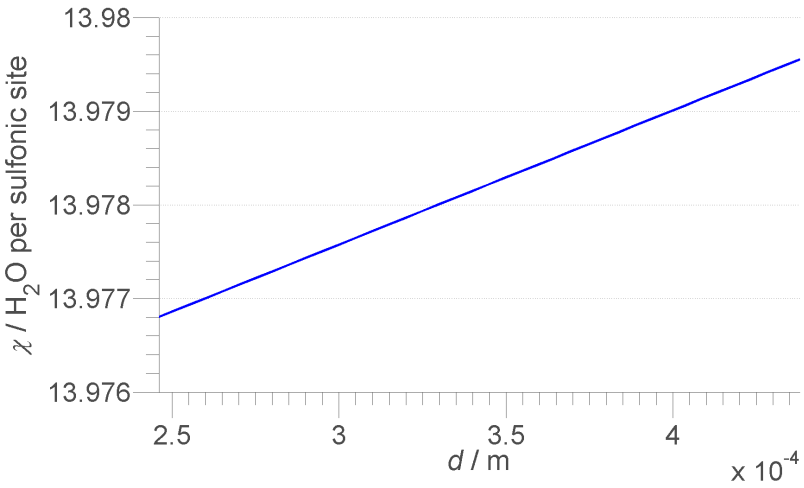

Figure 3.5.13: *Water content profile in the cell for* $j = 200$ $\mathrm{A/m^2}$.

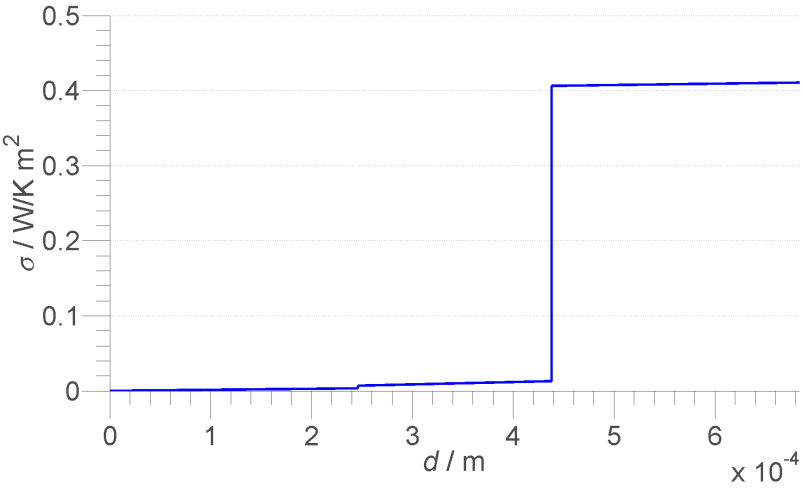

Figure 3.5.14: *Accumulated entropy production profile for* $j = 200$ $\mathrm{A/m^2}$.

3.5.5 Results for the fuel cell for j = 500 A/m^2

The profiles in each segment of the fuel cell for temperature, composition, electrical potential and heat flux are given in Figures 3.5.15, 3.5.16, 3.5.17 and 3.5.18 respectively for j = 500 A/m^2. The water content inside the membrane is given in Figure 3.5.19. The accumulated entropy production profile through the cell is given in Figure 3.5.20.

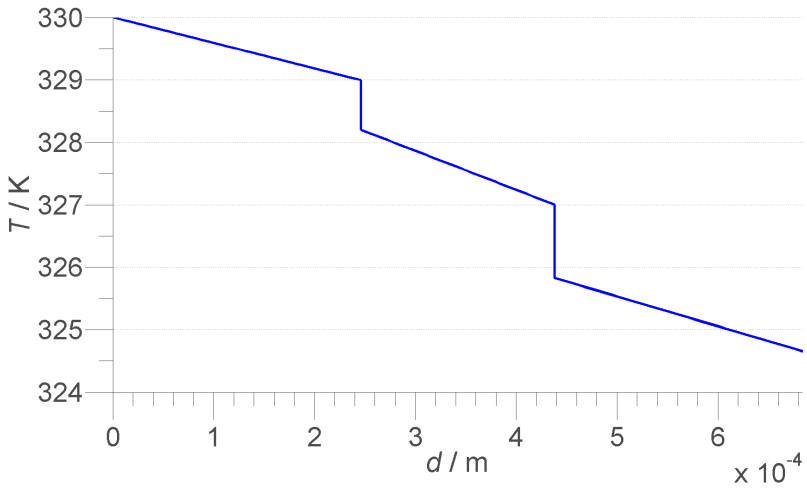

Figure 3.5.15: *Temperature profile in the cell for j = 500* A/m^2*, where T=330 K at the start of the anode backing is set.*

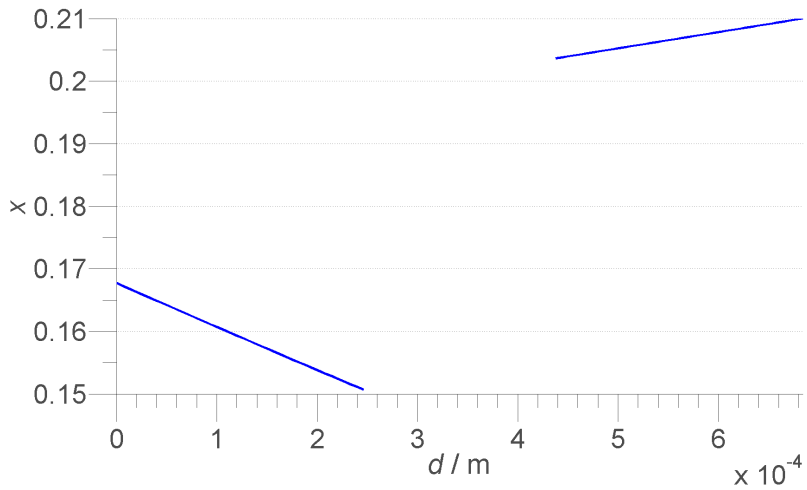

Figure 3.5.16: *Mole fraction profile in the cell for j = 500* A/m^2*, for water at the left side and oxygen at the right side.*

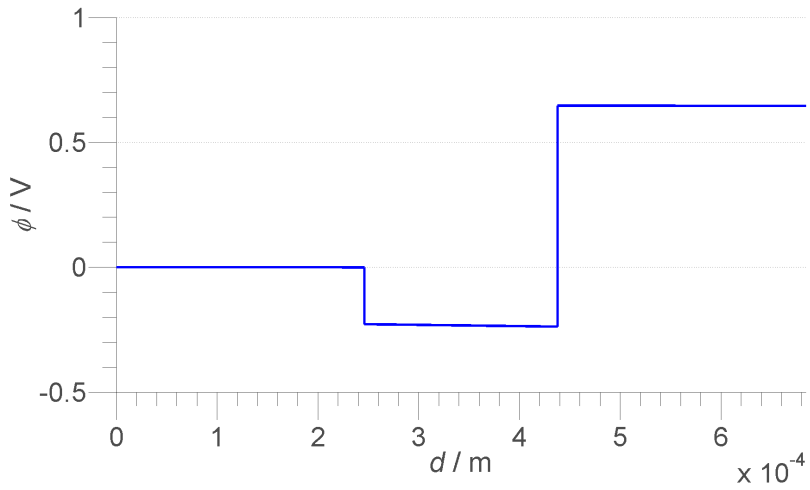

Figure 3.5.17: *Electrical potential profile in the cell for* $j = 500$ $\mathrm{A/m^2}$, $\phi^a = 0$ V *at the start of the anode backing due to definition.*

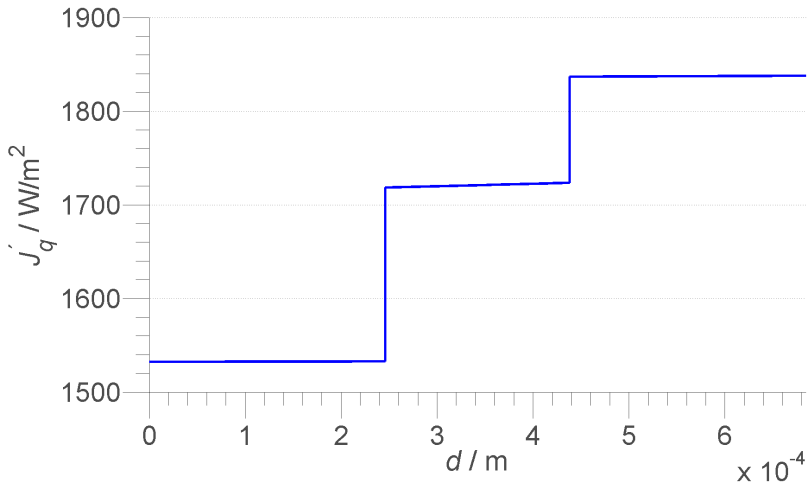

Figure 3.5.18: *Measurable heat flux profile in the cell for* $j = 500$ $\mathrm{A/m^2}$. *Heat is going into the anode backing and out of the cathode backing.*

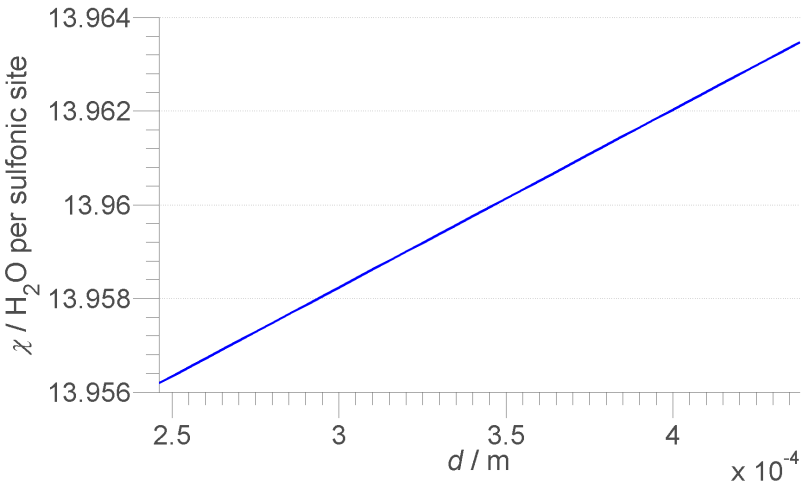

Figure 3.5.19: *Water content profile in the cell for* $j = 500$ $\mathrm{A/m^2}$.

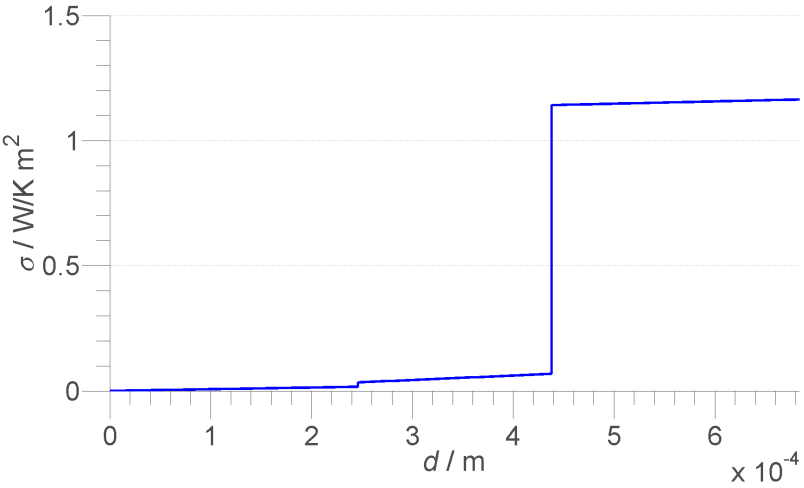

Figure 3.5.20: *Accumulated entropy production profile in the cell for* $j = 500$ $\mathrm{A/m^2}$.

3.5.6 Results for the fuel cell for j = 1000 A/m^2

The profiles in each segment of the fuel cell temperature, composition, electrical potential and heat flux are given in Figures 3.5.21, 3.5.22, 3.5.23 and 3.5.24 respectively for j = 1000 A/m^2. The water content inside the membrane is given in Figure 3.5.25. The accumulated entropy production profile through the cell is given in Figure 3.5.26.

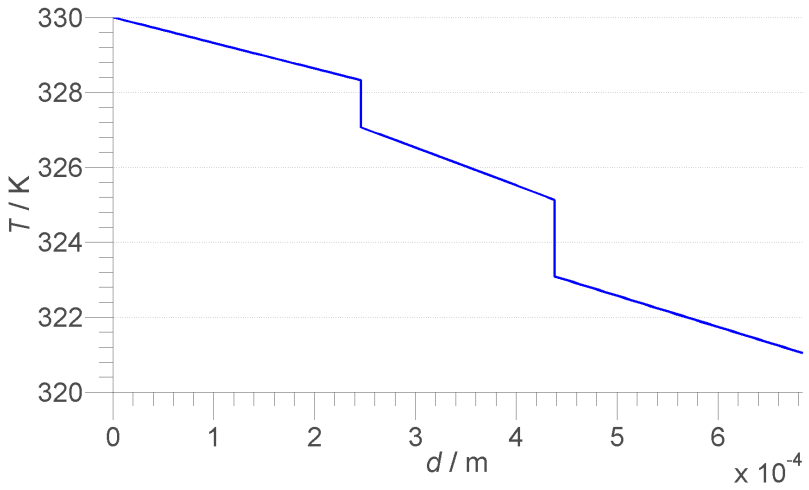

Figure 3.5.21: *Temperature profile in the cell for j = 1000* A/m^2*, where T=330 K at the start of the anode backing is set.*

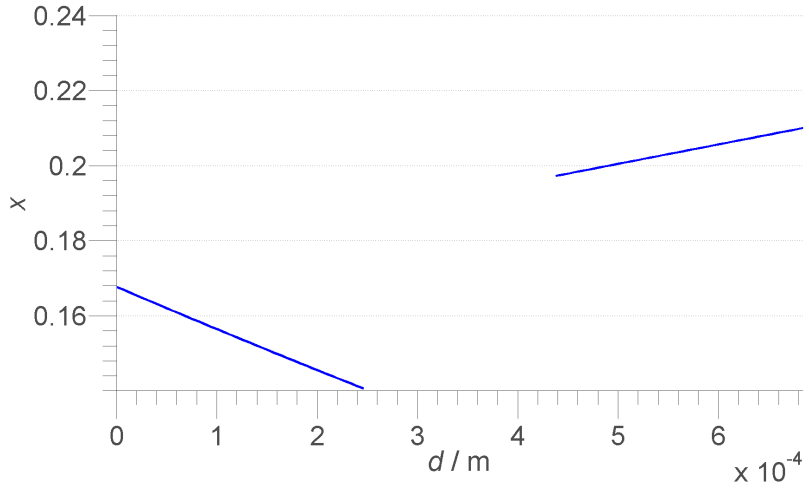

Figure 3.5.22: *Mole fraction profile in the cell for j = 1000* A/m^2*, for water at the left side and oxygen at the right side.*

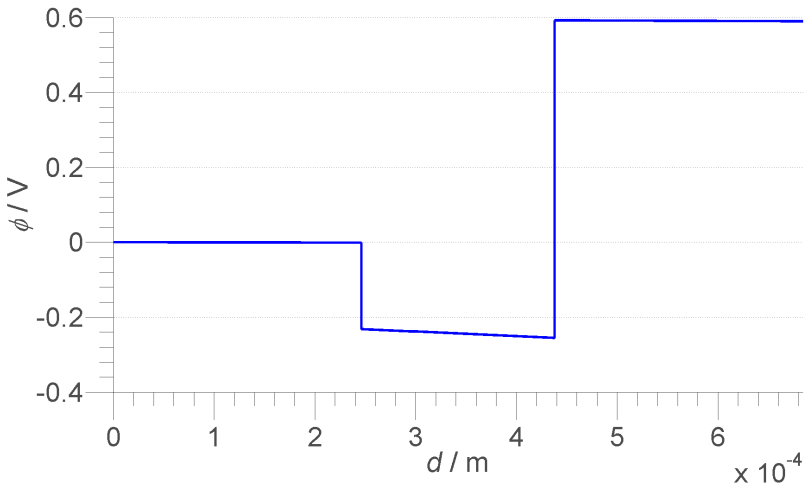

Figure 3.5.23: *Electrical potential profile in the cell for* $j = 1000$ $\mathrm{A/m^2}$, $\phi^a = 0$ V *at the start of the anode backing due to definition.*

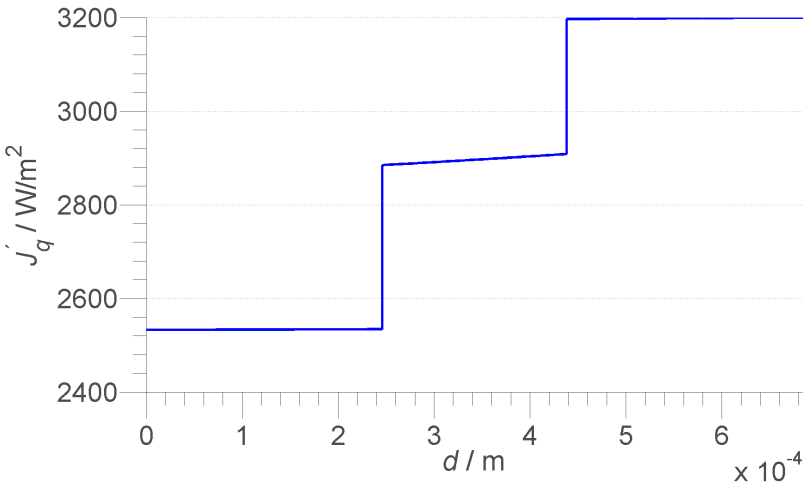

Figure 3.5.24: *Measurable heat flux profile in the cell for* $j = 1000$ $\mathrm{A/m^2}$. *Heat is going into the anode backing and out of the cathode backing.*

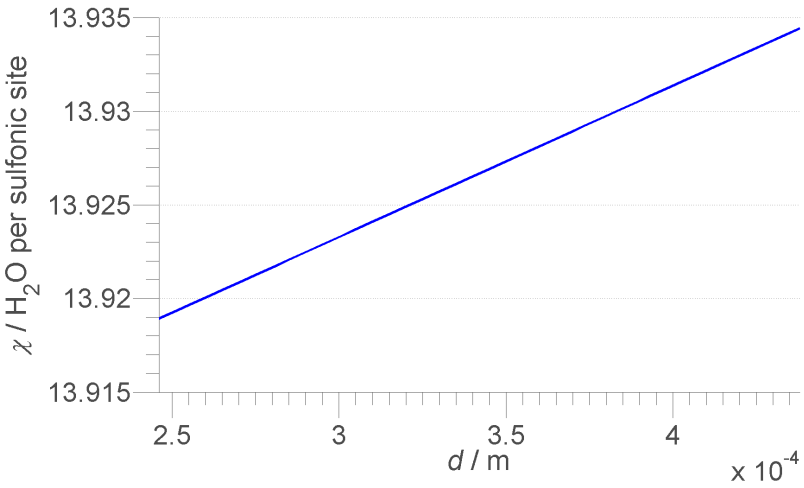

Figure 3.5.25: *Water content profile in the cell for j = 1000* $\mathrm{A/m^2}$.

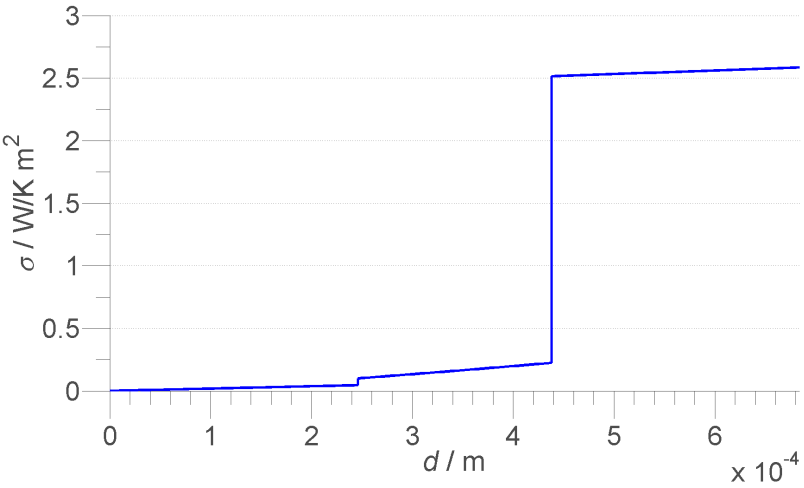

Figure 3.5.26: *Accumulated entropy production profile in the cell for j = 1000* $\mathrm{A/m^2}$.

3.5.7 Results for the fuel cell for j = 5000 $\mathrm{A/m^2}$

The profiles in each segment of the fuel cell for temperature, composition, electrical potential and heat flux are given in Figures 3.5.27, 3.5.28, 3.5.29 and 3.5.30 respectively for j = 5000 $\mathrm{A/m^2}$. The water content inside the membrane is given in Figure 3.5.31.

The accumulated entropy production profile through the cell is given in Figure 3.5.32. The entropy production for all the current densities is given in Table 3.5.3.

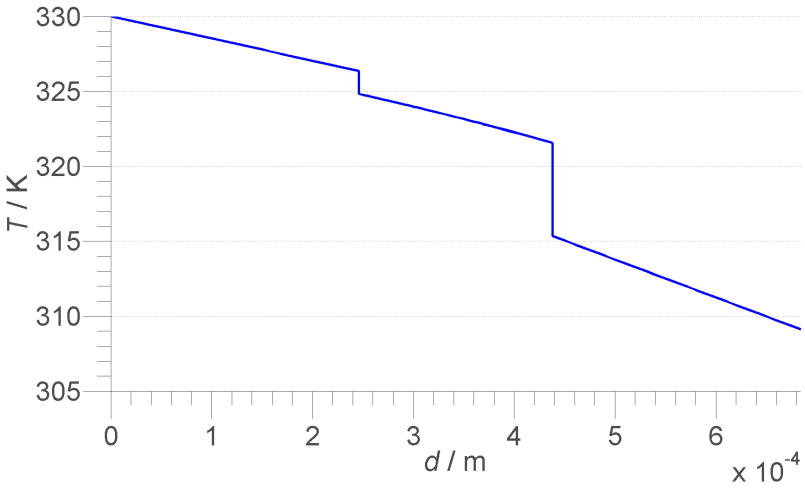

Figure 3.5.27: *Temperature profile in the cell for $j = 5000$ A/m^2, where T=330 K at the start of the anode backing is set.*

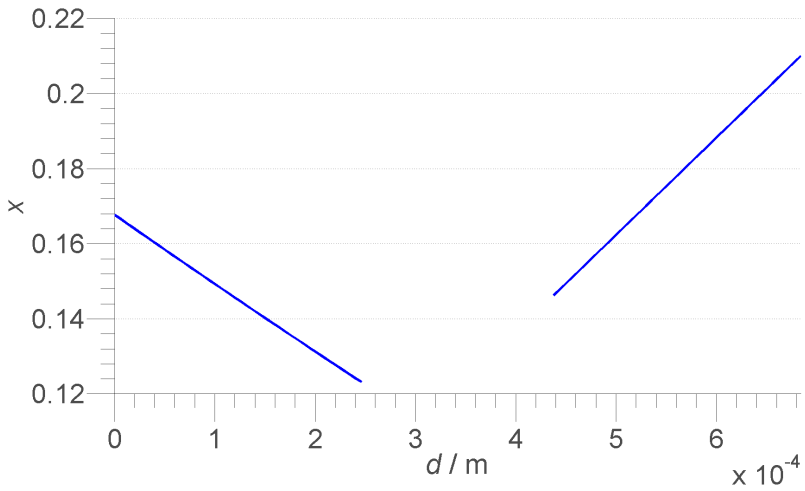

Figure 3.5.28: *Mole fraction profile in the cell for $j = 5000$ A/m^2, for water at the left side and oxygen at the right side.*

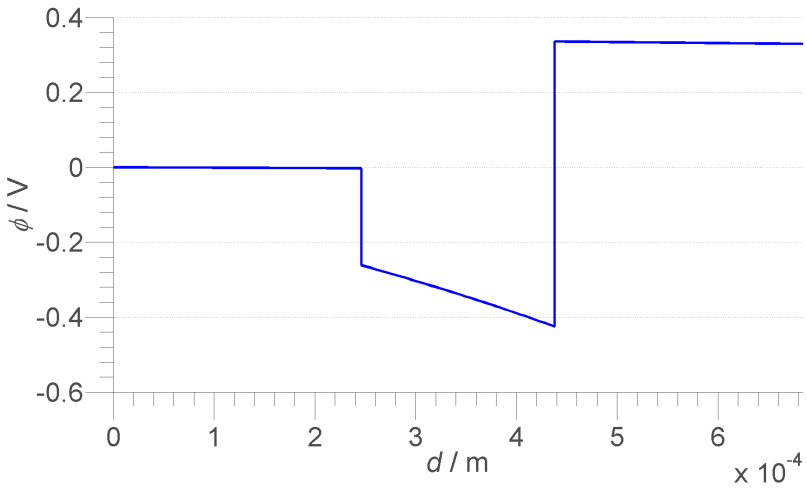

Figure 3.5.29: *Electrical potential profile in the cell for* $j = 5000$ $\mathrm{A/m^2}$, $\phi^a = 0$ V *at the start of the anode backing due to definition.*

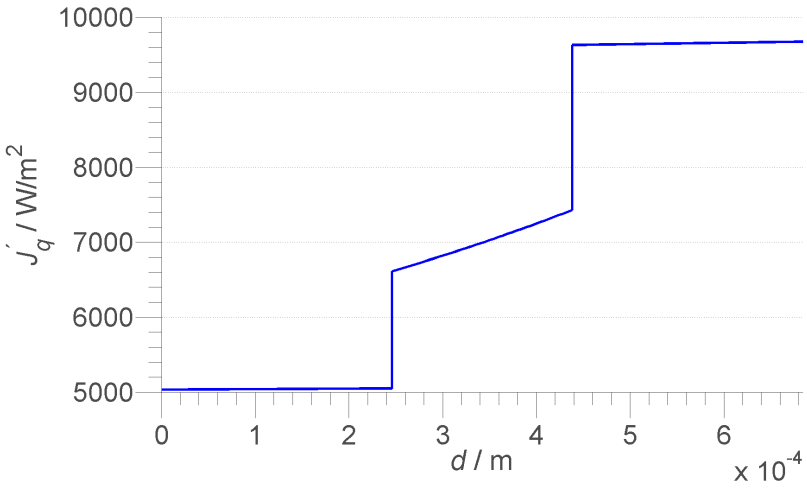

Figure 3.5.30: *Measurable heat flux profile in the cell for* $j = 5000$ $\mathrm{A/m^2}$. *Heat is going into the anode backing and out of the cathode backing.*

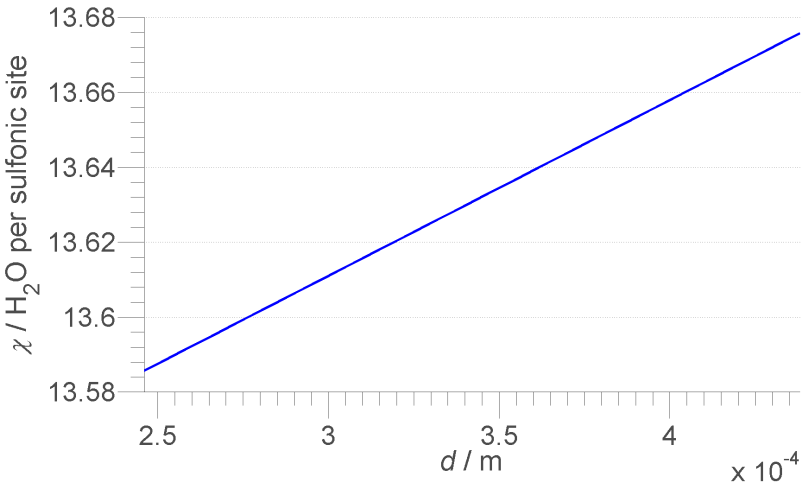

Figure 3.5.31: *Water content profile in the cell for* $j = 5000$ A/m^2.

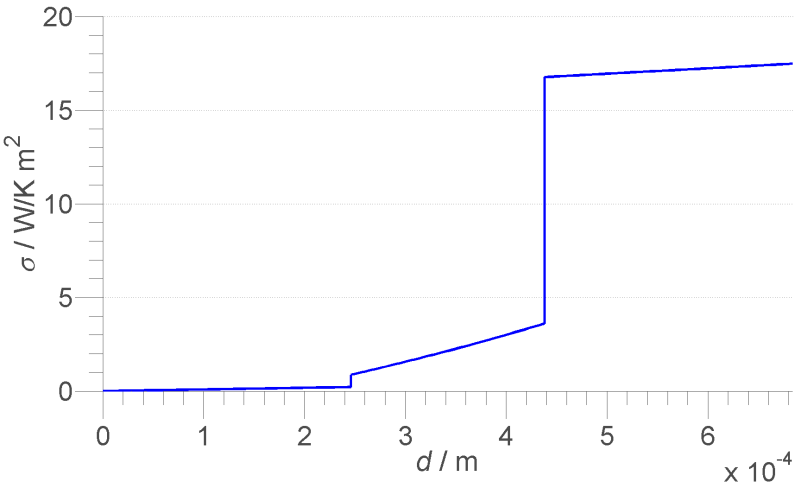

Figure 3.5.32: *Accumulated entropy production profile in the cell for* $j = 5000$ A/m^2.

Table 3.5.3: *Entropy production from total entropy balance and sum of segments*

j / A/m^2	Total entropy production, $\frac{dS_{irr}}{dt}$	Sum local entropy production, $\frac{dS_{irr}}{dt}$
200	0.298 W/K m^2	0.397 W/K m^2
500	0.881 W/K m^2	1.09 W/K m^2
1000	2.02 W/K m^2	2.34 W/K m^2
5000	13.8 W/K m^2	14.7 W/K m^2

3.5.8 Polarization profile

The polarization profile have been found by plotting the electrode potential between the anode and the cathode for varying current densities. This profile is given in Figure 3.5.33.

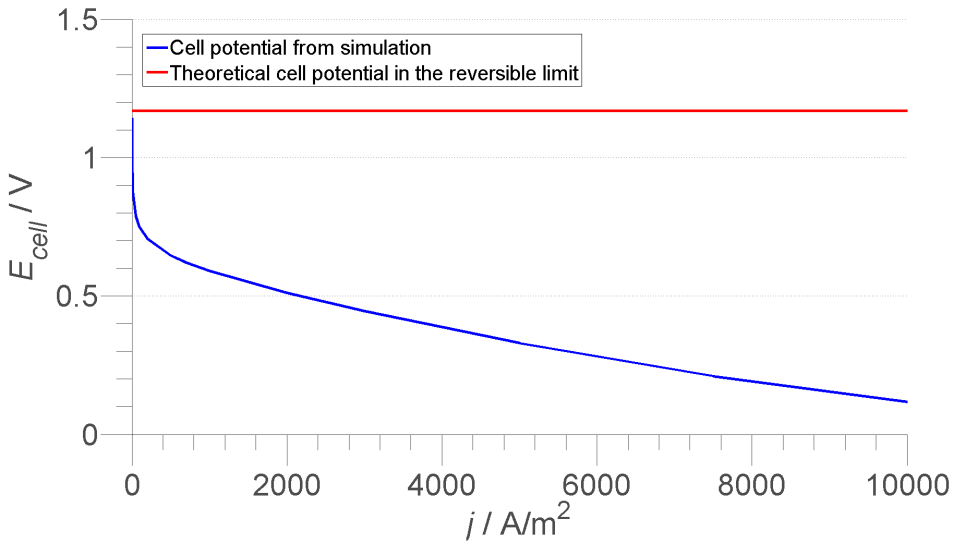

Figure 3.5.33: *Polarization curve for various current densities. The temperature at the anode backing is set as $T = 330$ K.*

3.6 Discussion

3.6.1 Water flux

As can be seen in Figures 3.5.1 and 3.5.2 the effective water transport number reaches a finite value when the current density goes to zero. A value of 0.1 A/m^2 is used because of implications with setting the value equal to zero, which makes the MatLab program to stop. There is a jump in value when the current density reaches zero, but this jump does not increase if the current density is set even lower than 0.1 A/m^2. This value for the anode backing corresponds to the reversible limit and it should be close to the value for the water transport number in the membrane, $t_w^m = 1.2$. The effective water transport number for the anode backing reaches a value very close to 1.2 when the current density reaches zero. The effective water transport number for the cathode backing is 0.5 higher than for the anode backing in the reversible limit, which is as expected from theory. The result for the effective water transport number is a good validity check for the simulation model with respect to accuracy and correctness. It is also an interesting result which shows the effective water transport number to decrease significantly as soon as a current is being drawn from the fuel cell. The drop in value as soon as a current is being drawn in the cell is unexplained. But one possible reason can be the resistances affecting the value for the water flux as soon as a current is drawn.

3.6.2 Measurable heat flux

It is apparent from the curves in Figures 3.5.3 and 3.5.4 that something is happening when the current density goes to zero, as the values are dropping asymptoticly when reaching very low current densities. This may originate from the way MatLab is calculating the heat flux with iterating over the total energy balance. For current densities close to zero it may be that the solution procedure is not optimal when trying to solve the functions, and small variations or assumptions may affect the measurable heat flux significantly. According to theory, the limit of $J_q^{'i}/(j/F)$ when the current density goes to zero is equal to the case of dT = 0 and $d\mu = 0$. Equation (3.21) shows that this limit equals the Peltier coefficient in segment i to the corresponding measurable heat flux in segment i. The limit when the current density goes to zero in Figures 3.5.3 and 3.5.4 should then equal the Peltier coefficient calculated in the simulation for temperature T = 330 K. The Peltier coefficient in both the anode and cathode backing is always nega-

tive regardless of the system. This can be seen from Equations (3.36) and (3.55). The entropy of the hydrogen gas is always positive and larger than the entropy of electrons, which results in a negative Peltier coefficient for the anode backing. It is a bit harder to see what the sign is for the cathode backing. The term $(J_w^a/(j/F) + 0.5)$ is always 0.5 or higher, which then gives a higher negative value from the term with the entropy of water compared to the term with the entropy of oxygen. This results in a negative value for the Peltier coefficient in the cathode backing as well. This implies that the limit value in the graphical solution should also be a negative value for both the anode and cathode backing. The curves are indeed going down to a negative value if one set the current density low enough, but it never reaches a finite value due to the calculation problems MatLab has at this limit. It is therefore impossible to predict what the limit value in these figures would approach, but it is possible to assume that the limit would reach a negative value as the curves are dropping. However, there will not be a finite value due to the problematic solution when the limit of current density reaches zero. The simulation program may need some changes in the future when this limit is reached to avoid these problems and get accurate results for the measurable heat flux close to zero current density.

3.6.3 Special case when current density goes to zero

The electrical potential plot in Figure 3.5.6 in the result section shows a cell potential of roughly 1.15 V for the reversible case when the current density goes to zero. This is very close to the calculated standard potential of the cell at 330 K, which has a value of 1.17 V. The fact that these are so close to each other is a good indication of the model being accurate and the results are in agreement with theory. The system is almost isothermal as can be seen in Figure 3.5.5 and constant temperature, $T = 330$ K, is then safely used for the calculation of Gibbs energy. The small difference between the theoretical calculated cell potential and the simulated cell potential is 0.02 V, a difference that can be caused by numerous reasons. The main reason is most likely due to how the solution procedure works with not being able to set j=0 A/m^2 as this stops the program and cause problems when dividing with zero. When this solution was found, a very small current density of 0.1 A/m^2 was used to ensure that the program did not fall into the region where it would have problems with finishing the calculation. The measurable heat flux profile for the reversible case is given in Figure 3.5.7 and this profile shows a

positive value for all segments, which means that heat is going into the anode backing and out of the cathode backing. The heat that is going out of the cathode backing is higher than the loss in the anode backing, which results in the fuel cell producing heat to the surroundings in overall. This is in agreement with theory. In the reversible limit $\Delta S_0 + \Delta_n S = 0$, which means that $\Delta S_0 > 0$ since $\Delta_n S < 0$. This means that there is a positive heat effect to the surroundings during reversible conditions, which is in agreement with the results. Earlier studies have shown fuel cells to produce heat to the surroundings[3] from both the anode and cathode backing. An explanation as to why this is not happening here, may originate from the previously mentioned problems the program has with finding the correct solution for very low current densities and a small difference in the cell potential. The heat flux could be close to zero in the case of no current density, and small fluctuations around zero is expected when the solution procedure reaches a limit value such as the current density going to zero. The profile of the entropy production in Figure 3.5.8 is very close to what one would expect for the reversible case. The entropy production is zero in all segments and approximately zero in the cathode backing. The reason for the small jump at the cathode backing originates from the fact that 0.1 A/m^2 is used, which then contributes somewhat to the overpotential. This results is in agreement with the reversible theoretical limit where the total entropy production, dS_{irr}/dt, is equal to 0 W/K m^2.

3.6.4 Temperature profiles

Knowledge about the temperature profiles inside the fuel cell is important because this may effect the material properties and overall efficiency of the fuel cell. The model deals with temperature independent properties, but it is possible to extend the simulation model to include temperature effects if the temperature jumps are significant. The temperature is dropping for all four values of current density, with the largest temperature being at the start of the anode backing and the lowest temperature being at the end of the cathode backing. This is in agreement with the work from Vie and Kjelstrup[2], where they observed that the anode was slightly hotter than the cathode at stationary state operation of a polymer electrolyte fuel cell with one Nafion 115 membrane. It is apparent from the profiles that higher current density leads to a higher temperature drop over the fuel cell. This can be seen with the temperature dropping from 330 K to approximately 310 K for j = 5000 A/m^2, which is a temperature difference of 20 K. The

lowest current density, j = 200 A/m^2, gives a temperature drop from 330 K to approximately 327.5 K, which is a temperature difference of 2.5 K. The temperature difference for current density equal to 200 A/m^2 is small enough that temperature effects in material properties can be neglected. For higher current densities, such as j = 5000 A/m^2, it may be important to include these temperature differences in the material properties, to get a more accurate simulation model.

The differential equations (3.31), (3.44) and (3.51) for the temperature in the different segments, and the figures in the result section, indicate that the dominating term is the term with the measurable heat flux when it comes to determination of the temperature gradient at lower current density. This term will contribute negatively to the temperature gradient for positive values of the measurable heat flux, and positively when the measurable heat flux is negative. The Peltier effect will always contribute negatively to the temperature gradient in the anode and cathode backing, while the membrane will have a positive contribution. This is due to the Peltier coefficient always being negative for the anode and cathode backing, and almost always positive for the membrane. This was explained in detail in the previous section for the anode and cathode backing, and similar explanation is valid for the membrane. From Equation (3.49) one can see that the term with the entropy of the protons is always larger than the term with the entropy of water, except for very high temperatures, which then results in a positive Peltier coefficient for all reasonable temperatures. From the differential equations it is clear that the peltier effect will contribute more at higher current densities as the term is linearly dependent of the current density. The coupled term with the mass flux is small compared to the effect form the measurable heat flux and and the Peltier effect and will therefore have close to zero contribution to the temperature gradient. The heat flux results show positive values for all the cases, and it is therefore possible to conclude that the temperature will always drop in the anode and cathode backing of the fuel cell. The higher the current density the larger the drops will be. The membrane will in most cases have a temperature drop as well, according to the results, but the Peltier effect will be the determinative term with higher current densities over 5000 A/m^2. This will result in the temperature increasing or varying non-linearly in the membrane if the current density is high enough. This was tested with the simulation model and current densities higher than 15 000 A/m^2 gave clear contributions of the Peltier effect to the temperature

inside the membrane. This trend was especially visible if the boundary temperature of the anode backing was lowered from T = 330 K to a lower temperature. In this test, the temperature profile became curved and had a peak in the middle of the membrane segment. This test showed how important temperature control is for the PEM fuel cell, and changing the operating temperature clearly affects the system. If the system had two constant temperatures instead of just one, which is the boundary condition at the anode backing, then the overall results would be different, and it would be of interest to study this in future simulations.

3.6.5 Water and oxygen profiles

The transport number for water in the Nafion membrane, t_w^m, is equal to 1.2. 1.2 moles of water is transported from the anode backing and into the membrane per 0.5 moles of hydrogen according to the stoichiometry. This implies that more water is drawn to the end of the anode backing and through the anode surface than hydrogen per second. With a constant pressure, this means that hydrogen accumulates in front of the anode surface. The mole fraction profiles of water in the anode backing support this theoretical approach with the mole fraction dropping down all the way from the start of the anode backing and to the anode surface for all current densities used. Hydrogen is also being drawn to anode surface and reacts according to the electrochemical reaction in Equation (3.1). Oxygen is being transported through the cathode backing and to the cathode surface where the electrochemical reaction in Equation (3.2) is occurring. The oxygen starts at a mole fraction of 0.21 at the right side of the cathode backing, equal to the mole fraction in air, and drops down the closer to the cathode surface one is. This is in agreement with the theoretical approach. Both the mole fraction of oxygen and water has an increasing drop with increased current density.

These results are found by assuming fixed porosity and no clogging of the pores by water. A study has been done on these effects[16] and it has been found that the effects may influence the access of oxygen. This should be included in the future for more accurate results as it may affect the results when trying to model a real fuel cell.

3.6.6 Electrical potential profiles

The anode backing and cathode backing have approximately no variation in electrical potential, as can be seen in the electrical potential profiles in the result section. This is the case for all four current densities used. The differential equations (3.33) and (3.53) for the electrical potential can explain this. The main reason for this originates from the low electrical resistance in the anode and cathode backing, along with the small gradients in these segments. Both the temperature gradient and mole fraction gradient are small in these sections, the highest being the temperature gradient in the anode and cathode backing for j = 5000 A/m^2. This is still not large enough to get a variation in the electrical potential when inserted into the equations due to the anode and cathode electrical resistances being small.

The membrane shows an increasing variation in electrical potential as the current density is increasing. The lowest current density, j = 200 A/m^2, is almost not visible, while the higher current densities shows a clear drop in electrical potential going from the anode side of the membrane to the cathode side. This is due to the ohmic resistance term of Equation (3.46). The electrical resistance coefficient, r^m is varying with respect to the water content, as shown in the theory section, and this parameter is larger than the electrical resistance in the anode and cathode backing. This is the main reason we get an electrical potential drop over the membrane. For j = 5000 A/m^2 the profile shows a non linear drop in the membrane, and this originates from the membrane's electrical resistance not being linear with respect to the water content and the fact that the water content varies more for higher current densities than for lower. The last term of Equation (3.46) has a linear dependency of the current density, and it is thus expected to see an increase in electrical potential difference for increasing current density, in accordance to the obtained results.

The anode surface has an electrical potential drop of close to 0.2 V for all four cases of current densities. The drop is slightly larger for higher current densities, but this difference is small compared to the value of 0.2 V. The potential drop occurring at the anode surface can be as a result of the disappearance of hydrogen due to electrochemical reaction taking place. Similar we have a jump in electrical potential at the cathode surface which is a result of the electrochemical reaction taking place at the surface with

Gibbs energy being converted to electrical energy. The jump is decreasing for increasing current density and this can be explained by the overpotential being larger for increased current density.

The difference in electrical potential at the start of the anode backing and the end of the cathode backing is the cell potential, which can be related to electrical energy. It is evident from looking at the profiles for the varying current densities that increasing the current density also decrease the cell potential and therefore also decrease the potential work that is obtainable from the fuel cell. The main reason for this is the increasing effect of the overpotential at the cathode surface when the current density is increasing, which can be seen in Equations (3.73) and (3.74).

3.6.7 Measurable heat flux profiles

Knowing how the measurable heat flux varies through the fuel cell is important for the design of the heat exchanger in the system. If the measurable heat flux is very large for a given system of the fuel cell then considerations need to be done to properly assure enough transport of heat in or out of the fuel cell. The measurable heat flux profiles are almost identical in behavior, but vary in value for the four different current densities. The measurable heat flux is constant through the anode backing and the cathode backing. The reason for this result originates from the energy balance in these segments; Equations (3.34) and (3.54). There is little enthalpy difference in these segments due to small temperature gradients and there is no variation in electrical potential which results in little change in the heat flux. There is a small difference in the heat flux over the Nafion membrane, which is increasing for increased current density. The single reason for this variation is the variation in electrical potential for higher current densities, which contributes with the term $\mathrm{j}(d\phi/dx)$. Since the electrical potential gradient is negative then it is clear from Equation (3.48) that the measurable heat flux has to increase to maintain the energy balance in the membrane, which we can see in the heat flux profiles. There is a very small change in enthalpies in the membrane due to low temperature gradients, which then has close to zero contribution to the energy balance.

The heat flux is changing in a non linear way in the membrane for higher current densities such as 5000 $\mathrm{A/m^2}$. The reason for this non-linearity is a result from the non linear

increase in electrical resistance in the membrane as the water content is changing. The change in water content is small at lower current densities when compared to higher current densities such as 5000 A/m^2. Which means the non-linear profile will only start to appear for higher current densities, and it will be more apparent as the current density increases. The largest variations in the heat flux for all current densities happens at the electrode surfaces. The main reasons for this is the electrochemical reactions taking place, as well as phase change of water.

Heat is going into the anode backing, and out of the cathode backing in the four cases of current densities.. The heat going out of the cathode backing is higher than what is coming into the anode backing, and the fuel cell is then producing heat, which is as expected. The heat flux is increasing with increased current density, which originates from the energy balances in the system. The temperature gradient is negative in all segments for the four current densities used in this simulation. The electrical potential gradient is also negative in all segments. By looking at the energy balances it is then clear that the measurable heat flux has to increase to compensate for this energy loss. The enthalpy of the start of the membrane is more negative than the enthalpy at the end of the anode backing, due to phase change in water going from gas to liquid. Additionally there is a jump in the electrical potential in the cathode surface. This jump is too small when compared to the change in enthalpy, which means that the cathode surface has an energy loss. This energy loss has to be compensated by an increase in the measurable heat flux, which results in the jump in measurable heat flux in the anode surface. The enthalpy at the start of the cathode backing is less negative than the enthalpy at the end of the membrane, due to phase change in water going from liquid to gas. However, the water flux in the cathode backing is higher than the water flux in the membrane, which results in a more negative term in the energy balance at the start of the cathode backing. Additionally there is a jump in the electrical potential in the cathode surface. The cathode surface then has two energy losses, similar as for the anode surface, which have to be compensated by an increase in the measurable heat flux.

The heat coming into the anode backing is as of now not yet accounted for, and earlier studies have shown the anode backing to produce heat to the surroundings[1]. There is however a difference in the system for this model and previously made models where

only one temperature is kept constant, which is the anode backing temperature of $T = 330$ K. It is therefore not possible to compare this model directly with other simulation results. This difference in system settings is expected to be the main reason for the difference from other simulation results. It would be interesting to see if the results from this model could match the results from earlier experiments if the cathode backing temperature also was controlled and set as constant. This is something that would be important to do in future work as it could be a good test for the validity of the model.

3.6.8 Entropy production

The entropy production increases with increasing electrical current j, as can be seen in the figures in the results section. The profiles are almost identical in behavior with there being very little entropy production in the anode and cathode backing. This can be understood from Equations (3.8) and (3.12). Since the gradients are small in the anode and cathode backing, then there is little contribution to the entropy production as well. The membrane has a bit more noticeable increase in the entropy production, and this is increasing as the current density goes up. The entropy production in Equation (3.10) is dependent on the gradient of the electrical potential, and it is therefore understandable that there is an increasing entropy production in the membrane for increased current density. The gradient of the electrical potential will be more negative the larger the current density is, so the entropy production increase is expected. There is very little entropy production in the anode surface for all four cases, while almost all entropy production in the fuel cell happens in the cathode surface. One of the main reasons for this energy loss in the cathode surface is the overpotential, but the ohmic resistances may also play a role in the entropy production.

The results from the accumulated entropy production calculated from integrating the local entropy production in each segment of the cell, which is shown in the Table 3.5.3 in the results section, is close in value to the entropy production calculated from the total entropy balance of the system in and out of the cell. This is an indication of the system being in agreement with the physical laws and it appears to be an accurate model for this 1-dimensional simplified PEM fuel cell. It appears that higher current densities give a smaller variation in difference of the two ways to calculate the entropy production, but this may be coincidental as there are too few results to back this up further. Further

study on the variations for various current density could provide important information as to which regions of the current density the model is accurate and which regions it may be less accurate. The variation between the two ways of calculating the entropy production may come from the assumptions that is done in the simulation model. Especially is the assumptions done for the surfaces of anode and cathode of interest, and getting better knowledge about would be important for the model. This may also explain the variations in accuracy of the calculated entropy production for various current densities. Higher current densities may in fact make the assumptions more correct, as the ohmic resistance term, which is linear in current density, starts to be more dominant. This would make the neglected temperature differences in the model less important.

3.6.9 Polarization profile

The polarization curves in Figure 3.5.33 shows how the cell potential in the fuel cell is decreasing with increasing current density. The main reason for this decrease is the increase in ohmic loss as current density is increasing. As the current density goes to zero the cell potential goes to the theoretical limit value at the reversible case, which equals 1.17 V for T = 330 K. This a good indication of the model being in agreement with earlier known results and theory. Comparing this result to other polarizations curves show this curve to decrease faster than what is expected. However, study has been done on how the proton exchange membrane fuel cell is affected by the operating temperature, pressure and stoichiometric ratio. This study is done by Bhatt et al.[17]. This study shows how lowering the operating temperature in the fuel cell leads to lower cell potential for all current densities. The polarization curve for 333 K as the operating temperature is shown in their results. To compare this with the results in this simulation model one has to use a mean temperature in the simulated fuel cell. The operating temperature would be close to T = 320 K if one takes an estimate for the current densities from 0 A/m^2 to 5000 A/m^2. The curve for T = 333 K from the work of Bhatt et al. is close to the simulated curve in our results, but it is still higher. If the trend is followed as Bhatt et al. suggest then the simulated results with an operating temperature of 320 K could match experimental results. There may still be variations between this simulated result and experimental results as a consequence of only holding the anode backing temperature constant. It is usually normal to hold the cathode backing temperature constant as well, and this may affect the simulation results so that they differ from

experimental results. A test of increasing the operating temperature was done, which indeed resulted in higher values for the polarization curve, as predicted by the study of Bhatt et al.[17]. The neglect of the anode overpotential in the simulation model may also affect the results somewhat compared to experimental results.

3.6.10 Sensitivity analysis

A sensitivity analysis was performed to get a clear picture of what parameters may influence the results when changed. It became evident that changing material parameters such as the heat conductivity or mass diffusion constant had an impact on the result. For instance, changing the heat conductivity in the anode and the cathode backing to 1 W/m K changed the temperature drop in the membrane significantly and thus made the temperature difference between the anode backing and the cathode backing larger. Increasing the heat conductivity also increased the measurable heat flux in all the segments in the cell, but the difference in value between the anode and cathode backing remained approximately the same.

The diffusion constant of water inside membrane was decreased with a factor of 1000 to see how this affected the results. This changed the results dramatically for some of the profiles. The temperature gradients changed in all segments of the cell. The gradients were now positive instead of negative, which resulted in a temperature increase in all segments of the cell. For instance was there a difference of approximately 10 K down from the anode backing to the cathode backing for j = 1000 A/m^2 when the diffusion constant was equal to $8.05 \cdot 10^{-8}$ m^2/s. With the new diffusion constant the temperature difference was 14 K up from the anode backing to the cathode backing. This is a difference of 24 K. The measurable heat flux also changes sign with the new diffusion constant. Heat was coming out of the anode backing and into the cathode backing, which is the reversed situation to the results obtained for the regular value of diffusion constant. The entropy production has increased with the decrease in the diffusion constant, especially for higher current densities. For the current density of j = 1000 A/m^2 the accumulated entropy production of the end of the cathode backing was almost twice as large as for the smaller diffusion constant. One possible explanation for this can be the increase in resistance for the mass diffusion through the Nafion membrane. This would result in a higher energy loss in the membrane due to mass transport.

It is not only the material properties that are affecting the results, but also input paramets, such as the temperature at the start of the anode backing, which shows how sensitive the simulation of the PEM fuel cell is with respect to system variations. By lowering the start temperature down to 300 K gave completely different results. The temperature in the fuel cell was increasing for almost all values of current density, while the measurable heat flux was negative in all segments. The electrical potential became very low with this temperature, which is in agreement with the experiments done by Bhatt et. al[17], where lowering the temperature would lower the fuel cells potential. The electrical potential became as low as under 0 V for some current densities, which indicates that this temperature would be too low as an operating temperature. The work of Balasubramanian et al.[18] shows that most PEM fuel cell to have an optimal working temperature between 50 ^oC and 90 ^oC (323 K and 363 K). However, they are emphasizing that the working conditions for the fuel cell differ in different surroundings and settings. They further focus on the important aspect of a mathematical simulation model that takes the surroundings of PEM fuel cell into consideration to get an accurate and efficient working temperature.

It is apparent that the change in the material properties are significant for the results. Temperature may change sign and value after what type of heat conductivity and mass diffusion constant that is used. More importantly, the entropy production is also affected by change in the material properties and the change is as much as a factor of 2 times as high when the diffusion constant in the membrane was lowered a factor of 1000. The diffusion constant of water has been studied in other papers[19] where the result is different from the one obtained by Villaluenga et al.[13]. It is therefore not given what the real diffusion constant for water in the Nafion membrane is, if looking at different results. The model predicts that changing the material properties can give different results, and it is then important to have accurate material properties if the results are to reflect a real PEM fuel cell. More study on the material properties, and not just the known quantities, but also the estimated ones, is essential to get a more robust model that can give more accurate results.

3.6.11 Reasonability of the assumptions for the surfaces

As can be seen in Appendix C, the assumption of $\Delta_{m,s}T = 0$ K and $\Delta_{s,m}T = 0$ K may not be correct. By doing some estimations, and using the results for the approximated model, made it possible to get a value for the jump between the surface and the membrane. This was - 1.5 K for current density equal to 500 A/m^2. This is larger than the temperature difference in the anode surface in Figure 3.5.15, which then would indicate that the temperature difference between the surface and the membrane should not be neglected. It is important to take notice that the measurable heat flux in the membrane is taken from the result when the temperature difference was neglected, and it is safe to assume that the heat flux would be different if this assumption was not made. It is still a result worth of mentioning, as it is apparent that the temperature difference between the membrane and surface should not be neglected, and it would enhance the model if further study was done on the surface and its behavior. A test was performed with inserting the calculated properties into the equations related to the anode and cathode surface. The results shows the effect of including the temperature drop between the surface and the membrane to have the largest effect for smaller current densities. The current density of 1000 A/m^2 was almost not effected at all, apart from temperature changes. The validity check with the difference between the total entropy production and the sum of the integrated local entropy production was not better when this was taken into consideration, so it is clear that this assumption may not be the main reason for the difference. The assumption with no variations in the chemical potential in the surfaces may be of more importance to the variations in the entropy production. To estimate this value further knowledge is needed for the surfaces to get an accurate value.

3.7 Conclusion

Theory of irreversible thermodynamics is used to describe the fluxes and forces in a Proton Exchange Membrane fuel cell. The system is reduced to only have variations in one dimension for simplicity and keeping the equations and simulation model on a basic level. The system consists of heat and transport of mass and charge. A simulation model has been established which gives profiles in temperature, mole fraction of components, electrical potential, measurable heat flux and entropy production in the anode and cathode backing, membrane and in the surfaces between each segment. Results from plotting the effective transport number of water in the anode and cathode backing provided information that further backs up the robustness and accuracy of the model. The plotting of the effective water transport number did also provide information about the validity of the model, as the effective water transport reached the theoretical limit 1.2 in the anode backing when the current density went to zero. This is in agreement with theory.

Four cases of different current densities have been studied in detail, j = 200 A/m^2, j = 500 A/m^2, j = 1000 A/m^2 and j = 5000 A/m^2 along with a theoretical approach when the current density goes to zero and we get a reversible fuel cell. The entropy production is calculated with summation of the integrated local entropy production and from the total entropy balance in the cell. The values are in agreement with each other, but there is a little difference which may originate from assumptions, especially from the calculations in the surfaces. Higher current densities seem to have more accurate results when it comes to entropy production, and this may originate from the assumptions being less important as the terms where the current density is included start to be more dominant.

The sensitivity analysis shows that having accurate knowledge about material properties is important to get a robust model that can predict better and more accurate results. Changing the mass diffusion constant inside the Nafion membrane or the heat conductivity in the anode and cathode backing gave different results, and some of the gradients changed sign and value. Similar sensitivity result for the assumptions in the surfaces gave an indication that the temperature difference between the surface and the membrane may not be close to zero as earlier expected. A calculation done with esti-

mated parameters from the results made this temperature difference larger than what was originally found for the surfaces. Further study on the material properties and especially on the behavior in the surface would be crucial to get a more robust model and more accurate results.

3.8 Nomenclature

Symbol	Units	Description
a_w	$-$	Water activity
χ	molecules H_2O per sulfonic site	Water content in the membrane
c_w	mol/m^3	Water concentration
$\Delta_n G$	kJ/mol	Gibbs energy change at temperature T and pressure p for reaction n
$\Delta_n H$	kJ/mol	Enthalpy change at temperature T pressure p for reaction n
$\Delta_n S$	J/K mol	Entropy change at temperature T pressure p for reaction n
D_j	m^2/s	Diffusion constant of component j
d_m	m	Membrane thickness
d_s	m	Anode and cathode backing thickness
dS_{irr}/dt	W/m^2 K	Total entropy production
E^0_{cell}	V	Maximum cell potential at temperature T in the reversible limit
E_{cell}	V	Cell potential at temperature T
F	C/mol	Faradays constant
H_i	kJ/mol	Entropy of component i
J^i_j	mol/m^{2}s	Mass flux of component j in segment i
$J^{'0}_{q'}$	W/m2	Start measurable heat flux in the anode backing
J'_q	W/m^2	Measurable heat flux
J_u	W/m^2	Energy flux
j	A/m^2	Current density
j^0	A/m^2	Exchange current density of oxygen in air
λ^i	W/m K	Thermal conductivity of segment i
λ^s_i	W/m^2 K	Thermal conductivity of surface segment i
L_{ij}	varies	Transport transference coefficient
l_{qq}	W/m	Phenomenological coefficient
$l_{q\mu}$	mol K/m s	Phenomenological coefficient
$l_{q\phi}$	C K/m s	Phenomenological coefficient
$l_{\mu q}$	mol K/m s	Phenomenological coefficient
$l_{\phi q}$	C K/m s	Phenomenological coefficient
$l_{\mu\mu}$	mol^2 K/J m s	Phenomenological coefficient

$l_{\mu\phi}$	mol K/V m s	Phenomenological coefficient
$l_{\phi\mu}$	mol K/V m s	Phenomenological coefficient
$l_{\phi\phi}$	C K/V m s	Phenomenological coefficient
M_m	kg/mol	Molar mass of the membrane
$\mu_{i,T}$	kJ/mol	Chemical potential of component i at temperature T
n	mol	Moles of electrons transported per reaction n
Ω	m^2	Cross-sectional area
π^i	J/mol	Peltier heat of segment i
ϕ^0	V	Start electrical potential in the anode backing
ϕ_{eff}	V	Effective electrical potential
ϕ	V	Electrical potential
p	Pa	Pressure
p_w^*	Pa	Saturated pressure of water
$q^{i,*}$	J/mol	Measurable heat of transfer of segment i
ρ_m	kg/m^3	Membrane density
R	J/K mol	Universal gas constant
r^i	Ω m	Electrical resistance of segment i
r^s	Ω m^2	Electrical resistance of the surface
S_i	J/K mol	Entropy of component i
σ^i	W/K m^2	Local entropy production in segment i
T	K	Temperature
T^0	K	Start temperature in the anode backing
t_w^m	–	Transport number of water in the membrane
W_{ideal}	kJ/mol	Maximum work
dW_{lost}/dt	W/m^2	Lost work
x_i	–	Mole fraction of component i
x_w^0	–	Start mole fraction of water in the anode backing
x_w^*	–	Mole fraction of saturated water

3.9 References

References

1. S. Kjelstrup, A. Røsjorde, Local and Total Entropy Production and Heat and Water Fluxes in a One-Dimensional Polymer Electrolyte Fuel Cell, J. Phys. Chem. B 2005, 109, 9020 - 9033.

2. P. J.S. Vie, S. Kjelstrup, Thermal conductivities from temperature profiles in the polymer electrolyte fuel cell, Electrochimica Acta 49 (2004) 1069 - 1077.

3. A.Z. Weber, J. Newman, Modeling transport in polymer-electrolyte fuel cells, Chem Rev. 2004 Oct;104(10):4679-726.

4. Mench, M. M.; Burford, D.; Davies, T. W. In Situ Temperature Distribution Measurement in an Operating Polymer Electrolyte Fuel Cell. In Proceedings of IMECE 03, 2003 ASME International Mechanical Engineering Congress and Exposition, Nov 16-21, 2003.

5. Kjelstrup, S., Bedeaux, D. (2008) Non-equilibrium thermodynamics for heterogeneous systems.

6. S. Kjelstrup, D. Bedaux, E. Johannessen, J. Gross, Non-Equilibrium Thermodynamics for Engineers, 2010.

7. M. M. Mench, Fuel Cell Engines, John Wiley & Sons, New Jersey, 2008.

8. S. Kjelstrup, P.J.S. Vie, D. Bedaux, Surface Chemistry and Electrochemistry of Membranes, chapter Irreersible thermodynamics of membrane surface transport with application to polymer fuel cells, 1999, 483 - 510.

9. T. E. Springer, T. A. Zawodzinski, S. Gottesfeld, Polymer Electrolyte Fuel Cell Model, J. Electrochem. Soc., Vol. 138, No. 8, August 1991.

10. O. S. Burheim, J.G. Pharoah, H. Lampert, P.J.S. Vie, S. Kjelstrup, Through-Plane Thermal Conductivity of PEMFC Porous Transport Layers.

11. O. S. Burheim, P.J.S. Vie, J.G. Pharoaha, S. Kjelstrup, Ex situ measurements of through-plane thermal conductivities in a polymer electrolyte fuel cell, Journal of Power Sources 195 (2010) 249 - 256.

12. P.J.S. Vie, Characterization and Optimazation of the Polymer Electrolyte Fuel cell, PhD thesis, NTNU, Norway 2002.

13. J.P.G Villaluenga, B. Seoana, V.M. Barragán, C. Ruiz-Bauzá, Thermo-osmosis of mixtures of water and methanol through a Nafion membrane.

14. A. Røsjorde, personal communication with regards his MatLab code on the PEM fuel cell.

15. E. M. Hansen, E. Egner, S. Kjelstrup, Peltier Effects in Electrode Carbon.

16. P. T. Nguyen, T. Berning, N. Djilali, Computational model of a PEM fuel cell with serpentine gas flow channels, Journal of Power Sources 130 (2004) 149 - 157.

17. S. Bhatt, B. Gupta, V. K. Sethi, M. Pandey, Polymer Exchange Membrane (PEM) Fuel Cell: A Review, International Journal of Current E ngineering and Technology, Vol.2, No.1 (March 2012).

18. B. Balasubramanian, F. Barbir, J. Neutzler , Operating temperature and pressure of PEM fuel cell systems in automative applications.

19. T.E. Springer, T.A. Zawodinski and S. Gottesfeld. Polymer electrolyte fuel cell model. J. Electrochem. Soc., 138:2334-2342, 1991.

A Experimental data

Experiments were done with GDL 10 AA

	ΔT, °C	Unit Rising	Time, min.	M, kg	Flux, kg/m2.s
30	0	0	0	0	0
	3	32	1575	0,00126	-5,61314E-06
	5	75	1140	0,002976	-1,83164E-05
	10	248	1225	0,009878	-5,65883E-05
45	0	0	0	0	0
	3	117	1390	0,004651	-2,34825E-05
	5	147	850	0,005848	-4,82832E-05
	10	245	600	0,009759	-0,000114135
60	0	0	0	0	0
	3	217	1010	0,008641	-6,00403E-05
	5	216	770	0,008601	-7,83905E-05
	10	424	585	0,016901	-0,000202736
75	0	0	0	0	0
	3	210	720	0,008362	-8,1501E-05
	5	316	720	0,012591	-0,000122723
	10	295	300	0,011754	-0,000274936

Experiments were done with GDL 10 BA

	ΔT, °C	Unit Rising	Time, min.	M, kg	Flux, kg/m2.s
30	0	0	0	0	0
	3	161	983	0,0064069	4,57383E-05
	5	128	611	0,0050902	5,84627E-05
	10	282	825	0,0112348	9,55645E-05
45	0	0	0	0	0
	3	127	740	0,0050503	4,78928E-05
	5	163	600	0,0064867	7,58678E-05
	10	260	480	0,010357	0,000151418
60	0	0	0	0	0
	3	165	445	0,0065665	0,000103552
	5	315	515	0,0125515	0,00017103
	10	434	360	0,0172996	0,000337224
75	0	0	0	0	0
	3	325	420	0,0129505	0,000216383
	5	387	366	0,0154243	0,00029574
	10	450	240	0,017938	0,000524503

Figure A.0.1: *Experimental data from the work of Akyalcin on water transport in a Nafion membrane system with heat and transport of water.*

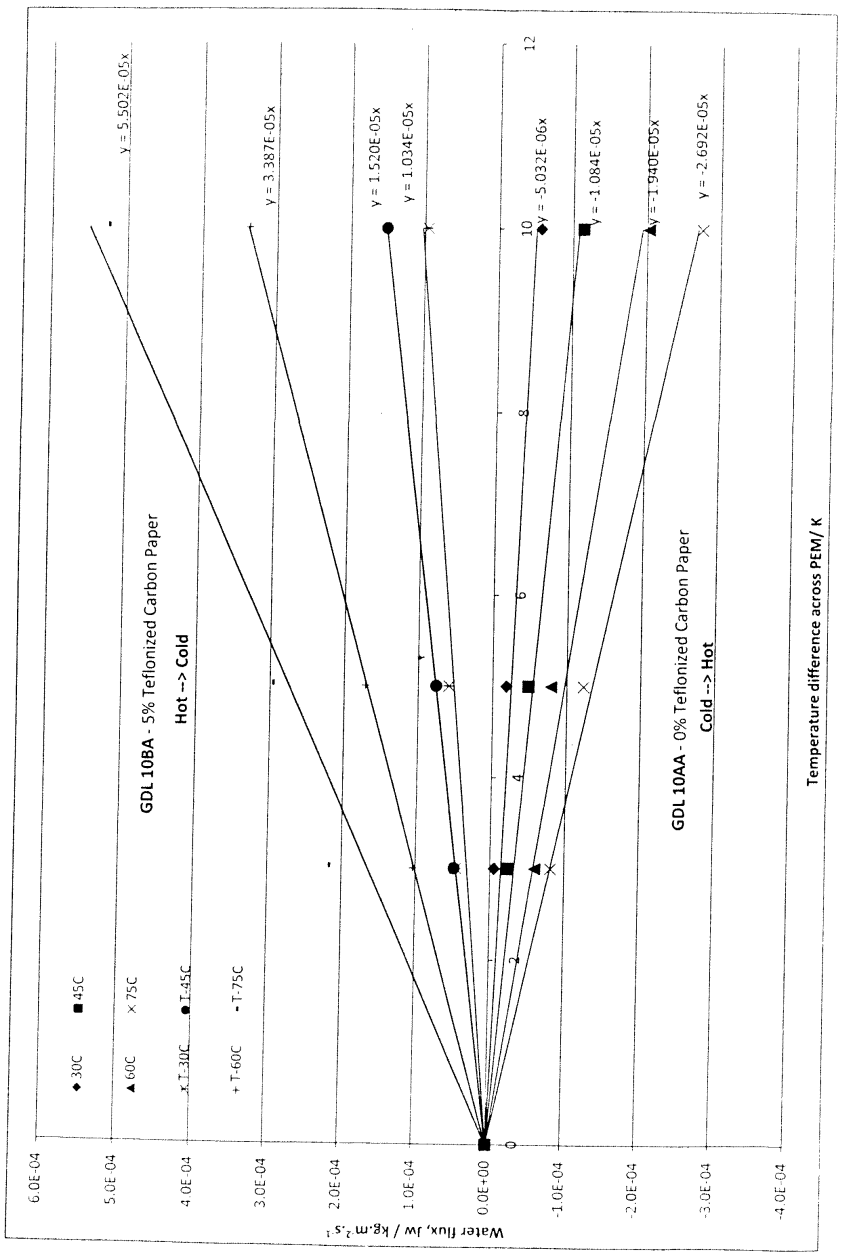

Figure A.0.2: *Experimental data from the work of Akyalcin on water transport plotted against the temperature difference.*

B Calculations of k-coefficient for heat of transfer

The k-coefficient used in the simulation to find the heat of transfer is calculated for each layer in the membrane cell system by using an approximation based on Equation 8.27 in [28]. This equation is calculating the heat of transfer through a surface s, which can be shown in Figure 16 where i is into the surface and o is out of the surface. For instance would the heat of transfer going through the surface between the bulk phase at the left and the Sigracet layer at the left be given by the heat resistivity at the i index side (bulk phase) and the o index side (Sigracet layer).

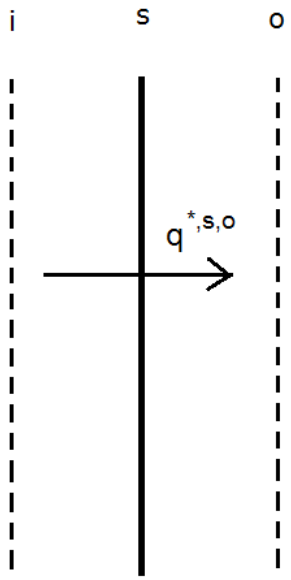

Figure B.0.3: *The surface s between two layers, where i is into the surface and o is out of the surface. There are three surfaces used in this system, one between the bulk phase and the Sigracet layer, one between the Sigracet layer and the Nafion membrane and one between the Nafion membrane and the other Sigracet layer.*

This approximation for the k-coefficient is given in Equation (B.1). Equation (B.3) is the same approximation, but with inserted thermal conductivity.

$$k = \frac{r_{qq}^{i}}{r_{qq}^{i} + r_{qq}^{o}} \tag{B.1}$$

$$k = \frac{\frac{1}{\lambda^i}}{\frac{1}{\lambda^i} + \frac{1}{\lambda^o}} \tag{B.2}$$

In these equations i refers to into the surface, while o refers to out of the surface, as shown in Figure 16. Below is calculations of the k-coefficients for GDL10AA and GDL10BA. The first three equations shows the calculations for GDL10AA for the surface between liquid water and Sigracet layer s_a, the surface between Sigracet layer s_a and membrane and the surface between the membrane and Sigracet layer s_c respectively. The last three equations shows the calculations for GDL10BA for the surface between liquid water and Sigracet layer s_a, the surface between Sigracet layer s_a and membrane and the surface between the membrane and Sigracet layer s_c respectively. In these equations, l refers to the bulk phase, s to a Sigracet phase and m refers to the Nafion membrane. The heat conductivity of water, λ_l is equal to 0.58 W/m K[32].

$$k_{l,s}(GDL10AA) = \frac{\frac{1}{0.42}}{\frac{1}{0.42} + \frac{1}{0.58}} = 0.58 \tag{B.3}$$

$$k_{s,m}(GDL10AA) = \frac{\frac{1}{0.229}}{\frac{1}{0.229} + \frac{1}{0.42}} = 0.65 \tag{B.4}$$

$$k_{m,s}(GDL10AA) = \frac{\frac{1}{0.42}}{\frac{1}{0.42} + \frac{1}{0.229}} = 0.35 \tag{B.5}$$

$$k_{l,s}(GDL10BA) = \frac{\frac{1}{0.33}}{\frac{1}{0.33} + \frac{1}{0.58}} = 0.64 \tag{B.6}$$

$$k_{s,m}(GDL10BA) = \frac{\frac{1}{0.229}}{\frac{1}{0.229} + \frac{1}{0.33}} = 0.59 \tag{B.7}$$

$$k_{m,s}(GDL10BA) = \frac{\frac{1}{0.33}}{\frac{1}{0.33} + \frac{1}{0.229}} = 0.41 \tag{B.8}$$

C Estimation of difference in chemical potential and temperature at the surface

In the model temperature differences is neglected when going from the anode surface to the membrane, and from the membrane to the cathode surface. In this appendix an estimation of the difference will be shown, where the surface parameters are estimated. This calculation is done on the surface between the anode backing and the membrane for j = 500 A/m^2 The equations for the difference in temperature taken from the theory section and is given in Equations (C.1) and (C.2).

$$\Delta_{a,s}T = -\frac{J_q^{'a}}{\lambda_a^s} + \frac{q^{*,a}}{\lambda_a^s}\left(J_w - t_w\frac{j}{F}\right) + \pi^a\frac{j}{\lambda_a^s F} \tag{C.1}$$

$$\Delta_{m,s}T = -\frac{J_q^{'m}}{\lambda_m^s} + \frac{q^{*,m}}{\lambda_m^s}\left(J_w^m - t_w^m\frac{j}{F}\right) + \pi^m\frac{j}{\lambda_m^s F} \tag{C.2}$$

From the graphical representation of the results one can withdraw information of the measurable heat fluxes. These quantities are roughly: $J_q^{'m} = 1955\ W/m^2$ at the anode surface and $J_q^{'m} = 1960\ W/m^2$ at the cathode surface. It is assumed that the temperature difference from the surface to the membrane is zero, but it is possible to calculate this difference if one estimate the surface parameter λ_m^s. One can assume that this parameter can be estimated in the same way as for the anode surface heat conductivity, dividing the membrane heat conductivity with the membrane thickness. Doing this results in an estimated value of $\lambda_m^s = 1190$ W/m K. The value of λ_m is taken from the program at the anode surface. The peltier effect is given in the theory section, and calculation of this at the given temperature at the anode surface, which is estimated to be 329 K from Figure 2, gives $\pi^m = 2041$ J/mol. The temperature at the cathode side of the surface is estimated to be 325 K, which gives gives $\pi^m = 2185$ J/mol. The heat of transfer, $q^{*,m}$ is found in the theory section, and calculation of this at these given temperatures gives $q^{*,m}$ = -25449 J/mol and $q^{*,m}$ = -25072 J/mol for the anode and cathode surface respectively. The calculated water flux in this simulation is equal to J_w^m=J_w^a = 0.0019 mol/m^2s, extracted from Table X. Inserted in Equations (C.1) and (C.2) gives the following calculation:

$$\Delta_{s,m}T = -\frac{1955}{1190} + \frac{-25449}{1190}\left(0.0019 - 1.2\frac{500}{96485}\right) + 2041\frac{500}{1190 \cdot 96485} = -1.5K \quad \text{(C.3)}$$

$$\Delta_{s,m}T = -\frac{1960}{1190} + \frac{-25072}{1190}\left(0.0019 - 1.2\frac{500}{96485}\right) + 2185\frac{500}{1190 \cdot 96485} = -1.5K \quad \text{(C.4)}$$

From these calculations the estimated temperature drop between the surface and the membrane at the anode side is -1.5 K and the estimated temperature drop between the surface and the membrane at the cathode side is -1.5 K.

D Collected results for all current densities

This Appendix has figures with collected results for all current densities plotted in the same figures.

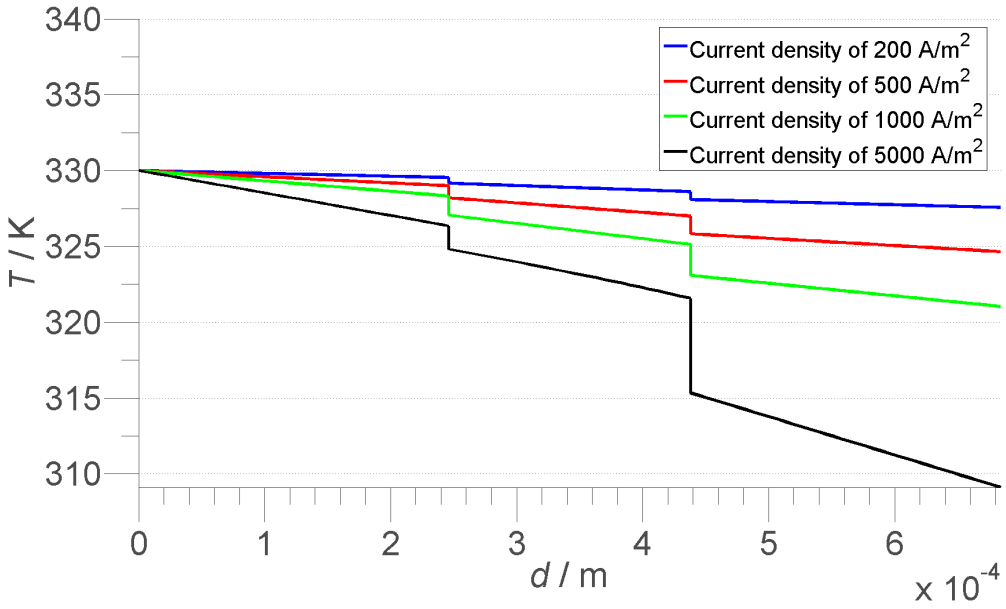

Figure D.0.4: *The temperature profiles in the system for j = 200 A/m^2, j = 500 A/m^2, j = 1000 A/m^2 and j = 5000 A/m^2.*

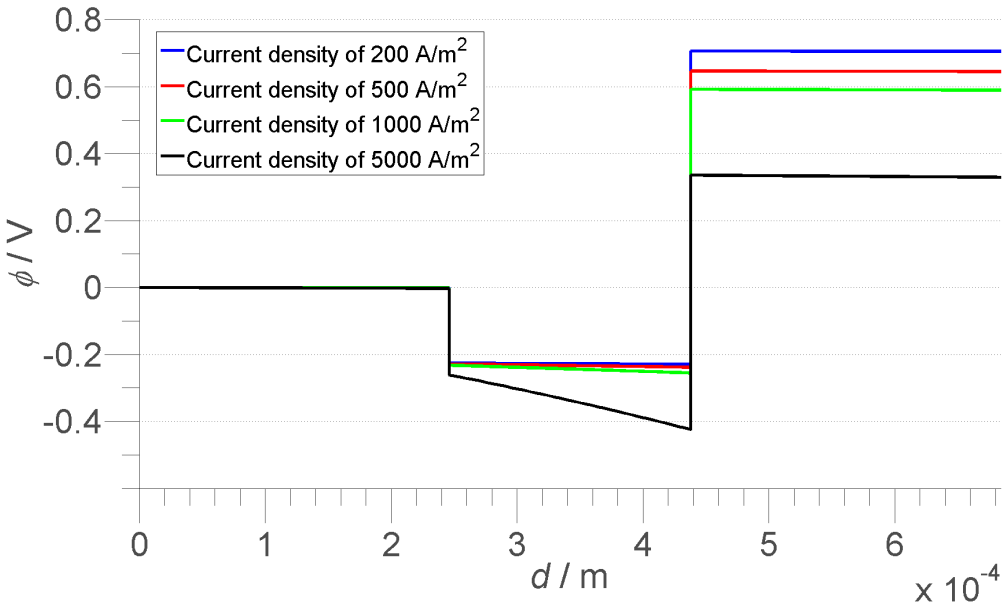

Figure D.0.5: *The electrical potential profiles in the system for $j = 200$ A/m^2, $j = 500$ A/m^2, $j = 1000$ A/m^2 and $j = 5000$ A/m^2.*

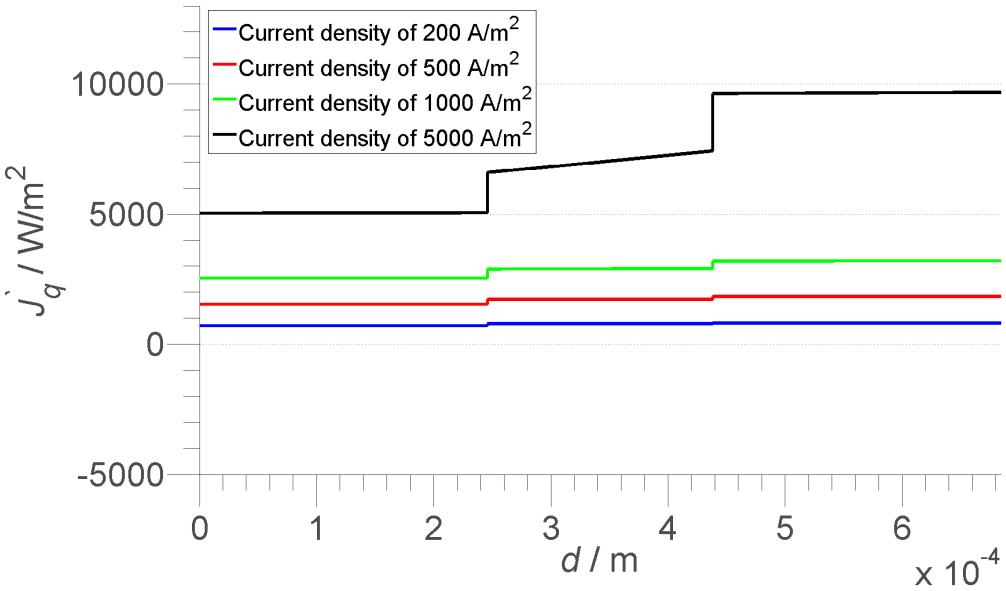

Figure D.0.6: *The measurable heat flux profiles in the system for $j = 200$ A/m^2, $j = 500$ A/m^2, $j = 1000$ A/m^2 and $j = 5000$ A/m^2.*

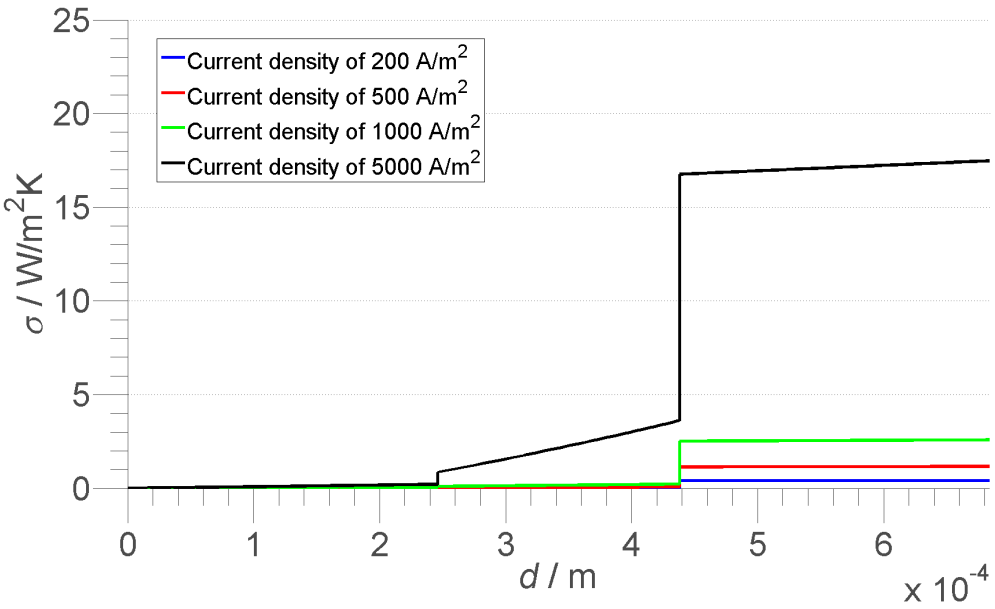

Figure D.0.7: *The accumulated entropy production in the system for* $j = 200$ *A/*m^2, $j = 500$ *A/*m^2, $j = 1000$ *A/*m^2 *and* $j = 5000$ *A/*m^2.

E MatLab scripts for Part 2

```
%-------------------------------------------------------------------------
% Caclulate transport trough the membrane
%-------------------------------------------------------------------------
function Master()
%

%%Main program with constants and for loop for iterating on the adsorption
%%enthalpy.


    clc;
    clear all;
    close all;
    global Const;

    avector = [];
    hvector = [];

    for i = 1:500

        % membrane/liquid constants

        %Const.absH1 = -50+i*50;
        %[J/mol       ] % water absorption into sigracet enthalpy
        Const.absH1 = -460;
        %[J/mol       ] % water absorption into sigracet enthalpy
        Const.absH2 = 0;
        %[J/mol       ] % enthalpy difference from membrane and liquid water

        Const.R = 8.314;
        %[J/(K mol) ] % universal gas constant
        Const.dm = 192E-6;
        %[m           ] % membrane thickness
        Const.ds = 246E-6;
        %[m           ] % Sigracet thickness
        Const.Lm = 0.229;
        %[ W/(K m)    ] % membrane conductivity
        Const.LsAA = 0.42;
        %[ W/(K m)    ] % Sigracet 10AA conductivity
```

```
40          %Const.LsAA = 0.33;
41          %[ W/(K m)    ] % Sigracet 10BA conductivity
42          Const.Lw = 0.58;
43          %[ W/(K m)    ] % Water conductivity
44          Const.Dm = 8.05E-9;
45          %[ m^2/s      ] % membrane diffusion coefficient
46          Const.Ds = 2.3E-8;
47          %[ m^2/s      ] % Sigracet diffusion coefficient
48          %ro = 1970;
49          %[kg/m^3      ] % membrane density, wet
50          ro = 1640;
51          %[kg/m^3      ] % membrane density, dry
52          Mm = 1.1;
53          %[kg/mol      ] % membrane molar mass
54          Const.mm = Mm/ro;
55          %[m^3/mol     ] % membrane concentration
56          Const.Mw = 0.018;
57          %[kg/mol      ] % water molar mass
58          Const.dfH = -285000;
59          %[J/mol       ] % water liquid enthalpy of formation
60          Const.cm = 75;
61          %[J/(K mol)   ] % water liquid heat capacity
62          Const.Tq = 298;
63          %[K           ] % reference temperature
64
65          Ta = 304.5;                    %[K] Temperature in bath a 30 oC
66          Tc = 301.5;                    %[K] Temperature in bath c 30 oC
67
68          %Ta = 319.5;                    %[K] Temperature in bath a 45 oC
69          %Tc = 316.5;                    %[K] Temperature in bath c 45 oC
70
71          %Ta = 334.5;                    %[K] Temperature in bath a 60 oC
72          %Tc = 331.5;                    %[K] Temperature in bath c 60 oC
73
74          %Ta = 349.5;                    %[K] Temperature in bath a 75 oC
75          %Tc = 346.5;                    %[K] Temperature in bath c 75 oC
76
77          P = 1e5;                       %[Pa] Pressure
78
79          [T, muT, Je, Jm] = TransportP(Ta, Tc, P);
80
```

```
81          x = [0, Const.ds, Const.ds+Const.dm, Const.ds+Const.dm+Const.ds];
82
83          dT = Ta-Tc;
84          a = Jm/dT;
85
86          avector = [avector a];
87          hvector = [hvector Const.absH1];
88
89      %figure('Units', 'pixels', ...
90      %'Position', [100 100 500 375]);
91      %hold on
92      %TFit = line(x,T);
93      %xfit1 = line([x(2) x(2)], [T(4) T(1)]);
94      %xfit2 = line([x(3) x(3)], [T(4) T(1)]);
95      %hXLabel = xlabel('d / m'                    );
96      %hYLabel = ylabel('T / K'                 );
97      %set(TFit                          , ...
98      %'LineWidth'        , 4          );
99      %set(xfit1                           , ...
100     %'Color'            , 'k'          , ...
101     %'LineStyle'        , '--'         , ...
102     %'LineWidth'       , 4            );
103     %set(xfit2                           , ...
104     %'Color'            , 'k'          , ...
105     %'LineStyle'        , '--'         , ...
106     %'LineWidth'       , 4            );
107
108
109     %xlim([0 x(4)])
110     %ylim([T(4) T(1)])
111
112     %hLegend = legend( ...
113     %[TFit, xfit1], ...
114     %'Simulated flux coefficient values from MatLab' , ...
115     %'Experimental flux coefficient value'                , ...
116     %'location', 'NorthWest' );
117
118
119     %set( gca                          , ...
120     %'FontName'    , 'Helvetica', ...
121     %'FontSize',45)
```

```
122     % set([hXLabel, hYLabel], ...
123     %'FontName'    , 'AvantGarde');
124     %set(hLegend              , ...
125     %'FontSize'    , 30          );
126     %set([hXLabel, hYLabel]   , ...
127     %'FontSize'    , 55          );
128
129     %set(gca, ...
130     %'Box'         , 'off'       , ...
131     %'TickDir'     , 'out'       , ...
132     %'TickLength'  , [.04 .04]   , ...
133     %'XMinorTick'  , 'on'        , ...
134     %'YMinorTick'  , 'on'        , ...
135     %'YGrid'       , 'on'        , ...
136     %'XColor'      , [.3 .3 .3], ...
137     %'YColor'      , [.3 .3 .3], ...
138     %'LineWidth'   , 1           );
139
140     %Experimental water flux, divided with the molar mass of water to
141     %get the unit mol/m^2 s
142
143     a = -5.032e-6/Const.Mw;
144     b = -1.084e-5/Const.Mw;
145     c = -1.94e-5/Const.Mw;
146     d = -2.692e-5/Const.Mw;
147     e = 1.034e-5/Const.Mw;
148     f = 1.52e-5/Const.Mw;
149     g = 3.387e-5/Const.Mw;
150     h = 5.502e-5/Const.Mw;
151
152     x = [];
153
154     for j = 1:1:10000
155         ny = -1+j*1;
156         x = [x ny];
157         y(j) = h;
158     end
159
160     end
161
162     %%Plotting of the simulated mass flux from different adsorption
```

```
163     %%enthalpy values with experimental mass flux.
164
165     figure('Units', 'pixels', ...
166     'Position', [100 100 500 375]);
167     hold on;
168     hFit    = line(hvector,avector);
169     expFit = line(x,y);
170     hXLabel = xlabel('\it{\DeltaH_{abs}} \rm{/ J/mol}'                    );
171     hYLabel = ylabel('\it{J_{w}/\DeltaT} \rm{/ mol/m^{2}s K}'
    );
172     set(hFit                              , ...
173     'LineWidth'         , 4             );
174     set(expFit                            , ...
175     'Color'             , 'r'             , ...
176     'LineStyle'         , '--'          , ...
177     'LineWidth'        , 4              );
178
179    hLegend = legend( ...
180     [hFit, expFit], ...
181     'Simulated values from MatLab' , ...
182     'Experimental value'                  , ...
183     'location', 'NorthWest' );
184
185     xlim([6800 8200])
186     ylim([0.0024 0.0036])
187
188      set( gca                            , ...
189     'FontName'    , 'Helvetica', ...
190     'FontSize',45)
191      set([hXLabel, hYLabel], ...
192     'FontName'    , 'AvantGarde');
193     set(hLegend                 , ...
194     'FontSize'    , 30            );
195     set([hXLabel, hYLabel]   , ...
196     'FontSize'    , 45            );
197
198     set(gca, ...
199     'Box'           , 'off'      , ...
200     'TickDir'       , 'out'      , ...
201     'TickLength'    , [.04 .04] , ...
202     'XMinorTick'    , 'on'       , ...
```

```
203       'YMinorTick'   , 'on'        , ...
204       'YGrid'        , 'on'        , ...
205       'XColor'       , [.3 .3 .3], ...
206       'YColor'       , [.3 .3 .3], ...
207       'LineWidth'    , 1           );
208
209
210   end
211
212
213
214
215   % a_{a}=a_{c}
216   %This is the main program for calculation the mass and heat flux through
217   %the membrane cell system.
218   function [T, muT, Je, Jm] = TransportP(Ta, Tc, Pa)
219   %
220
221       global Const;
222
223       Tm = (Ta + Tc) / 2;
224       %The pressure is really not needed as this is a system with liquid
225       %water and the resistivities are dependable on the pressure.
226       Pc = Pa;
227       am = 1;
228
229       Rqq = rqq_s(Ta) + rqq_m(Tm) + rqq_s(Tc);
230       Rmq = rmq_s1(Ta) + rmq_m(Tm) + rmq_s2(Tc);
231       Rmm = rmm_s1(Ta) + rmm_m(Tm,am) + rmm_s2(Tc);
232       R = Rmm*Rqq - Rmq*Rmq;
233
234
235       dT = 1/Tc-1/Ta;
236       Jm = -dT *(Hg(Tm)*Rqq + Rmq)/R;
237       Je = +dT *(Hg(Tm)*Rmq + Rmm)/R;
238       dT_sa = Je*rqq_s(Ta) + Jm*rmq_s1(Ta);
239       dT_mm = Je*rqq_m(Tm) + Jm*rmq_m(Tm);
240       dT_sc = Je*rqq_s(Tc) + Jm*rmq_s2(Tc);
241       dmuT_sa = - Je*rmq_s1(Ta) - Jm*rmm_s1(Ta);
242       dmuT_mm = - Je*rmq_m(Tm) - Jm*rmm_m(Tm,am);
243       dmuT_sc = - Je*rmq_s2(Tc) - Jm*rmm_s2(Tc);
```

```
244
245
246         T = [Ta, 1/(1/Ta + dT_sa), 1/(1/Ta + dT_sa + dT_mm),...
247             1/(1/Ta + dT_sa + dT_mm + dT_sc)];
248         muT = [0, dmuT_sa, dmuT_sa + dmuT_mm, dmuT_sa + dmuT_mm + dmuT_sc];
249
250
251 %
252 end
253
254 %-------------------------------------------------------------------------
255 % Resistivities, calculation for r_{qq}, r_{mq} and r_{mm} for Sigracet
256 % layers and membrane.
257 %-------------------------------------------------------------------------
258
259
260 function r = rqq_m(T)
261     global Const;
262     r = Const.dm/Const.Lm./T./T;
263 end
264
265 function r = rqq_s(T)
266     global Const;
267     r = Const.ds/Const.LsAA./T./T;
268 end
269
270
271 function r = rmq_m(T)
272     global Const;
273     k = (1/Const.Lm)/((1/Const.Lm)+(1/Const.LsAA));
274     h = k*(Const.absH1-Const.absH2) + Hg(T)+Const.absH2;
275     r = - h*Const.dm/Const.Lm./T./T;
276 end
277
278 function r = rmq_s1(T)
279     global Const;
280     k = (1/Const.LsAA)/((1/Const.Lw)+(1/Const.LsAA));
281     h = -k*Const.absH1 + Hg(T)+Const.absH1;
282     r = - h*Const.ds/Const.LsAA./T./T;
283 end
284
```

```
285 function r = rmq_s2(T)
286     global Const;
287     k = (1/Const.LsAA)/((1/Const.Lm)+(1/Const.LsAA));
288     h = -k*(Const.absH1-Const.absH2) + Hg(T)+Const.absH1;
289     r = - h*Const.ds/Const.LsAA./T./T;
290 end
291
292 function r = rmm_m(T,a)
293     global Const;
294     k = (1/Const.Lm)/((1/Const.Lm)+(1/Const.LsAA));
295     h = k*(Const.absH1-Const.absH2) + Hg(T)+Const.absH2;
296     mu = Const.R*T*Const.mm./a./Xia(a);
297     r = h.^2*Const.dm/Const.Lm./T./T + mu*Const.dm/Const.Dm./T;
298 end
299
300 function r = rmm_s1(T)
301     global Const;
302     k = (1/Const.LsAA)/((1/Const.Lw)+(1/Const.LsAA));
303     h = -k*Const.absH1 + Hg(T)+Const.absH1;
304     mu = Const.R*T/22240;
305     r = h.^2*Const.ds/Const.LsAA./T./T + mu*Const.ds/Const.Ds./T;
306 end
307
308 function r = rmm_s2(T)
309     global Const;
310     k = (1/Const.LsAA)/((1/Const.Lm)+(1/Const.LsAA));
311     h = -k*(Const.absH1-Const.absH2) + Hg(T)+Const.absH1;
312     mu = Const.R*T/22240;
313     r = h.^2*Const.ds/Const.LsAA./T./T + mu*Const.ds/Const.Ds./T;
314 end
315
316
317
318 %-------------------------------------------------------------------------
319 % Table functions
320 %-------------------------------------------------------------------------
321
322 % enthalpy of gas
323 function r = Hg(T)
324 %
325     global Const;
```

```
326         r = Const.dfH + Const.cm*(T-Const.Tq);
327         %r = Const.dfH;
328     %
329     end
330
331     % membrane water content
332     function xi = Xi(a)
333     %
334         if (length(a) == 1)
335         %
336             if      (a<0)
337                 xi = 0.0043;
338             elseif  (a<1)
339                 xi = 0.043 + 17.81*a - 39.85*a.^2 + 36.0*a.^3;
340             elseif  (a<3)
341                 %xi = 14 + 1.4*(a-1);
342                 xi = 16.8 - 2.8*exp(-(a-1)/0.0607);
343             else
344                 xi = 16.8;
345             end
346             %xi = 14 + 1.4*(a-1);
347             %xi = 0.0043 + 17.81*a - 39.85*a.^2 + 36.0*a.^3;
348         %
349         else
350         %
351             %cannot use xi = Xi(a) since there is checking of the
352             %value of each a inside xi
353             xi = ones(size(a));
354             for i=1:length(a)
355                 xi(i) = Xi(a(i));
356             end
357         %
358         end
359     %
360     end
361
362     % membrane water content derivative
363     function xi = Xia(a)
364     %
365         if (length(a) == 1)
366         %
```

```
367             if       (a<0)
368                 xi = 0;
369             elseif   (a<1)
370                 xi = 17.81 - 2*39.85*a + 3*36.0*a.^2;
371             elseif   (a<3)
372                 %xi = 1.4;
373                 xi = 2.8/0.0607*exp(-(a-1)/0.0607);
374             else
375                 xi = 0;
376             end
377             %xi = 1.4;
378             %xi = 17.81 - 2*39.85*a + 3*36.0*a.^2;
379         %
380         else
381         %
382             %cannot use xi = Xi(a) since there is checking of the
383             %value of each a inside xi
384             xi = ones(size(a));
385             for i=1:length(a)
386                 xi(i) = Xia(a(i));
387             end
388         %
389         end
390 %
391 end
```

F MatLab scripts for Part 3

```
%-------------------------------------------------------------------
% MAIN PROGRAM! All materiel properties and constants used in
%the simulation is listed in the string Const. Three current
%densities are used, 200, 500, 1000 and 5000 A/m^2 which gives 4
%sets of results.
%-------------------------------------------------------------------
function Master3()

    clc;
    clear all;
    close all;
    global Const;

    phi = [];
    Jqja = [];
    Jqjc = [];
    Jwaj = [];
    Jwcj = [];

    j = [200 500 1000 5000];
    %[A/m^2]        % Current density

    for i = 1:length(j)
        Const.j = j(i);

        %General constants used in the simulation

        Const.R = 8.314;
        %[J/(K mol) ] % universal gas constant
        Const.F = 96500;
        %[C/mol      ] % Faradays constant
        Const.Tq = 298;
        %[K           ] % reference temperature
        Const.p = 1.013e5;
        %[Pa          ] % Standard pressure


        %Materiel properties of the Nafion membrane

```

```
40          Const.dm = 192E-6;
41          %[m             ] % membrane thickness
42          Const.Dm = 8.05E-8;
43          %[ m^2/s        ] % membrane diffusion coefficient
44          Const.rodry = 1640;
45          %[kg/m^3        ] % membrane density dry
46          Const.Mm = 1.1;
47          %[kg/mol        ] % membrane molar mass
48          Const.Mw = 0.018;
49          %[kg/mol        ] % water molar mass
50          Const.twm = 1.2;
51          %[Unitless      ] % Transport number of water
52
53
54          %Materiel properties of the anode and cathode backing
55
56          Const.ds = 246E-6;
57          %[m             ] % Sigracet thickness
58          Const.LsAA = 0.42;
59          %[ W/(K m)      ] % Sigracet 10AA conductivity
60          %Const.LsAA = 0.33;
61          %[ W/(K m)      ] % Sigracet 10BA conductivity
62          Const.rac = 1e-4;
63          %[ ohm m) ] %Electrical resistance in anode/cathode
64          Const.Dwh = 5e-5;
65          %[ m^2/s) ] %Binar diffusion of hydrogen and water
66          Const.DON = 5e-5;
67          %[ m^2/s) ] %Binar diffusion of oxygen and nitrogen
68          Const.xO20 = 0.21;
69          %[ unitless ] %Mole fraction of oxygen at the end of
70          % the cathode backing
71
72
73          %Materiel properties of the Anode and Cathode surface
74
75          Const.Ls = 1710;
76          %[ W/(K m^2) ] % Surface 10AA conductivity
77          %Const.Ls = 1300;
78          %[ W/(K m^2) ] % Surface 10BA conductivity
79          Const.rs = 7.2e-6;
80          %[ohm m^2]  %Electrical resistance
```

```
81          Const.tad = 0;
82          %[Unitless    ] % Combined transport number of water and
83          % hydrogen in the anode backing
84
85
86
87          %%Fluxes%%
88
89          Const.j0 = 2.5e-3;
90          %[A/m^2]        % Exchange current density
91          Const.JH2 = Const.j/(2*Const.F);
92          %[kg/m^s s]     % Hydrogen gas flux
93          Const.JO2 = -Const.j/(4*Const.F);
94          %[kg/m^s s]     % Oxygen gas flux
95
96
97          %%Entalphies and entropies%%
98
99          Const.Hwm = -285e3;
100         %[J/mol]        %Entalphy for liquid water
101         Const.Swm = 70;
102         %[J/K mol]      %Entropy for liquid water
103         Const.cpw = 75;
104         %[J/K mol]      %Cp for liquid water
105         Const.Hwgas = -242e3;
106         %[J/mol]        %Entalphy for gas water
107         Const.Swgas = 189;
108         %[J/K mol]      %Entropy for gas water
109         Const.cpwgas = 34;
110         %[J/K mol]      %Cp for gas water
111         Const.Sh = 192;
112         %[J/K mol]      %Entropy for protons (H+)
113         Const.cpH = 21;
114         %[J/K mol]      %Cp for protons (H+)
115         Const.Se = -2;
116         %[J/K mol]      %Entropy for electrons
117         Const.Sh2 = 131;
118         %[J/K mol]      %Entropy for H2 (g)
119         Const.cph2 = 29;
120         %[J/K mol]      %Cp for H2 (g)
121         Const.So2 = 205;
```

```
122         %[J/K mol]    %Entropy for O2 (g)
123         Const.cpo2 = 29;
124         %[J/K mol]    %Cp for O2 (g)
125         Const.Hwvap = Const.Hwgas-Const.Hwm;
126         %[J/K mol]    %Vaporization enthalpy of water
127
128
129         %Starting parameters
130
131         Const.T0 = 330;
132         %[K]    %Temperature of the start at the anode backing
133         Jq0 = 1000;
134         %[W/m^2]    %Initial heat flux guess
135
136         options = optimset('Display','iter');
137
138
139         %Starting the simulation
140
141         [X, FVAL, EXITFLAG] = fsolve('Transport', Jq0, options, 0);
142
143         [res,phic, Jqa, Jqc, Jwa, Jwc, Xvector, Yvector] = Transport(X,1);
144
145
146          %Making the vectors for plotting the profiles in the same plot
147
148          phi = [phi, phic];
149          Jqa = Jqa/(Const.j/Const.F);
150          Jqc = Jqc/(Const.j/Const.F);
151          Jwaj1 = Jwa/(Const.j/Const.F);
152          Jwcj1 = Jwc/(Const.j/Const.F);
153          Jqja = [Jqja, Jqa];
154          Jqjc = [Jqjc, Jqc];
155          Jwaj = [Jwaj, Jwaj1];
156          Jwcj = [Jwcj, Jwcj1];
157
158          Temp(:,i) = Yvector(:,1);
159          phi2(:,i) = Yvector(:,2);
160          x(:,i) = Yvector(:,3);
161          Jq(:,i) = Yvector(:,4);
162          S(:,i) = Yvector(:,5);
```

```
163
164         Thickness(:,i) = Xvector(:,1);
165
166
167     end
168
169
170     %Plotting the Temperatures
171
172                 figure('Units', 'pixels', ...
173         'Position', [100 100 500 375]);
174         hold on
175         myfit1 = line(Thickness(:,1), Temp(:,1));
176         myfit2 = line(Thickness(:,2), Temp(:,2));
177         myfit3 = line(Thickness(:,3), Temp(:,3));
178         myfit4 = line(Thickness(:,4), Temp(:,4));
179         hXLabel = xlabel('\it{d} \rm{/ m}'                    );
180         hYLabel = ylabel('\it{T} \rm{/ K}'                 );
181         xlim([0 Thickness(end,end)])
182         ylim([Temp(end,4) 340])
183         set(myfit1                          , ...
184         'LineWidth'         , 4           );
185         set(myfit2                            , ...
186         'LineWidth'         , 4             , ...
187         'Color'             , 'r'           );
188         set(myfit3                            , ...
189         'LineWidth'         , 4           , ...
190         'Color'             , 'g'           );
191         set(myfit4                            , ...
192         'LineWidth'         , 4           , ...
193         'Color'             , 'k'           );
194
195
196
197         hLegend = legend( ...
198        [myfit1, myfit2, myfit3, myfit4], ...
199         'Current density of 200 \rm{A/m^{2}}' , ...
200         'Current density of 500 \rm{A/m^{2}}' , ...
201         'Current density of 1000 \rm{A/m^{2}}' , ...
202         'Current density of 5000 \rm{A/m^{2}}' , ...
203       'location', 'NorthWest' );
```

```
204
205          set( gca                          , ...
206          'FontName'    , 'Helvetica', ...
207          'FontSize',45);
208          set([hXLabel, hYLabel], ...
209          'FontName'    , 'AvantGarde');
210          set(hLegend                , ...
211          'FontSize'    , 30            );
212          set([hXLabel, hYLabel]   , ...
213          'FontSize'    , 45            );
214
215         set(gca, ...
216         'Box'          , 'off'      , ...
217         'TickDir'      , 'out'      , ...
218         'TickLength'   , [.04 .04] , ...
219         'XMinorTick'   , 'on'       , ...
220         'YMinorTick'   , 'on'       , ...
221         'YGrid'        , 'on'       , ...
222         'XColor'       , [.3 .3 .3], ...
223         'YColor'       , [.3 .3 .3], ...
224         'LineWidth'    , 1          );
225
226
227      %Plotting the phi
228
229                  figure('Units', 'pixels', ...
230         'Position', [100 100 500 375]);
231         hold on
232         myfit1 = line(Thickness(:,1), phi2(:,1));
233         myfit2 = line(Thickness(:,2), phi2(:,2));
234         myfit3 = line(Thickness(:,3), phi2(:,3));
235         myfit4 = line(Thickness(:,4), phi2(:,4));
236         hXLabel = xlabel('\it{d} \rm{/ m}'                      );
237         hYLabel = ylabel('\it{\phi} \rm{/ V}'                  );
238         xlim([0 Thickness(end,end)])
239         ylim([-0.6 0.8])
240         set(myfit1                              , ...
241         'LineWidth'        , 4            );
242         set(myfit2                              , ...
243         'LineWidth'        , 4              , ...
244         'Color'            , 'r'             );
```

```
245         set(myfit3                              , ...
246         'LineWidth'         , 4                 , ...
247         'Color'             , 'g'                 );
248         set(myfit4                              , ...
249         'LineWidth'         , 4                 , ...
250         'Color'             , 'k'                 );
251
252
253
254         hLegend = legend( ...
255        [myfit1, myfit2, myfit3, myfit4], ...
256        'Current density of 200 \rm{A/m^{2}}' , ...
257        'Current density of 500 \rm{A/m^{2}}' , ...
258        'Current density of 1000 \rm{A/m^{2}}' , ...
259        'Current density of 5000 \rm{A/m^{2}}' , ...
260       'location', 'NorthWest' );
261
262          set( gca                              , ...
263          'FontName'    , 'Helvetica', ...
264          'FontSize',45);
265          set([hXLabel, hYLabel], ...
266          'FontName'    , 'AvantGarde');
267          set(hLegend                 , ...
268          'FontSize'    , 30            );
269          set([hXLabel, hYLabel]   , ...
270          'FontSize'    , 45            );
271
272         set(gca, ...
273         'Box'           , 'off'     , ...
274         'TickDir'       , 'out'     , ...
275         'TickLength'    , [.04 .04] , ...
276         'XMinorTick'    , 'on'      , ...
277         'YMinorTick'    , 'on'      , ...
278         'YGrid'         , 'on'      , ...
279         'XColor'        , [.3 .3 .3], ...
280         'YColor'        , [.3 .3 .3], ...
281         'LineWidth'     , 1           );
282
283
284          %Plotting the mole fraction
285
```

```
286                 figure('Units', 'pixels', ...
287         'Position', [100 100 500 375]);
288         hold on
289         myfit1 = line([Thickness(1,1) Thickness(40,1)], [x(1,1) x(40,1)]);
290         myfit2 = line([Thickness(1,2) Thickness(40,2)], [x(1,1) x(40,2)]);
291         myfit3 = line([Thickness(1,3) Thickness(40,3)], [x(1,1) x(40,3)]);
292         myfit4 = line([Thickness(1,4) Thickness(40,4)], [x(1,1) x(40,4)]);
293         myfit5 = line([Thickness(41,1) Thickness(end,1)], [x(41,1) x(end,1)]);
294         myfit6 = line([Thickness(41,2) Thickness(end,2)], [x(41,1) x(end,2)]);
295         myfit7 = line([Thickness(41,3) Thickness(end,3)], [x(41,1) x(end,3)]);
296         myfit8 = line([Thickness(41,4) Thickness(end,4)], [x(41,1) x(end,4)]);
297         hXLabel = xlabel('\it{d} \rm{/ m}'                    );
298         hYLabel = ylabel('\it{x} '                );
299         xlim([0 Thickness(end,end)])
300         ylim([0.11 0.21])
301         set(myfit1                          , ...
302         'LineWidth'       , 4           );
303         set(myfit2                          , ...
304         'LineWidth'       , 4             , ...
305         'Color'           , 'r'           );
306         set(myfit3                          , ...
307         'LineWidth'       , 4           , ...
308         'Color'           , 'g'           );
309         set(myfit4                          , ...
310         'LineWidth'       , 4           , ...
311         'Color'           , 'k'           );
312         set(myfit5                          , ...
313         'LineWidth'       , 4           );
314         set(myfit6                          , ...
315         'LineWidth'       , 4             , ...
316         'Color'           , 'r'           );
317         set(myfit7                          , ...
318         'LineWidth'       , 4           , ...
319         'Color'           , 'g'           );
320         set(myfit8                          , ...
321         'LineWidth'       , 4           , ...
322         'Color'           , 'k'           );
323
324
325
326         hLegend = legend( ...
```

```
        [myfit1, myfit2, myfit3, myfit4], ...
        'Current density of 200 \rm{A/m^{2}}', ...
        'Current density of 500 \rm{A/m^{2}}', ...
        'Current density of 1000 \rm{A/m^{2}}', ...
        'Current density of 5000 \rm{A/m^{2}}', ...
       'location', 'NorthWest');

         set( gca                       , ...
         'FontName'   , 'Helvetica', ...
         'FontSize',45);
         set([hXLabel, hYLabel], ...
         'FontName'   , 'AvantGarde');
         set(hLegend                , ...
         'FontSize'   , 30           );
         set([hXLabel, hYLabel]  , ...
         'FontSize'   , 45           );

        set(gca, ...
        'Box'         , 'off'     , ...
        'TickDir'     , 'out'     , ...
        'TickLength'  , [.04 .04] , ...
        'XMinorTick'  , 'on'      , ...
        'YMinorTick'  , 'on'      , ...
        'YGrid'       , 'on'      , ...
        'XColor'      , [.3 .3 .3], ...
        'YColor'      , [.3 .3 .3], ...
        'LineWidth'   , 1         );


        %Plotting the measurable heat flux

                  figure('Units', 'pixels', ...
        'Position', [100 100 500 375]);
        hold on
        myfit1 = line(Thickness(:,1), Jq(:,1));
        myfit2 = line(Thickness(:,2), Jq(:,2));
        myfit3 = line(Thickness(:,3), Jq(:,3));
        myfit4 = line(Thickness(:,4), Jq(:,4));
        hXLabel = xlabel('\it{d} \rm{/ m}'                          );
        hYLabel = ylabel('\it{J_{q}^{`}} \rm{/ W/m^{2}}'                  );
        xlim([0 Thickness(end,end)])
```

```
368         ylim([-5000  13000])
369         set(myfit1                              , ...
370         'LineWidth'         , 4             );
371         set(myfit2                              , ...
372         'LineWidth'         , 4               , ...
373         'Color'             , 'r'              );
374         set(myfit3                              , ...
375         'LineWidth'         , 4             , ...
376         'Color'             , 'g'              );
377         set(myfit4                              , ...
378         'LineWidth'         , 4             , ...
379         'Color'             , 'k'              );
380
381
382
383         hLegend = legend( ...
384        [myfit1, myfit2, myfit3, myfit4], ...
385        'Current density of 200 \rm{A/m^{2}}' , ...
386        'Current density of 500 \rm{A/m^{2}}' , ...
387        'Current density of 1000 \rm{A/m^{2}}' , ...
388        'Current density of 5000 \rm{A/m^{2}}' , ...
389       'location', 'NorthWest' );
390
391          set( gca                              , ...
392          'FontName'    , 'Helvetica', ...
393          'FontSize',45);
394          set([hXLabel, hYLabel], ...
395          'FontName'    , 'AvantGarde');
396          set(hLegend                 , ...
397          'FontSize'    , 30             );
398          set([hXLabel, hYLabel]   , ...
399          'FontSize'    , 45             );
400
401         set(gca, ...
402         'Box'           , 'off'      , ...
403         'TickDir'       , 'out'      , ...
404         'TickLength'    , [.04  .04] , ...
405         'XMinorTick'    , 'on'       , ...
406         'YMinorTick'    , 'on'       , ...
407         'YGrid'         , 'on'       , ...
408         'XColor'        , [.3  .3  .3], ...
```

```
409         'YColor'       , [.3 .3 .3], ...
410         'LineWidth'    , 1          );
411
412
413         %Plotting entropy production
414
415                   figure('Units', 'pixels', ...
416         'Position', [100 100 500 375]);
417         hold on
418         myfit1 = line(Thickness(:,1), S(:,1));
419         myfit2 = line(Thickness(:,2), S(:,2));
420         myfit3 = line(Thickness(:,3), S(:,3));
421         myfit4 = line(Thickness(:,4), S(:,4));
422         hXLabel = xlabel('\it{d} \rm{/ m}'                          );
423         hYLabel = ylabel('\it{\sigma} \rm{/ W/m^{2}K}'                  );
424         xlim([0 Thickness(end,end)])
425         ylim([0 25])
426         set(myfit1                          , ...
427         'LineWidth'        , 4           );
428         set(myfit2                            , ...
429         'LineWidth'        , 4              , ...
430         'Color'            , 'r'             );
431         set(myfit3                            , ...
432         'LineWidth'        , 4           , ...
433         'Color'            , 'g'             );
434         set(myfit4                            , ...
435         'LineWidth'        , 4           , ...
436         'Color'            , 'k'             );
437
438
439
440         hLegend = legend( ...
441        [myfit1, myfit2, myfit3, myfit4], ...
442        'Current density of 200 \rm{A/m^{2}}' , ...
443        'Current density of 500 \rm{A/m^{2}}' , ...
444        'Current density of 1000 \rm{A/m^{2}}' , ...
445        'Current density of 5000 \rm{A/m^{2}}' , ...
446       'location', 'NorthWest' );
447
448          set( gca                          , ...
449          'FontName'    , 'Helvetica', ...
```

```
450          'FontSize',45);
451          set([hXLabel, hYLabel], ...
452          'FontName'    , 'AvantGarde');
453          set(hLegend                , ...
454          'FontSize'    , 30          );
455          set([hXLabel, hYLabel]   , ...
456          'FontSize'    , 45          );
457
458         set(gca, ...
459         'Box'          , 'off'      , ...
460         'TickDir'      , 'out'      , ...
461         'TickLength'   , [.04 .04] , ...
462         'XMinorTick'   , 'on'       , ...
463         'YMinorTick'   , 'on'       , ...
464         'YGrid'        , 'on'       , ...
465         'XColor'       , [.3 .3 .3], ...
466         'YColor'       , [.3 .3 .3], ...
467         'LineWidth'    , 1          );
468
469
470
471        %Plotting the polarization curve
472
473                  figure('Units', 'pixels', ...
474         'Position', [100 100 500 375]);
475         hold on
476         myfit1 = line(j, phi);
477         myfit2 = line([0 j(end)],...
478             [1.17 1.17]);
479         hXLabel = xlabel('\it{j} \rm{/ A/m^{2}}'                    );
480         hYLabel = ylabel('\it{E_{cell}} \rm{/ V}'                 );
481         set(myfit1                              , ...
482         'LineWidth'        , 4            );
483         set(myfit2                              , ...
484         'LineWidth'        , 4              ,...
485         'Color'            , 'r'          );
486
487
488
489         hLegend = legend( ...
490        [myfit1, myfit2], ...
```

```
491         'Cell potential from simulation ' , ...
492         'Theoretical cell potential in the reversible limit'                , ...
493        'location', 'NorthWest' );
494
495           set( gca                          , ...
496           'FontName'    , 'Helvetica', ...
497           'FontSize',45);
498           set([hXLabel, hYLabel], ...
499           'FontName'    , 'AvantGarde');
500           set(hLegend                , ...
501           'FontSize'    , 30            );
502           set([hXLabel, hYLabel]   , ...
503           'FontSize'    , 45            );
504
505          set(gca, ...
506          'Box'          , 'off'      , ...
507          'TickDir'      , 'out'      , ...
508          'TickLength'   , [.04 .04] , ...
509          'XMinorTick'   , 'on'       , ...
510          'YMinorTick'   , 'on'       , ...
511          'YGrid'        , 'on'       , ...
512          'XColor'       , [.3 .3 .3], ...
513          'YColor'       , [.3 .3 .3], ...
514          'LineWidth'    , 1          );
515
516
517
518          %Plotting the heat flux divided by j/F
519
520                    figure('Units', 'pixels', ...
521          'Position', [100 100 500 375]);
522          hold on
523          myfit1 = line(j, Jqja);
524          myfit2 = [];
525          hXLabel = xlabel('\it{j} \rm{/ A/m^{2}}'                          );
526          hYLabel = ylabel('\it{J_{q}^{`,a}/(j/F)} \rm{/ J/mol}'
    );
527          set(myfit1                                , ...
528          'LineWidth'         , 4            );
529          set(myfit2                                , ...
530          'LineWidth'         , 4            );
```

```
531
532           set( gca                          , ...
533           'FontName'    , 'Helvetica', ...
534           'FontSize',45);
535           set([hXLabel, hYLabel], ...
536           'FontName'    , 'AvantGarde');
537           set([hXLabel, hYLabel]   , ...
538           'FontSize'    , 45           );
539
540          set(gca, ...
541          'Box'          , 'off'      , ...
542          'TickDir'      , 'out'      , ...
543          'TickLength'   , [.04 .04] , ...
544          'XMinorTick'   , 'on'       , ...
545          'YMinorTick'   , 'on'       , ...
546          'YGrid'        , 'on'       , ...
547          'XColor'       , [.3 .3 .3], ...
548          'YColor'       , [.3 .3 .3], ...
549          'LineWidth'    , 1          );
550
551
552
553
554         %Plotting the heat flux divided by j/F
555
556                    figure('Units', 'pixels', ...
557          'Position', [100 100 500 375]);
558          hold on
559          myfit1 = line(j, Jqjc);
560          myfit2 = [];
561          hXLabel = xlabel('\it{j} \rm{/ A/m^{2}}'                          );
562          hYLabel = ylabel('\it{J_{q}^{',c}/(j/F)} \rm{/ J/mol}'
    );
563          set(myfit1                              , ...
564          'LineWidth'        , 4           );
565          set(myfit2                              , ...
566          'LineWidth'        , 4           );
567
568
569           set( gca                          , ...
570           'FontName'    , 'Helvetica', ...
```

```
'FontSize',45);
set([hXLabel, hYLabel], ...
'FontName'   , 'AvantGarde');
set([hXLabel, hYLabel]  , ...
'FontSize'   , 45          );

set(gca, ...
'Box'         , 'off'     , ...
'TickDir'     , 'out'     , ...
'TickLength'  , [.04 .04] , ...
'XMinorTick'  , 'on'      , ...
'YMinorTick'  , 'on'      , ...
'YGrid'       , 'on'      , ...
'XColor'      , [.3 .3 .3], ...
'YColor'      , [.3 .3 .3], ...
'LineWidth'   , 1         );


%Plotting the mass flux divided by j/F

          figure('Units', 'pixels', ...
'Position', [100 100 500 375]);
hold on
myfit1 = line(j, Jwaj);
myfit2 = [];
hXLabel = xlabel('\it{j} \rm{/ A/m^{2}}'                    );
hYLabel = ylabel('\it{J_{w}^{a}/(j/F)}'                  );
set(myfit1                          , ...
'LineWidth'       , 4           );
set(myfit2                          , ...
'LineWidth'       , 4           );

 set( gca                           , ...
 'FontName'   , 'Helvetica', ...
 'FontSize',45);
 set([hXLabel, hYLabel], ...
 'FontName'   , 'AvantGarde');
 set([hXLabel, hYLabel]  , ...
 'FontSize'   , 45          );

set(gca, ...
```

```
612         'Box'         , 'off'      , ...
613         'TickDir'     , 'out'      , ...
614         'TickLength'  , [.04 .04]  , ...
615         'XMinorTick'  , 'on'       , ...
616         'YMinorTick'  , 'on'       , ...
617         'YGrid'       , 'on'       , ...
618         'XColor'      , [.3 .3 .3] , ...
619         'YColor'      , [.3 .3 .3] , ...
620         'LineWidth'   , 1          );
621
622         %Plotting the mass flux divided by j/F
623
624                   figure('Units', 'pixels', ...
625         'Position', [100 100 500 375]);
626         hold on
627         myfit1 = line(j, Jwcj);
628         myfit2 = [];
629         hXLabel = xlabel('\it{j} \rm{/ A/m^{2}}'                    );
630         hYLabel = ylabel('\it{J_{w}^{c}/(j/F)}'                  );
631         set(myfit1                              , ...
632         'LineWidth'       , 4           );
633         set(myfit2                              , ...
634         'LineWidth'       , 4           );
635
636
637
638          set( gca                       , ...
639          'FontName'    , 'Helvetica', ...
640          'FontSize',45);
641          set([hXLabel, hYLabel], ...
642          'FontName'    , 'AvantGarde');
643          set([hXLabel, hYLabel]   , ...
644          'FontSize'    , 45          );
645
646         set(gca, ...
647         'Box'         , 'off'      , ...
648         'TickDir'     , 'out'      , ...
649         'TickLength'  , [.04 .04]  , ...
650         'XMinorTick'  , 'on'       , ...
651         'YMinorTick'  , 'on'       , ...
652         'YGrid'       , 'on'       , ...
```

```
653             'XColor'        , [.3  .3  .3], ...
654             'YColor'        , [.3  .3  .3], ...
655             'LineWidth'     , 1           );
656
657 end
```

```
%-------------------------------------------------------------------
% Main subroutine that calls on the smaller subroutine of
% each segment in the fuel cell. Plotting the profiles
% in the end of the code.
%-------------------------------------------------------------------

function [res,phic,Jqa,Jqc,Jwa,Jwc,Xvector,Yvector] = Transport(Jqa, plot)


    global Const;

    Jqa0 = Jqa;

    %Finding the correct water flux for the given heat flux
    [Jw0, res, EXITFLAG] = fzero('Jw',0,[],Jqa0);

    if EXITFLAG ~= 1
        error('No appropriate value of Jw found')
    end

    Const.Jwm = Jw0;

    Const.Jwa = Const.Jwm;
    Jwa = Const.Jwa;
    %[kg/m^s s]    % Water flux in anode and membrane
    Const.Jwc = Const.Jwa + Const.j/(2*Const.F);
    Jwc = Const.Jwc;
    %[kg/m^s s]    % Water flux in catode
    Const.twc = ((Const.Jwa/(Const.j/Const.F))+0.5);
    twc = Const.twc;
    %[Unitless]    % Transport number for water in the cathode

    %Calculating the saturation prsesure and mole fraction of
    %water at the start of the anode backing
    pw = ps(Const.T0);
    xw0 = pw/(Const.p);

    %Definition of the half cell potential of hydrogen and
    %zero entropy at the start of the anode backing
    phi0 = 0;
    s0 = 0;
```

```
42
43      %Boundary conditions at the start of the anode backing
44      ya0 = [Const.T0, phi0, xw0, Jqa0, s0];
45
46      %Length of the different segments in the fuel cell
47      xanode = linspace(0, Const.ds, 40);
48      xmembrane = linspace(Const.ds, Const.ds+Const.dm, 40);
49      xcathode = linspace(Const.ds+Const.dm,Const.ds*2+...
50          Const.dm,40);
51
52      %Calculating the variations of the unknown parameters in the
53      %anode backing
54      [Xanode, Yanode] = ode15s('anode',xanode,ya0);
55
56      %Calculating the jumps of the unknown parameters in the
57      %anode surface
58      [Yanodesurf, dS(2)] = anodesurf(Yanode(end,:));
59
60      %Procedure to find the activity at the start of the membrane
61      lambda2 = 14;
62
63      lambda1 = lambda2 - ((Const.twm*(Const.j/Const.F)-...
64          Const.Jwm)*Const.Mm/(Const.Dm*Const.rodry))*Const.dm;
65
66      a1 = activity(lambda1);
67
68      for i = 1:length(a1)
69          if a1(i) < 1
70              a = a1(i);
71          end
72      end
73
74      %Boundary conditions at the start of the membrane
75      Tm0 = Yanodesurf(1);
76      phim0 = Yanodesurf(2);
77      am0 = a;
78      Jqm0 = Yanodesurf(3);
79      sm0 = Yanode(end,5)+dS(2);
80
81      ym0 = [Tm0, phim0, am0, Jqm0, sm0];
82
```

```
83      %Calculating the variations of the unknown parameters in the
84      %membrane
85      [Xmembrane, Ymembrane] = ode15s('membrane',xmembrane,ym0);
86
87      for k = 1:length(Xmembrane)
88          chi(k) = Xi(Ymembrane(k,3));
89      end
90
91      %Calculating the jumps of the unknown parameters in the
92      %cathode surface
93      [Ycatodesurf, dS(4)] = cathodesurf(Ymembrane(end,:));
94
95      %Boundary conditions at the start of the cathode
96      Tc0 = Ycatodesurf(1);
97      phic0 = Ycatodesurf(2);
98      Jqc0 = Ycatodesurf(3);
99      xO20 = Const.xO20 - Const.j/(4*Const.F*Const.DON)*Const.ds;
100     sc0 = Ymembrane(end,5)+dS(4);
101
102     yc0 = [Tc0, phic0, xO20, Jqc0, sc0];
103
104     %Calculating the variations of the unknown parameters in the
105     %cathode
106     [Xcathode, Ycatode] = ode15s('Cathode',xcathode,yc0);
107
108     Tc = Ycatode(end,1);
109     phic = Ycatode(end,2);
110     Jqc = Ycatode(end,4);
111
112     %%%%Total energy balance%%%%%
113
114     Ein = Const.JH2*(Const.cph2*(Const.T0-Const.Tq))+...
115         Const.Jwa*(Const.Hwgas+Const.cpwgas*...
116         (Const.T0-Const.Tq))+Jqa0;
117
118     Eut = Const.JO2*(Const.cpo2*(Tc-Const.Tq))+Const.Jwc*...
119         (Const.Hwgas+Const.cpwgas*(Tc-Const.Tq))+Jqc+...
120         Const.j*phic;
121
122     res(1) = Eut - Ein;
123
```

```
124     %%%%%%%%%%ENTROPY%%%%%%%%%%%%%%%
125
126     for i = 1:length(Xanode)
127         [extra, sigma1(i)] = anode(Xanode(i), Yanode(i,:));
128     end
129     dS(1) = trapz(Xanode,sigma1)*(Const.ds);
130
131     for i = 1:length(Xmembrane)
132         [extra, sigma2(i)] = membrane(Xmembrane(i),...
133             Ymembrane(i,:));
134     end
135     dS(3) = trapz(Xmembrane,sigma2)*(Const.ds+Const.dm);
136
137     for i = 1:length(Xcathode)
138         [extra, sigma3(i)] = Cathode(Xcathode(i), Ycatode(i,:));
139     end
140     dS(5) = trapz(Xcathode,sigma3)*(2*Const.ds+Const.dm);
141
142     dStot = sum(dS);
143
144     Xvector = [Xanode;Xmembrane;Xcathode];
145     Yvector = [Yanode;Ymembrane;Ycatode];
146
147
148
149
150     %%%%%%%%%%%%%%%%%%TOTAL ENTROPY BALANCE%%%%%%%%%%
151
152     Swc = Const.Swgas+Const.cpwgas*log(Tc/Const.Tq);
153     SO2 = Const.So2+Const.cpo2*log(Tc/Const.Tq)-Const.R*log(0.21);
154     Swa = Const.Swgas+Const.cpwgas*log(Const.T0/Const.Tq);
155     SH2 = Const.Sh2+Const.cph2*log(Const.T0/Const.Tq);
156
157     dStotal = (Jqc/Tc)-(Jqa0/Const.T0)+(Const.j/Const.F)*...
158         (0.5*(Swc-SH2)-0.25*SO2)+Const.Jwa*(Swc-Swa);
159
160     res(2) = dStotal - dStot;
161
162
163
164
```

```
165     if plot == 1
166
167         %Plotting the temperature profile through each segment
168
169          figure('Units', 'pixels', ...
170         'Position', [100 100 500 375]);
171         hold on
172         myfit1 = line(Xanode, Yanode(:,1));
173         myfit2 = line([Xanode(end) Xanode(end)],...
174             [Yanode(end,1) Yanodesurf(1)]);
175         myfit3 = line(Xmembrane, Ymembrane(:,1));
176         myfit4 = line([Xmembrane(end) Xmembrane(end)],...
177             [Ymembrane(end,1) Ycatodesurf(1)]);
178         myfit5 = line(Xcathode, Ycatode(:,1));
179         hXLabel = xlabel('\it{d} \rm{/ m}'                    );
180         hYLabel = ylabel('\it{T} \rm{/ K}'                 );
181         xlim([0 xcathode(end)])
182         set(myfit1                          , ...
183         'LineWidth'         , 4             );
184         set(myfit2                          , ...
185         'LineWidth'         , 4             );
186         set(myfit3                          , ...
187         'LineWidth'         , 4             );
188         set(myfit4                          , ...
189         'LineWidth'         , 4             );
190         set(myfit5                          , ...
191         'LineWidth'         , 4             );
192
193
194          set( gca                           , ...
195          'FontName'    , 'Helvetica', ...
196          'FontSize',45);
197          set([hXLabel, hYLabel], ...
198          'FontName'    , 'AvantGarde');
199          set([hXLabel, hYLabel]   , ...
200          'FontSize'    , 45            );
201
202
203         set(gca, ...
204         'Box'           , 'off'      , ...
205         'TickDir'       , 'out'      , ...
```

```
206         'TickLength'   , [.04  .04] , ...
207         'XMinorTick'   , 'on'       , ...
208         'YMinorTick'   , 'on'       , ...
209         'YGrid'        , 'on'       , ...
210         'XColor'       , [.3  .3  .3], ...
211         'YColor'       , [.3  .3  .3], ...
212         'LineWidth'    , 1          );
213
214
215
216         %Plotting the mole fraction profile through anode and
217         %cathode backing
218
219          figure('Units', 'pixels', ...
220         'Position', [100  100  500  375]);
221         hold on
222         myfit1 = line(Xanode, Yanode(:,3));
223         myfit2 = line(Xcathode, Ycatode(:,3));
224         hXLabel = xlabel('\it{d} \rm{/ m}'                    );
225         hYLabel = ylabel('\it{x}'                 );
226         xlim([0 xcathode(end)])
227         set(myfit1                              , ...
228         'LineWidth'        , 4            );
229         set(myfit2                              , ...
230         'LineWidth'        , 4            );
231
232          set( gca                          , ...
233          'FontName'    , 'Helvetica', ...
234          'FontSize',45);
235          set([hXLabel, hYLabel], ...
236          'FontName'    , 'AvantGarde');
237          set([hXLabel, hYLabel]   , ...
238          'FontSize'    , 45           );
239
240         set(gca, ...
241         'Box'          , 'off'      , ...
242         'TickDir'      , 'out'      , ...
243         'TickLength'   , [.04  .04] , ...
244         'XMinorTick'   , 'on'       , ...
245         'YMinorTick'   , 'on'       , ...
246         'YGrid'        , 'on'       , ...
```

```
247         'XColor'       , [.3  .3  .3], ...
248         'YColor'       , [.3  .3  .3], ...
249         'LineWidth'    , 1           );
250
251
252
253         %Plotting the electrical potential profile through each segment
254
255           figure('Units', 'pixels', ...
256         'Position', [100  100  500  375]);
257         hold on
258         myfit1 = line(Xanode, Yanode(:,2));
259         myfit2 = line([Xanode(end) Xanode(end)],...
260             [Yanode(end,2) Yanodesurf(2)]);
261         myfit3 = line(Xmembrane, Ymembrane(:,2));
262         myfit4 = line([Xmembrane(end) Xmembrane(end)],...
263             [Ymembrane(end,2) Ycatodesurf(2)]);
264         myfit5 = line(Xcathode, Ycatode(:,2));
265         hXLabel = xlabel('\it{d} \rm{/ m}'                    );
266         hYLabel = ylabel('\it{\phi} \rm{/ V}'                 );
267         xlim([0 xcathode(end)])
268         set(myfit1                              , ...
269         'LineWidth'        , 4              );
270         set(myfit2                              , ...
271         'LineWidth'        , 4              );
272         set(myfit3                              , ...
273         'LineWidth'        , 4              );
274         set(myfit4                              , ...
275         'LineWidth'        , 4              );
276         set(myfit5                              , ...
277         'LineWidth'        , 4              );
278
279
280          set( gca                               , ...
281          'FontName'    , 'Helvetica', ...
282          'FontSize',45);
283          set([hXLabel, hYLabel], ...
284          'FontName'    , 'AvantGarde');
285          set([hXLabel, hYLabel]   , ...
286          'FontSize'    , 45             );
287
```

```
288         set(gca, ...
289         'Box'         , 'off'      , ...
290         'TickDir'     , 'out'      , ...
291         'TickLength'  , [.04 .04] , ...
292         'XMinorTick'  , 'on'       , ...
293         'YMinorTick'  , 'on'       , ...
294         'YGrid'       , 'on'       , ...
295         'XColor'      , [.3 .3 .3], ...
296         'YColor'      , [.3 .3 .3], ...
297         'LineWidth'   , 1          );
298
299
300         %Plotting the heat flux profile through each segment
301
302               figure('Units', 'pixels', ...
303         'Position', [100 100 500 375]);
304         hold on
305         myfit1 = line(Xanode, Yanode(:,4));
306         myfit2 = line([Xanode(end) Xanode(end)],...
307             [Yanode(end,4) Yanodesurf(3)]);
308         myfit3 = line(Xmembrane, Ymembrane(:,4));
309         myfit4 = line([Xmembrane(end) Xmembrane(end)],...
310             [Ymembrane(end,4) Ycatodesurf(3)]);
311         myfit5 = line(Xcathode, Ycatode(:,4));
312         hXLabel = xlabel('\it{d} \rm{/ m}'                          );
313         hYLabel = ylabel('\it{J_{q}^{t}} \rm{/ W/m^{2}}'                  );
314         xlim([0 xcathode(end)])
315         set(myfit1                          , ...
316         'LineWidth'       , 4           );
317         set(myfit2                          , ...
318         'LineWidth'       , 4           );
319         set(myfit3                          , ...
320         'LineWidth'       , 4           );
321         set(myfit4                          , ...
322         'LineWidth'       , 4           );
323         set(myfit5                          , ...
324         'LineWidth'       , 4           );
325
326          set( gca                           , ...
327          'FontName'    , 'Helvetica', ...
328          'FontSize',50);
```

```
329          set([hXLabel, hYLabel], ...
330          'FontName'    , 'AvantGarde');
331          set([hXLabel, hYLabel]   , ...
332          'FontSize'    , 45           );
333
334         set(gca, ...
335         'Box'          , 'off'      , ...
336         'TickDir'      , 'out'      , ...
337         'TickLength'   , [.04 .04] , ...
338         'XMinorTick'   , 'on'       , ...
339         'YMinorTick'   , 'on'       , ...
340         'YGrid'        , 'on'       , ...
341         'XColor'       , [.3 .3 .3], ...
342         'YColor'       , [.3 .3 .3], ...
343         'LineWidth'    , 1          );
344
345
346         %Plotting the accumulated entropy production profile
347         %through each segment
348
349                   figure('Units', 'pixels', ...
350         'Position', [100 100 500 375]);
351         hold on
352         myfit1 = line(Xanode, Yanode(:,5));
353         myfit2 = line([Xanode(end) Xanode(end)],...
354             [Yanode(end,5) Ymembrane(1,5)]);
355         myfit3 = line(Xmembrane, Ymembrane(:,5));
356         myfit4 = line([xmembrane(end) Xmembrane(end)],...
357             [Ymembrane(end,5) Ycatode(1,5)]);
358         myfit5 = line(Xcathode, Ycatode(:,5));
359         hXLabel = xlabel('\it{d} \rm{/ m}'                          );
360         hYLabel = ylabel('\it{\sigma} \rm{/ W/K m^{2}}'                    );
361         xlim([0 xcathode(end)])
362         set(myfit1                          , ...
363         'LineWidth'        , 4           );
364         set(myfit2                          , ...
365         'LineWidth'        , 4           );
366         set(myfit3                          , ...
367         'LineWidth'        , 4           );
368         set(myfit4                          , ...
369         'LineWidth'        , 4           );
```

```
370         set(myfit5                              , ...
371         'LineWidth'         , 4             );
372
373          set( gca                           , ...
374          'FontName'     , 'Helvetica', ...
375          'FontSize',50);
376          set([hXLabel, hYLabel], ...
377          'FontName'     , 'AvantGarde');
378          set([hXLabel, hYLabel]   , ...
379          'FontSize'     , 45            );
380
381         set(gca, ...
382         'Box'           , 'off'      , ...
383         'TickDir'       , 'out'      , ...
384         'TickLength'    , [.04 .04] , ...
385         'XMinorTick'    , 'on'       , ...
386         'YMinorTick'    , 'on'       , ...
387         'YGrid'         , 'on'       , ...
388         'XColor'        , [.3 .3 .3], ...
389         'YColor'        , [.3 .3 .3], ...
390         'LineWidth'     , 1          );
391
392
393
394         %Plotting the water content in the membrane
395
396                   figure('Units', 'pixels', ...
397         'Position', [100 100 500 375]);
398         hold on
399         myfit1 = line(Xmembrane, chi);
400         hXLabel = xlabel('\it{d} \rm{/ m}'                          );
401         hYLabel = ylabel('\it{\chi} \rm{/ H_{2}O per sulfonic site}'
    );
402         xlim([Xanode(end),Xmembrane(end)])
403         set(myfit1                              , ...
404         'LineWidth'         , 4             );
405
406
407          set( gca                           , ...
408          'FontName'     , 'Helvetica', ...
409          'FontSize',50);
```

```
410             set([hXLabel, hYLabel], ...
411             'FontName'    , 'AvantGarde');
412             set([hXLabel, hYLabel]   , ...
413             'FontSize'    , 45           );
414
415
416           set(gca, ...
417           'Box'           , 'off'      , ...
418           'TickDir'       , 'out'      , ...
419           'TickLength'    , [.04 .04] , ...
420           'XMinorTick'    , 'on'       , ...
421           'YMinorTick'    , 'on'       , ...
422           'YGrid'         , 'on'       , ...
423           'XColor'        , [.3 .3 .3], ...
424           'YColor'        , [.3 .3 .3], ...
425           'LineWidth'     , 1          );
426
427
428       end
429
430   end
```

```
%%%%%%%%%%%%%%%%ANODE%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    function [dy, Sigma] = anode(x,y)

        global Const

        T = y(1);
        phi = y(2);
        xw = y(3);
        Jq = y(4);
        sigma = y(5);

        %The parameters used in the calculations for the anode backing
        xH2 = 1-xw;
        Jda = ((Const.Jwa/xw)-(Const.JH2/xH2))*xw;
        Sh2 = Const.Sh2 + Const.cph2*log(T/Const.Tq);
        Se = -2;
        Swg = Const.Swgas + Const.cpwgas*log(T/Const.Tq);
        pelta = T*(-0.5*Sh2-Se);
        qa = -T*Swg;

        %The differential equations in the anode backing
        dTdx = -(1/Const.LsAA)*(Jq - qa*(Jda - Const.tad*...
             (Const.j/Const.F))-pelta*(Const.j/Const.F));
        dxwdx = -((qa*xw)/(Const.R*T*T))*dTdx - (1/Const.Dwh)*...
            (Jda-Const.tad*(Const.j/Const.F));
        dphidx = -(pelta/(T*Const.F))*dTdx-((Const.tad*Const.R*T)/...
            (Const.F*xw))*dxwdx-Const.rac*Const.j;
        dJqdx = -Const.j*dphidx-Const.JH2*Const.cph2*dTdx-Const.Jwa*...
            Const.cpwgas*dTdx;

        dsigmadx = -(Jq/(T*T))*dTdx-Const.R*Jda*(1/xw)*dxwdx-Const.j*...
            (1/T)*dphidx;

        dy = [dTdx
            dphidx
            dxwdx
            dJqdx
            dsigmadx];

        if nargout > 1
```

```
42                 Sigma = -(Jq/(T*T))*dTdx-Const.R*Jda*(1/xw)*dxwdx-Const.j*...
43                     (1/T)*dphidx;
44             end
45
46         end
```

```
%%%%%%%%%%%%ANODE SURFACE CALCULATION%%%%%%%%%%%%%%%%

    function [Ys, Sigma] = anodesurf(Ystart)

        global Const

        Tas = Ystart(1);
        phias = Ystart(2);
        xwas = Ystart(3);
        Jqas = Ystart(4);

        %Parameters used in the calculation of the anode surface
        Hh2 = Const.cph2*(Tas-Const.Tq);
        Hw = Const.Hwgas+Const.cpwgas*(Tas-Const.Tq);
        Sh2 = Const.Sh2 + Const.cph2*log(Tas/Const.Tq);
        pelta = Tas*(-0.5*Sh2-Const.Se);
        qa = - Tas*(Const.Swgas+Const.cpwgas*log(Tas/Const.Tq));
        Hwm = Const.Hwm+Const.cpw*(Tas-Const.Tq);
        qm = - Tas*(Const.Swm+Const.cpw*log(Tas/Const.Tq));
        Swm = Const.Swm + Const.cpw*log(Tas/Const.Tq);
        Shm = Const.Sh + Const.cpH*log(Tas/Const.Tq);

        DeltaG = -(Hh2-Tas*Sh2);


        %Temperature drops
        DeltaasT = (-Jqas+qa*(Const.Jwa-Const.twm*Const.j/Const.F)+...
            pelta*Const.j/Const.F)/Const.Ls;
        DeltasmT = 0;          %Assumption

        %Electrical potential drop
        Deltaphieff = -(pelta/(Tas*Const.F))*DeltaasT-Const.rs*Const.j;
        Deltaphi = Deltaphieff + (1/(2*Const.F))*(Hh2-Tas*Sh2);

        Hwvap = Const.Hwvap + (Const.cpwgas - Const.cpw)*(Tas-Const.Tq);

        %Energy balance
        Jqm = Jqas - Const.j*Deltaphi + Const.Jwm*Hwvap+Const.JH2*Hh2;

        %Output values at the anode surface
        Ts = Tas + DeltaasT;
```

```
42          Tsm = Ts + DeltasmT;
43          phim = phias + Deltaphi;
44
45          Sigma = Jqas*((1/Ts)-(1/Tas))-(Const.j/Ts)*Deltaphieff;
46
47          Ys = [Tsm phim Jqm];
48
49      end
```

```
%%%%%%%%%%MEMBRANE CALCULATION%%%%%%%%%%%%%%%

    function [dy,Sigma] = membrane(x,y)

        global Const

        T = y(1);
        phi = y(2);
        a = y(3);
        Jqm = y(4);
        sigmam = y(5);

        %Parameters that are used in the membrane calculation
        Swm = Const.Swm + Const.cpw*log(T/Const.Tq);
        Sh = Const.Sh+Const.cpH*log(T/Const.Tq);
        peltm = T*(Sh - Const.twm*Swm);
        qm = -T*Swm;
        rm = 1/(exp(1268*((1/303)-(1/T))*(0.5139*Xi(a)-0.326)));
        Lm = 0.177+3.7e-3*Xi(a);

        %Differential equations in the membrane
        dTdx = -(Jqm/Lm)+(qm/Lm)*(Const.Jwm-Const.twm*...
            (Const.j/Const.F))+peltm*Const.j/(Lm*Const.F);
        dadx = -(Const.Jwm-Const.j*(Const.twm/Const.F))*Const.Mm/...
            (Xia(a)*Const.rodry*Const.Dm);
        dphidx = -(peltm/(T*Const.F))*dTdx-((Const.twm*Const.R*T)/...
            (Const.F*a))*dadx-rm*Const.j;
        dJqdx = -Const.j*dphidx-Const.Jwm*Const.cpw*dTdx;

        dsigmadx = -(Jqm/(T*T))*dTdx-Const.R*Const.Jwm*(1/a)*...
            dadx-Const.j*(1/T)*dphidx;

        dy = [dTdx
            dphidx
            dadx
            dJqdx
            dsigmadx];

        if nargout > 1
            Sigma = -(Jqm/(T*T))*dTdx-Const.R*Const.Jwm*(1/a)*...
                dadx-Const.j*(1/T)*dphidx;
```

```
42                end
43
44            end
```

```
%%%%%%%%%%%%CATHODE SURFACE CALCULATION%%%%%%%%%%%%%%%%

function [Yout, Sigma] =cathodesurf(Yin)

    global Const

    Tms = Yin(1);
    phim = Yin(2);
    awm = Yin(3);
    Jqm = Yin(4);

    %Parameters used in the cathode surface calculation
    Swc = Const.Swgas + Const.cpwgas*log(Tms/Const.Tq);
    Swm = Const.Swm + Const.cpw*log(Tms/Const.Tq);
    Sh = Const.Sh +Const.cpH*log(Tms/Const.Tq);
    SO2 = Const.So2 + Const.cpo2*log(Tms/Const.Tq)-Const.R*log(0.21);
    pic = Tms*(0.25*SO2-Const.Se-((Const.Jwa/(Const.j/Const.F))+0.5)*Swc);
    qc = -Tms*Swc;

    DeltaG = (1/(2*Const.F))*(Const.Hwgas+Const.cpwgas*...
        (Tms-Const.Tq)-Tms*(Const.Swgas+Const.cpwgas*...
        log(Tms/Const.Tq)))-(1/(4*Const.F))*(Const.cpo2*...
        (Tms-Const.Tq)-Tms*Const.cpo2*log(Tms/Const.Tq));
    Overpot = (2*Const.R*Tms/Const.F)*log(Const.j/Const.j0);


    %Calculation part to find the measurable heat flux in the cathode
    %backing at the surface
    Energy = Jqm+Const.Jwm*Const.Hwm-Const.Jwc*Const.Hwgas-Const.JO2*...
        Const.cpo2*(Tms-Const.Tq);

    Energy2 = Energy-Const.j*(-Overpot+(1/Const.F)*((-pic/Tms)*...
        (-1/Const.Ls)*(-qc*(Const.Jwc-Const.twc*(Const.j/Const.F))-pic*...
        (Const.j/Const.F))...
        -Const.rs*Const.j)-DeltaG);

    Energy3 = 1-(1/Const.F)*(pic/Tms)*(1/Const.Ls);

    Jqc = Energy2/Energy3;

    %Temperature drop in the cathode surface
```

```
42      DeltascT = -(1/Const.Ls)*(Jqc-qc*(Const.Jwc-Const.twc*...
43          (Const.j/Const.F))-pic*Const.j/(Const.F));
44
45      DeltamsT = 0;           %Assumption
46      Ts = Tms + DeltamsT;
47      Tsc = Ts+DeltascT;
48
49      %Electrical potential drop in the cathode surface
50      Deltaphieff = (1/Const.F)*(-(pic/Tsc)*DeltascT-Const.rs*Const.j)-...
51          Overpot;
52      Deltaphi = Deltaphieff-DeltaG;
53
54      phic = phim+Deltaphi;
55
56      Yout = [Tsc phic Jqc];
57
58      Sigma = Jqc*((1/Tsc)-(1/Tms))-Const.j*(1/Tms)*Deltaphieff;
59
60  end
```

```
%%%%%%%%%%%%%%%CATHODE%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [dy,Sigma] = Cathode(x,y)

    global Const


    T = y(1);
    phi = y(2);
    xO2 = y(3);
    Jq = y(4);
    sigmac = y(5);


    Swc = Const.Swgas + Const.cpwgas*log(T/Const.Tq);
    qc = -T*Swc;
    Hwc = Const.Hwgas + Const.cpwgas*(T-Const.Tq);
    Ho2 = Const.cpo2*(T-Const.Tq);
    SO2 = Const.So2 + Const.cpo2*log(T/Const.Tq)-Const.R*log(0.21);
    peltc = T*(0.25*SO2-Const.Se-((Const.Jwa/(Const.j/Const.F))+...
        0.5)*Swc);


    dTdx = -(1/Const.LsAA)*(Jq - peltc*(Const.j/Const.F));
    dxO2dx = Const.j/(4*Const.F*Const.DON);
    dphidx = (-peltc/(T*Const.F))*dTdx-((Const.R*T)/...
        (4*Const.F*xO2))*dxO2dx-Const.rac*Const.j;
    dJqdx = -Const.j*dphidx-Const.Jwc*Const.cpw*dTdx-Const.JO2*...
        Const.cpo2*dTdx;

    dsigmadx = -(Jq/(T*T))*dTdx-Const.JO2*(1/T)*Const.R*(1/xO2)*...
        dxO2dx-Const.j*(1/T)*dphidx;

    dy = [dTdx
        dphidx
        dxO2dx
        dJqdx
        dsigmadx];

    if nargout > 1
        Sigma = -(Jq/(T*T))*dTdx-Const.JO2*(1/T)*Const.R*(1/xO2)*...
```

```
42                      dxO2dx-Const.j*(1/T)*dphidx;
43              end
44
45          end
```

```
%%%%%Finding the correct water flux in the anode backing/membrane

function res = Jw(Jw0,Jq0)

    global Const

    %Parameters used in the calculation
    pw = ps(Const.T0);
    xw0 = pw/(Const.p);
    Sh2 = Const.Sh2 + Const.cph2*log(Const.T0/Const.Tq);
    Se = -2;
    Swg = Const.Swgas + Const.cpwgas*log(Const.T0/Const.Tq);
    pelta = Const.T0*(-0.5*Sh2-Se);
    qa = -Const.T0*Swg;

    %Finding the temperature at the end of the anode backing
    T1 = (((-Jq0 + pelta*(Const.j/Const.F)+qa*Jw0)*Const.ds)/...
        Const.LsAA) + Const.T0;

    %Saturation pressure at temperature T1
    Psat1 = ps(T1);

    %Mole fraction of water at the end of the anode backing
    xw1 = xw0 - (Jw0*Const.ds/Const.Dwh);

    %Activty at the end of the anode backing
    a1 = Const.p*xw1/Psat1;

    %Water content at the activity a1
    lambda1 = Xi(a1);

    lambda2 = 14;

    Jwm = -(Const.Dm*Const.rodry/Const.Mm)*(lambda2 - lambda1)/...
        Const.dm+Const.twm*(Const.j/Const.F);

    res = Jw0 - Jwm;

end
```

```
%Finding the activity from known water content

function a = activity(lambda)

    if lambda < 0
        disp('Lambda is negative!')
        pause
    end
    if lambda > 16.8
        disp('Lambda is too large!')
        pause
    end

    poly = [36.0 - 39.85 17.81 (0.043 - lambda)];
    r = roots(poly);
    a = [];
    for i = 1:length(r)
        if imag(r(i)) == 0
            a = [a; r(i)];
        end
    end

    if a > 1
        a = (lambda - 14)/1.4 + 1;
        if (a < 1) | (a > 3)
            disp('No physical activity found)')
            pause
        end
    end
end
```

```
% Finding the saturation pressure at temperature T
function f = ps(T)
%
    f = -2846.4 + 411.24*(T-273.15) - 10.554*(T-273.15).^2 +...
        0.16636*(T-273.15).^3;
%
end
```

```
% membrane water content
function xi = Xi(a)
%
    if (length(a) == 1)
    %

        if  (a<1)
            xi = 0.043 + 17.81*a - 39.85*a.^2 + 36.0*a.^3;
        elseif  (a<3)
            xi = 14 + 1.4*(a-1);
        else
            xi = 16.8;
        end
    else
    %
        %cannot use xi = Xi(a) since there is checking of the
        %value of each a inside xi ???
        xi = ones(size(a));
        for i=1:length(a)
            xi(i) = Xi(a(i));
        end
    %
    end
%
end
```

```
1
2
3  % membrane water content derivative
4  function xi = Xia(a)
5  %
6       if (length(a) == 1)
7       %
8            if       (a<0)
9                 xi = 0;
10           elseif   (a<1)
11                xi = 17.81 - 2*39.85*a + 3*36.0*a.^2;
12           elseif   (a<3)
13                xi = 1.4;
14           else
15                xi = 0;
16           end
17           %xi = 1.4;
18           %xi = 17.81 - 2*39.85*a + 3*36.0*a.^2;
19      %
20       else
21      %
22           %cannot use xi = Xi(a) since there is checking of the
23           %value of each a inside xi ???
24           xi = ones(size(a));
25           for i=1:length(a)
26                xi(i) = Xia(a(i));
27           end
28      %
29       end
30 %
31 end
```