Capacitive Storage of Ions Modelled in a Pore Network using Electrical Double Layer Theories: Helmholtz, Gouy-Chapman-Stern, and modified Donnan

by

Michael McKague

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Chemical Engineering

Waterloo, Ontario, Canada, 2021

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Capacitive Deionization (CDI) is an electro-driven water desalination technology that has been researched extensively over the last two decades. A CDI cell consists of a pair of charged porous electrodes separated by a dielectric spacer. The electrodes are typically a pair of porous electrodes where ions are stored within the electrical double layers (EDLs) of micropores. Accurate modelling of capacitive electrodes depends heavily on EDL models. There have been many attempts at modelling CDI but most studies treat the electrode as a continuum with effective properties. Pore structure has been shown to have a significant effect on CDI cell performance and a Pore Network Model (PNM) is one such tool for modeling the effects of pore structure. To the best of the author's knowledge, there is no prior PNM of CDI or any PNM, in general, that incorporates EDL theory at all. Therefore, this work is the first attempt at applying electrical double layer models to a pore network for simulating the storage of ions into micropores. The most common EDL models used in studying CDI are: Helmholtz, Gouy-Chapman-Stern, and modified Donnan. In this work, each of these EDL theories are applied to a network of pores for modelling the storage of ions into micropores. For computational efficiency, only the macropore space was resolved in the pore network and neighbouring micropore regions where ions are electrosorbed was treated with a sub-model. A test 2D network of 294 pores consisting of anode, spacer, and cathode subregions was used to demonstrate the use of developed numerical method equations for modelling charging in a stop flow CDI cell. Other simplifying assumptions made were electroneutrality in the macropores, equal anion and cation diffusivities, and monovalent salt solution. These assumptions reduced our problem to a transient fickian diffusion problem coupled with an ionic transport algorithm. The open-source pore network modelling package called `OpenPNM` was used for pore network modelling and solving our coupled transport equations. Results from `OpenPNM` were validated by comparing to `COMSOL` and the relative error for concentration and potential was always less than 10% and 15% respectively at all times of the simulation. Charging dynamics of the pore network was demonstrated along with a demonstration on how to obtain discharge curves during the flow stage. Lastly, a pore network model was applied to a stop flow CDI cell found in literature. A 3D network of pores was developed and calibrated to have the same pore size distribution, porosity, and permeability as the electrode-spacer assembly in literature. Discharge curves from the calibrated pore network model and literature were found to have good agreement, but it was difficult to model correctly due to likely turbulence in the outlet and the conductivity sensor being downstream of the outflow port. This is the beginning of using pore networks to model EDL based ion storage.

Acknowledgements

Firstly, I would like to acknowledge my supervisor, Dr. Jeffrey Gostick for his weekly commitment to meet with me to help guide my project. From day one, Jeff has always believed in and encouraged me to think of myself as a talented engineer. Jeff has taught me a lot about pore network modelling and software development in general. I would also like to thank Dr. Amin Sadeghi for all his work in helping me learn how to use `OpenPNM` and answering any of my technical pore network modelling or software development questions. It has been a real blessing to have both Jeff and Amin as technical advisors throughout the duration of my project. I am looking forward to continue working with them in the future.

Above all, I must acknowledge and owe it all to my Lord and Saviour Jesus Christ whom I serve. I am reminded daily of His sacrifice on that wooden cross for my sins. Christ, though he was God, did not cling to his divine privilege (Philippians 2:6). Instead, Jesus took the form of a slave. He became mortal flesh walking and living in His own creation. He proved that he was God when he died and was raised back to life (Romans 1:4). Death could not keep Jesus down because in the beginning he Himself breathed out life. Ultimately, we are all faced with death and Christ beat death. No other religion, theory, ideology, or philosophy has a Saviour that beat death Himself. That is why I owe it all to Christ. Thank you Lord Jesus!

Lastly, I would like to thank my family for their support as I have worked on my master’s thesis. My master’s defense was well attended thanks to the support of my family which I owe to my God. I would especially like to thank my Dad for his advice as an experienced engineer. He listens to my problems and asks questions to ensure that I have thought through and can explain my research.

Dedication

This thesis is dedicated to my boys (gvuboys.ca).

Table of Contents

List of Figures

List of Tables

List of Abbreviations

AEM	Anion Exchange Membrane
ASAR	Average Salt Adsorption Rate
CC	Constant Current
CDI	Capacitive Deionization
CEM	Cation Exchange Membrane
CV	Constant Voltage
ED	Electrodialysis
EDL	Electrical Double Layer
FCDI	Flow Capacitive Deionization
FEM	Finite Element Method
GCS	Gouy-Chapman-Stern
HCAM	Hierarchical Carbon Aerogel Monoliths
MCDI	Membrane Capacitive Deionization
mD	modified Donnan
MED	Multi-Effect Distillation
mSAC	Maximum Salt Adsorption Capacity
MSF	Multi-Stage Flash
NME	Numerical Differential Equation
PDE	Partial Differential Equation
PNM	Pore Network Modelling
RC	Resistive-Capacitive
RVD	Reversed Voltage Desorption
SAC	Salt Adsorption Capacity
SAR	Salt Adsorption Rate
SEM	Scanning Electron Microscope
TDS	Total Dissolved Solids
ZVD	Zero Voltage Desorption

Chapter 1

Introduction

At least one third of the world's population suffers from water scarcity [1]. Growing population and finite fresh water supply further aggravates concerns of meeting global water demand. Water desalination is the leading solution to fight water scarcity with installed global capacity of 97.2 million m^3 per day. As well, it has received increasing interest in the last couple of decades with desalination capacity growing at a rate of 7% per annum since 2010 [2].

There are several different desalination technologies available. Desalination technologies can be categorized based on their driving force for separation: pressure-, thermal-, and electro-driven devices [3]. Electro-driven devices use an applied voltage field to remove ions from water. Capacitive deionization is an example of such device [4], [5]. It consists of two charged porous electrodes placed on either side of a spacer channel. The electrodes may be either capacitive or pseudocapacitive electrodes. A voltage field is applied between either type of electrode but the storage mechanism for ion removal is different. For capacitive electrodes, ions are stored in electrical double layers (EDLs) within the microstructure of the electrode while pseudocapacitive electrodes primarily store ions via intercalation. Capacitive electrodes that store ions in EDLs are the focus of this work. Electrodialysis (ED) is an example of another water treatment technology that stores ions in electrical double layers [6].

To model the storage of ions into EDLs, transport equations are coupled with an appropriate EDL model that accounts for the uptake amount and rate of ions within the micropores. Continuum models have been developed and presented in the literature which couple transport algorithms to EDL theory for modelling CDI systems. However, to the best of our knowledge, there does not exist a pore network model (PNM) which couples EDL models with transport equations. There is only one pore scale study on capacitive deionization that uses the Lattice Boltzmann method [7]. Pore network modelling, a subcategory of pore scale modelling, uses a resistor in series like network to model transport in porous media. Therefore, it considers the effect of pore microstructure directly without need of experiment for determination of effective transport properties. Pore network modelling has already been applied to electrochemical devices such as fuel cells and therefore, it shows promise for applications such as electro-driven water desalination technologies or any other process which stores ions in electrical double layers. Traditionally, researchers develop different electrode materials using essentially a trial-and-error approach. This time-consuming approach can be aided by the use of pore network modelling. It is proposed that pore network modelling be used for determining the optimal pore microstructure first and then developing the electrode that has that optimal pore microstructure second. Research has shown that pore microstructure has significant effect on CDI cell performance [8]. However, we caution the reader to think that optimizing pore microstructure means it will make significant progress on thermodynamic efficiency because of a recent study by Patel et al. [3].

However, before PNM can be used for an optimization study on capacitive deionization, EDL models must be incorporated into a pore network modeling framework, which is the main outcome of the present work. For simplicity, a two-dimensional cubic network of 294 pores was generated and used for validations and verification of the new framework. The network is subdivided into cathode, spacer, and anode regions. Simulations were run on this network coupling a transient fickian diffusion algorithm with ionic transport algorithm and different EDL models. The EDL models applied in this paper were: Helmholtz, Gouy-Chapman-Stern, and modified Donnan. The open-source pore network modelling package `OpenPNM` was used. Results were compared with `COMSOL` to check that EDL models were implemented in the pore network model correctly. Charging dynamics were demonstrated on the same 294 pore network for a stop flow CDI cell. In a stop flow CDI cell, there is no flow during charging. Other simplifying assumptions were electroneutrality in macropores, equal anion and cation diffusivities, and monovalent salt solution. Lastly, the modified Donnan model was used to match a stop flow CDI cell discharge curve from literature [9]. The pore network was calibrated to match pore size distribution, permeability, and porosity of actual electrode and electrode-spacer assembly. This thesis begins with a background that includes a review of electrical double layer models followed by a model development section where continuum and numerical method equations are derived.

Chapter 2

Background

2.1 Water Scarcity

Satellite images of the earth show that over 70% of the earth is covered with water. However, it is well known that most of the water on earth's surface is saline or brackish sea water. In fact, less than 3% of the water on earth's surface is potable. On a global scale, there is enough fresh water available to meet water demand on an annual basis [1]. However, spatial and temporal variations in the demand and supply of water is the fundamental reason for water scarcity [1], [10]. For one, there is a geographic mismatch between freshwater sources and human settlements. Secondly, there may be insufficient supply caused by failed infrastructure or institutions to treat and distribute water properly. The United Nations describes water scarcity as being due to a physical shortage (i.e. geographical shortage) or barriers to access due to failed institutions or improper infrastructure [11]. Other stresses such as growing population, climate change, and economic growth further aggravate concerns of water scarcity [12].

There have been several attempts made to quantify the number of people effected by water scarcity in the world [13]–[22]. These reports range from predicting 1.7 to 4 billion people as affected by severe water scarcity. Precisely determining the effect of water scarcity on the human population is a difficult task. For example, different authors have different thresholds for determining what constitutes as water scarcity. Kummu et al. used less than 1000 m^3 of water per capita per year as a benchmark for severe water scarcity while Hoekstra et al. used water consumption to water availability ratio greater than 2 [19], [21]. In essence, water scarcity is a battle between supply and demand of water.

Another important consideration is how the selection of temporal and spatial resolution effects the calculated effect of water scarcity. Mekonnen et al. suggested that other works underestimate the number of people effected by water scarcity by mistakenly averaging out the wet parts of the year with the dry parts. By using a monthly temporal resolution, the authors were able to suggest that almost four billion people in the world suffer from water scarcity at least 1 month a year [1]. All other works investigated range from 1.7 to 3.2 billion people effected by water scarcity which fits the general notion that one third of people experience water scarcity.

Figure 2.1 is a map that shows the annual average of monthly potable or "blue water" scarcity taken [1]. Dark red represents water scarcity greater than 5 while dark green indicates water scarcity from 0 to 0.2. The term blue water is used in these studies to indicate fresh surface or ground water sources. Blue water scarcity is calculated here as the ratio of blue water footprint (or consumption) to blue water availability. This map helps identify where there is water stressed regions in the world.

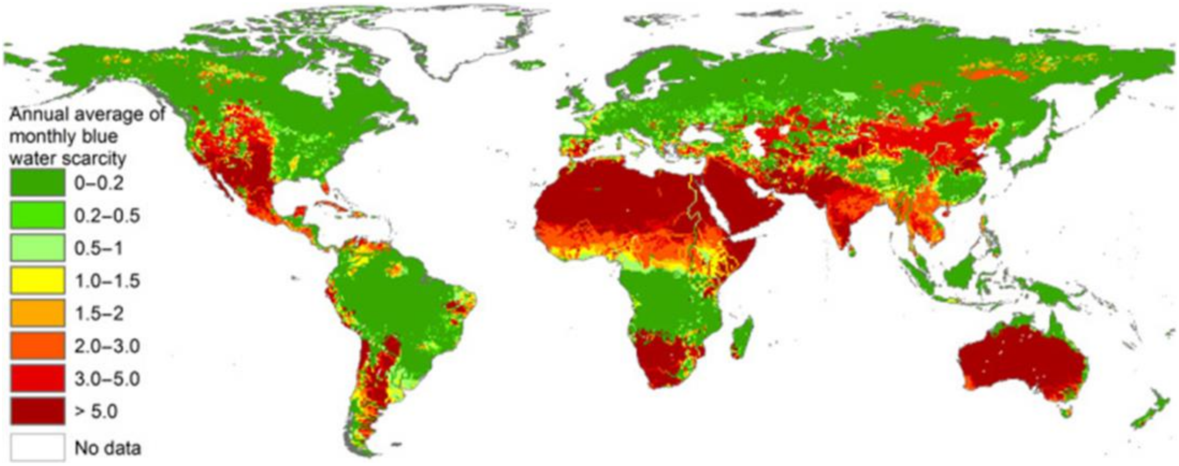


Figure 2.1 Colour map of world showing spatial distribution of annual average of monthly blue water scarcity. Blue water scarcity is determined from the ratio of blue water footprint (or consumption) to blue water availability. Dark red indicates blue water scarcity greater than 5 while dark green indicates blue water scarcity between 0 and 0.2. Reprinted from [1]. © Mekonnen et al., some rights reserved; exclusive licensee AAAS. Distributed under a CC BY-NC 4.0 License (http://creativecommons.org/licenses/by-nc/4.0/)

2.2 Water Desalination

Water desalination is one of the leading solutions to overcoming water scarcity. As of February 2020, there was 97.2 million m^3 of water desalination capacity installed globally. The number of installed plants at that time was 16,876 worldwide. Global desalination capacity has been growing at a rate of 7% per annum since 2010 [2]. Figure 2.2 shows the increasing cumulative capacity of water desalination projects globally since 1970. Here, cumulative desalination capacity considers only installed plants and not contracted projects. Ihsanullah et al. forecasts cumulative desalination capacity will reach 200 million m^3 by the year 2030 [23].

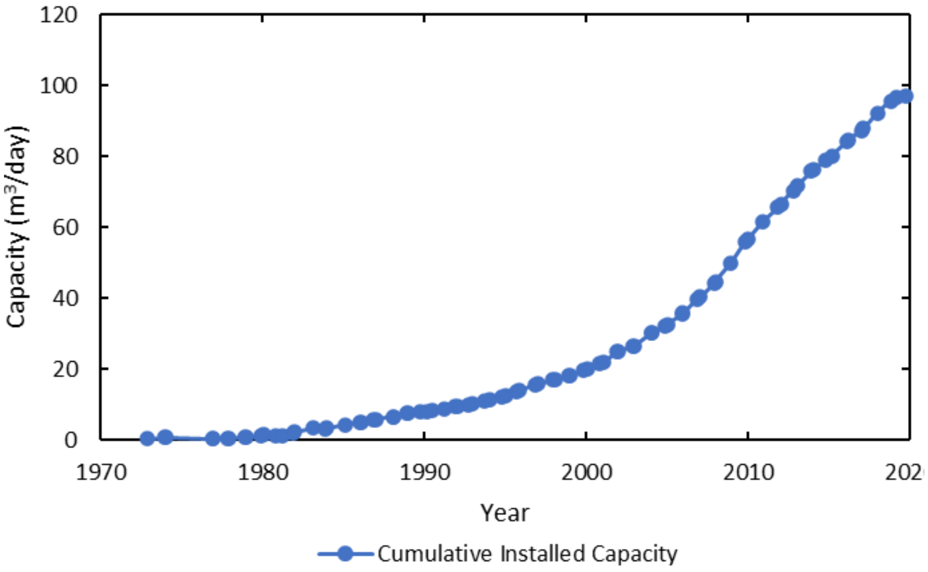


Figure 2.2 The cumulative installed capacity of water desalination projects globally. This figure was adapted from reference [2].

The type of water treated varies among desalination plants. Waters were categorized into six types according to DesalData (2018): seawater: 20,000-50,000 ppm TDS, brackish water: 3,000-20,000 ppm TDS, River water: 500-3,000 ppm TDS, Pure Water: < 500 ppm TDS, Brine: > 50,000 ppm TDS, and wastewater [24]. Pure water is typically used for industrial applications where a very pure water is required [25]. Figure 2.3 shows the breakdown of water desalination plants by the type of water treated. Seawater is treated most often taking 61% of produced water. It is followed by brackish water (21%) and river water (8%). These feedwater types make up 90% of the produced water. The remaining 10% is made up of wastewater (6%), pure water (4%), and brine (<1%). That is the percentage of produced water with certain feedwater. On the left-hand side is the breakdown of water but by number of plants. There are fewer sweater desalination plants then brackish water plants, but seawater plants have a capacity that is three times more than brackish water.

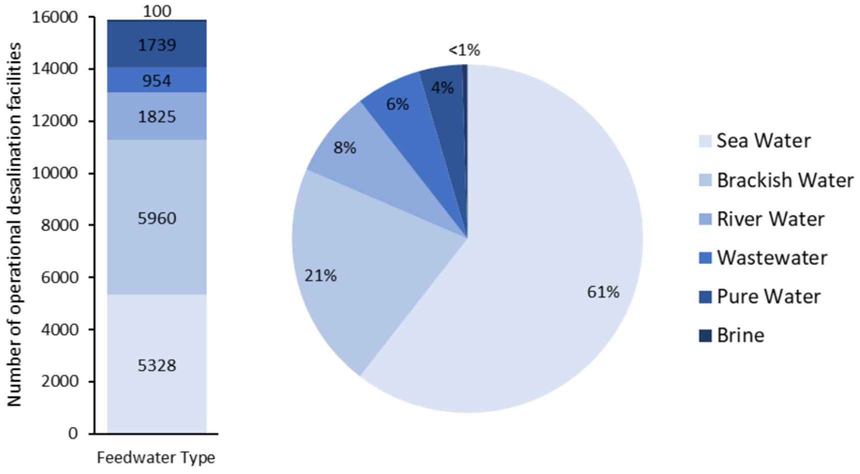


Figure 2.3 Break down of water desalination plants based on feed water type. The bar chart breaks down the number of operational facilities by type of feed water while the pie chart shows percent of water production by type of feed water. This figure was adapted from reference [25].

The most prevalent commercially available desalination technologies can be subdivided into two categories: membrane and thermal desalination. Membrane technologies, mainly reverse osmosis (RO) apply a pressure difference across a semi-permeable membrane to separate salt from water [26]. Meanwhile, thermal driven technologies that include Multi-effect distillation (MED) and multi-stage flash (MSF) work by boiling and condensing water [27], [28]. More novel desalination technologies include: nanofiltration (pressure-driven), electrodialysis, and capacitive deionization. Patel et al. more broadly defines desalination technologies by way of pressure-driven, thermal-driven, and electro-driven in order to include more novel desalination technologies such as electrodialysis and capacitive deionization which are both electro-driven technologies [3].

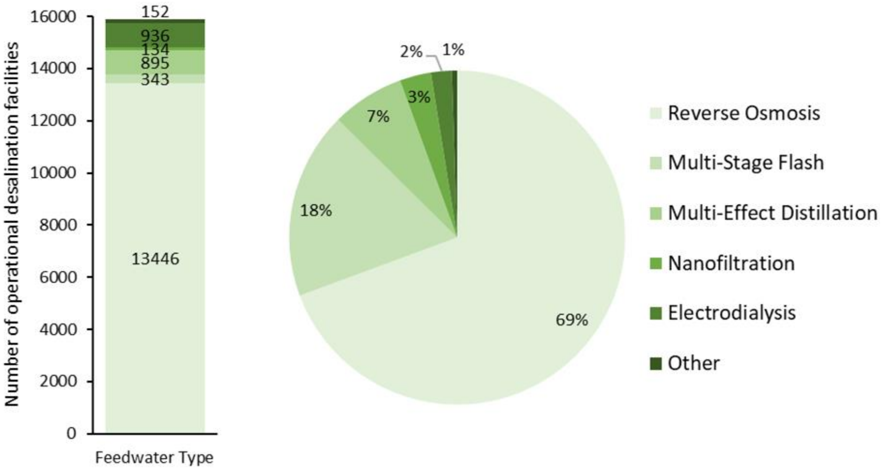


Figure 2.4 Break down of water desalination plants by technology. The bar chart compares the number of operational desalination facilities across technology type while the pie chart breaks down the percent of water production by type. This figure was adapted from reference [25].

Figure 2.4 shows the breakdown of water desalination plants based on technology [25]. The bar on the left counts the number of operating facilities for each technology type while the pie chart breaks down produced water. Reverse Osmosis clearly dominates the water desalination technologies used with 69% of production and 13,446 operating facilities. Reverse Osmosis is followed by the thermal-driven category: MSF (18%) and MED (7%) for water produced. A total of 3% of produced water uses nanofiltration and 2% of water produced uses electrodialysis. There are roughly 1000 electrodialysis water desalination plants in the world. While capacitive deionization is a more novel desalination technology, a few working plants exist such as a municipal wastewater reuse desalination plant with capacity of 60,000 m^3 per day uses CDI desalination modules produced by EST, China. Meanwhile, there is a coal mine wastewater remediation plant with capacity of 5,000 m^3 per day also using CDI modules produced by EST [4].

2.3 Capacitive Deionization

In the previous section, different desalination technologies were discussed and among those were electro-driven devices that included capacitive deionization. Capacitive Deionization works by storing ions in the electrical double layers (EDLs) of micropores in porous carbon electrodes. Therefore, the modelling of EDL theory in this work focuses on its application to studying CDI. A CDI cell consists primarily of a pair of oppositely charged electrodes on either side of a spacer. The spacer prevents electrical contact between the adjacent electrodes while allowing space for water to flow. The spacer is sometimes referred to as the separator. Capacitive Deionization is a dynamic adsorption/desorption process that consists of charging and discharging steps under normal operating conditions. During charging, ions are adsorbed into the electrical double layer structure creating a treated freshwater stream. After the EDLs are saturated with ions, the CDI cell is discharged allowing ions to flow back into solution creating a concentrated brine stream.

CDI is mostly under research and development as there are few attempts at commercialization. Blair and Murphy were the first to publish a paper on a CDI-type of device in 1960 [4], [29]. Over the next 40 years, however, there was very little research interest in CDI and in fact the term CDI was not used until Farmer et al. used it in 1995 [4], [30], [31]. One of the early developments, however, was in the 1970s when Johnson and Newman published a paper on desalting which identified the actual mechanism for ion removal which is now well known today as electrical double layer theory [5], [32]. The slow early development of CDI was due to the limited performance of electrode materials and CDI theory [33]. Research interest re-emerged at the dawn of the 21st century. Figure 2.5 shows the number of publications on Capacitive Deionization since the year 2000. The number of publications has increased exponentially and reached over 400 in the year 2020. As of the end of June 2021, there has been 199 publications on the topic of Capacitive Deionization.

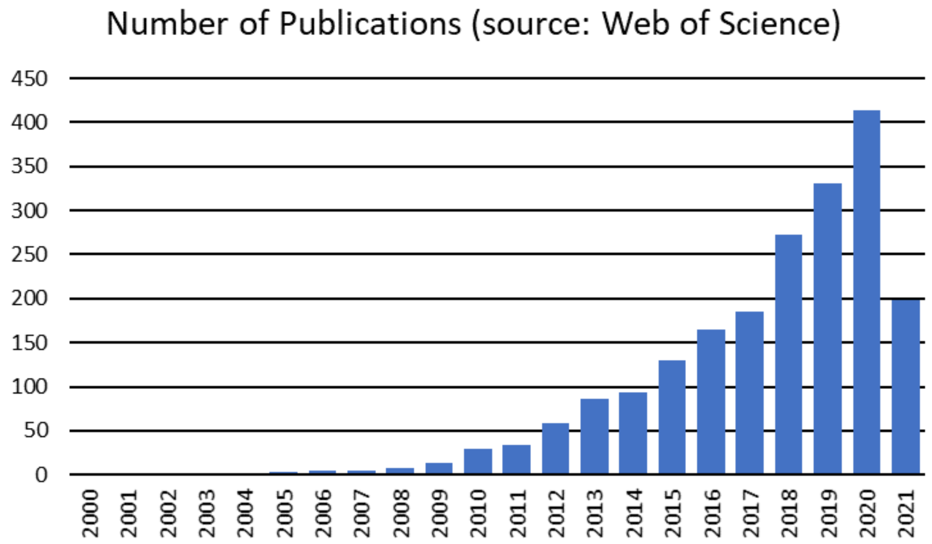


Figure 2.5 Number of publications on the topic of Capacitive Deionization using Web of Science as the source. There has been an exponential increase in the number of publications reaching over 400 in the year 2020. At the time this figure was made it was June 2021 and so the year 2021 is incomplete.

In this section, capacitive deionization is reviewed starting with a description of the architecture of a CDI cell. This includes a discussion of the two most common architecture types (flow-by and flow through) along with their potential drawbacks/limitations of each type while briefly mentioning other less common types of architecture. Second, different operating modes during charging and discharging will be explored along with their drawbacks and limitations. This includes a description of the stop flow CDI cell operation which will be used more in this work. Following this, electrode materials will be reviewed that includes the difference between intercalation and capacitive electrodes. Lastly, a review will be carried out on the performance metrics used to judge CDI performance. What these performance metrics are, how they are calculated, and what values are considered acceptable will be discussed.

2.3.1 Architecture

Here, the architecture of a CDI cell is reviewed. There are two common types of cell architectures: flow-by and flow-through. Figure 2.6 shows a schematic of a) flow-through and b) flow-by cell designs. In the flow-through design, water flows parallel to the applied electric field while in the flow-by case water flows perpendicular to the applied electric field. The flow-through case requires electrodes with a bi-modal pores size distribution consisting of large macropores for water flow and small micropores for high salt sorption. This means higher pressure is needed to drive flow in the flow through case although potentially proper material selection/design could significantly reduce pressure requirements. The flow-by cell introduces much longer diffusion lengths for the ions to reach into the electrode to find a site at which it can absorb, so incurs ohmic resistances. Cell architectures are made up of a pair of electrodes, anode and cathode. The anode and cathode are defined for the charging step which is opposite in the study of electrodes in batteries. The anode is defined as where the anions go during charging while the cathode is where the cations go during charging. Separating anode and cathode is the spacer or otherwise referred to as the separator. In flow-by cell designs, the spacer is typically larger than in flow-through cell designs in order to allow more water to flow between electrodes. Flow through cell designs can have a very thin spacer as the spacer needs to only act as an insulating layer to prevent an electrical short between electrodes. In one flow through CDI cell study, the spacer was only 1% of the thickness of the entire electrode-spacer-electrode assembly. The benefit of this cell design was reported to be more compact and potentially faster desalination due to shorter diffusion timescales in the spacer [9], [34]. The flow-by cell design was most popular among early studies. It was the first CDI cell design used in Blair and Murphy's work [29]. Oren and Sofer used it in their 1978 and 1983 works. The flow through CDI design was first used by Johnson and Newman in their 1971 paper [32].

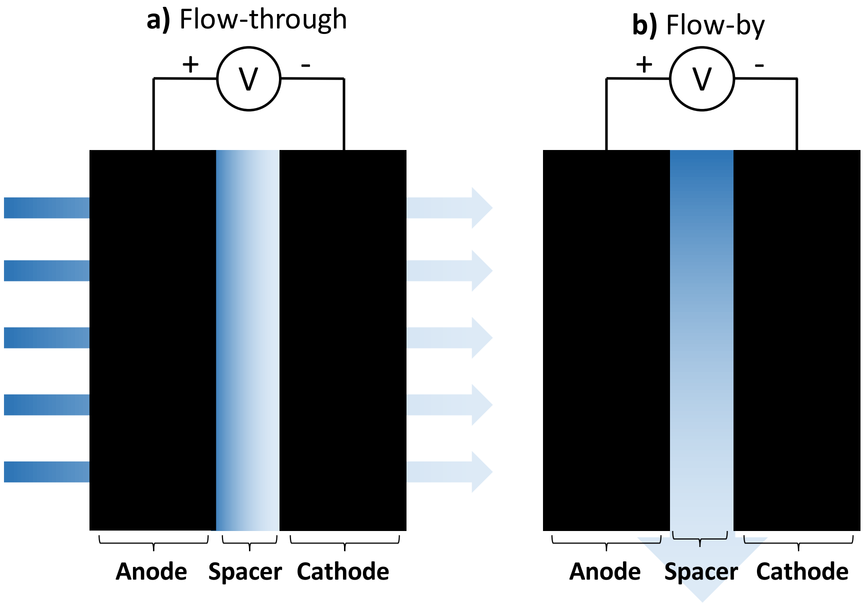


Figure 2.6 Schematic showing two CDI cell architecture types: a) flow-through and b) flow-by CDI architecture. The CDI cells are drawn in charging configuration with ions being adsorbed into the EDL structure. Feed water is darker than then effluent water to distinguish salt water from the treated fresh water.

Besides flow-by and flow-through architecture designs, there have been various other architectures studied. These include membrane capacitive deionization or MCDI where anion and cation exchange membranes (AEM and CEM) are fitted in front of anode and cathode respectively. This helps prevent co-ion expulsion which improves charge efficiency and can even increase storage in electrode macropores [4], [35]. This design was taken after the electrodialysis cell design where feedwater is bounded by both anion and cation exchange membranes [36]. Lee et al. was the first to demonstrate a MCDI cell [37].

Other CDI architectures include: inverted (i-CDI), hybrid, and flow CDI (FCDI) cell designs. The inverted CDI cell design chemically treats a surface charge onto one electrode. This causes inverted behaviour where charging of the CDI cell results in desorption of ions and discharging results in Electrosorption of ions. This architecture has been shown to have long operating times holding performance for 600 hours [4], [38]. Hybrid CDI consists of one faradaic electrode and one non-faradaic (i.e. capacitive) electrode. Lee et al. shows that a hybrid CDI architecture can drastically increase salt adsorption capacity to 31.2 mg g^{-1} where a normal CDI system has a capacity of 13.5 mg g^{-1} [39]. Lastly, flow CDI systems use a slurry-based flow electrode instead of a static electrode and was first utilized in 2013 [40]. Flow electrodes have been developed already for electrochemical energy storage systems. The benefits of FCDI is the continuous operation resulting from continuously pumping new porous electrode particles. There is no need for a desorption step and this effectively increases the capacitance of the cell [4].

2.3.2 Operating Modes

Capacitive deionization is a dynamic adsorption/desorption process consisting of charging and discharging steps. As part of CDI system design, there are different operating conditions that can be used for both charging and discharging steps. During charging, there is constant current and constant voltage operations. One of the benefits of constant current is that you get a constant effluent concentration as first demonstrated on a MCDI cell [41]. Constant voltage experiments on CDI are the most traditional and have been used for a long time however, this does not mean by any means that CV charging is better than CC. Over the years, different comparisons have been made between CC and CV charging to find which one is better [42]. Several reports have concluded that CC is more energy efficient than CV charging [40], [43]–[45]. One report concluded that CV has greater salt adsorption then CC charging [43]. However, it is difficult to say that one charging mode is better than the other because it is application dependent. Wang and Lin defined better as consuming less energy at the same rate of ion removal [42]. But yet, those researchers could not conclusively say one operating mode was better than the other. Therefore, it is best for engineers to consider what is best for their design based on system requirements and review the necessary literature to guide them. During discharging, either constant current or constant voltage modes can also be used. It is more of a question by researchers, however, whether or not reversed-voltage desorption (RVD) or zero-voltage desorption (ZVD) is used. Reversed-voltage desorption is when the polarity is reversed while zero-voltage desorption is when the voltage is removed. Porada et al. discusses these two operating modes for desorption in more detail for CDI and MCDI cells [5].

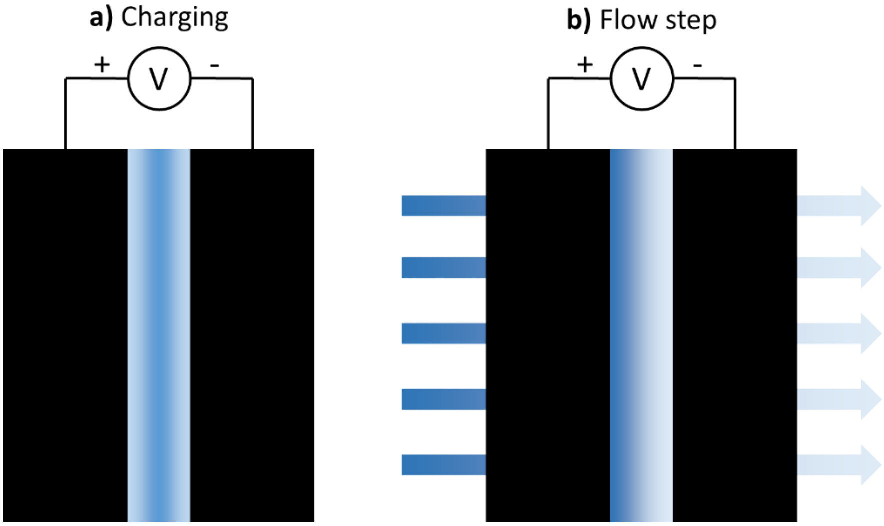


Figure 2.7 Schematic of a stop flow CDI cell showing a) charging and b) flow steps. During charging a constant voltage is applied. Once the cell is charged, the flow step begins pushing treated freshwater out of the cell for collection. During the flow step the voltage is maintained to prevent ions from leaving micropores. Not shown is when the voltage is removed and desorption of ions occurs.

Lastly, stop flow CDI is a special case of CDI cell operation. The modelling work carried out in this paper will be done for a stop flow CDI cell design. This operating mode is called stop flow because during charging the flow is stopped but afterwards, once charging is finished, flow proceeds. Figure 2.7 shows a schematic of a stop flow CDI cell in a) charging and b) flow steps. During the flow step, the voltage is maintained as to avoid allowing ions stored in micropores to enter macropores and leave the cell. After the effluent treated water has been collected from a stop flow CDI cell, the cell can be shorted, and ions desorb releasing a salty brine solution. Examples of a stop flow CDI cell can be found in literature [9], [46].

2.3.3 Performance metrics

In studying capacitive deionization, a class of performance metrics have been developed to better quantify and compare the performance of CDI systems. In this section, a brief review is provided on the common performance metrics used to study capacitive deionization. The metrics that will be discussed are salt adsorption capacity (SAC), salt adsorption rate (SAR), and charge efficiency. Suss et al. defines some best practices for measuring and recording these parameters [4].

The salt adsorption capacity (SAC) is a measurement that has been reported a lot as a judge for CDI cell performance. Soffer and Folman were the first to publish the idea of a salt adsorption capacity [47]. The salt adsorption capacity is the measure of the amount of salt adsorbed during a charging step. If the charging step is long enough that the system reaches saturation, then this is often recorded as the maximum salt adsorption capacity (mSAC). Calculating the salt adsorption in batch mode operation is simply the change in concentration multiplied by the total solution volume. Meanwhile, continuous mode requires the integral with respect to time be taken of the concentration difference multiplied by the flow rate. After determining the amount of salt that was adsorbed, it is typically normalized by the electrode mass or volume and therefore reported as either mg g^{-1} or mg mL^{-1}. Figure 2.8 shows a plot of measured salt adsorption capacities from various papers using a variety of different electrodes/architectures. Capacitive electrodes had a salt adsorption capacity of around 10 mg g^{-1} but using battery electrodes the salt adsorption capacity recently reached over 30 mg g^{-1}. This gives an idea of the historic range of salt adsorption capacities.

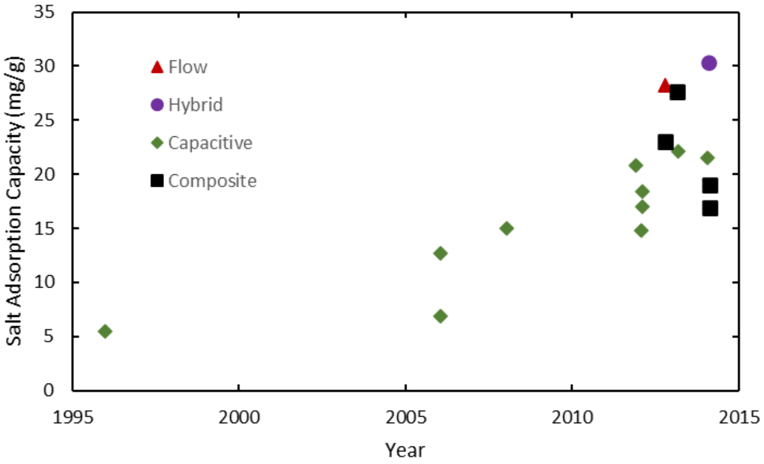


Figure 2.8 Plot showing the increasing salt adsorption capacities from 1995 to 2015 based on type of CDI architecture. The introduction of battery electrodes in a hybrid CDI cell saw SAC reach over 30 mg g^{-1}. This figure was taken from reference [4].

The average salt adsorption rate (ASAR) is another metric for communicating CDI cell performance. Salt adsorption capacity tells us how much salt the CDI system can hold while the salt adsorption rate gives information about the rate at which salt is taken up. Zhao et al. explains how to optimize the salt adsorption rate in a MCDI cell [48]. The salt adsorption rate is calculated by taking the amount of salt adsorbed and dividing by both the electrode mass and time to normalize the data. The time can be the charging time or total cycle time. Suss et al. proposed a convention that the time be the entire cycle time [4]. Unlike the salt capacity which can be considered an electrode property, the salt adsorption rate is a property of the entire CDI cell system as it is dependent on the electrode, material thicknesses/spacing, charging time, and architecture. In fact, the electrode microporosity can play a significant effect on the average salt adsorption rate. Porada et al. calculated the effect of electrode macroporosity on the time it takes to reach 50% SAC and it was found that intermediate values of electrode microporosity were fastest [4], [49].

The final metric of interest is the charge efficiency, Λ. The charge efficiency is calculated by taking the amount of salt adsorbed and dividing by the amount of charge transferred for one cycle. The ratio of salt adsorbed to charge transferred was first mentioned in the work by Johnson and Newman in their early attempts to consider charge efficiency [32]. The term charge efficiency was not used until Avraham et al. in 2009 [50], [51]. Theoretically, the charge efficiency must be less than unity but may approach unity. The reason why we cannot achieve, or it is very difficult to achieve unit charge efficiency is because of the phenomena co-ion expulsion. In order to neutralize a charge on an electrode surface, counter-ions can either enter the micropore or co-ions can be expelled from the micropore. It is the latter case that refers to co-ion expulsion and this phenomenon is what limits the charge efficiency. As mentioned previously AEM and CEM fixed to anode and cathode respectively as in MCDI can help reduce co-ion expulsion improving the charge efficiency. The charge efficiency is a key parameter in determining the energy consumption. Figure 2.9 shows how the energy consumption decreases with charge efficiency including differences in MCDI and CDI designs [34]. Generally, higher charge efficiency means smaller electrical energy consumption.

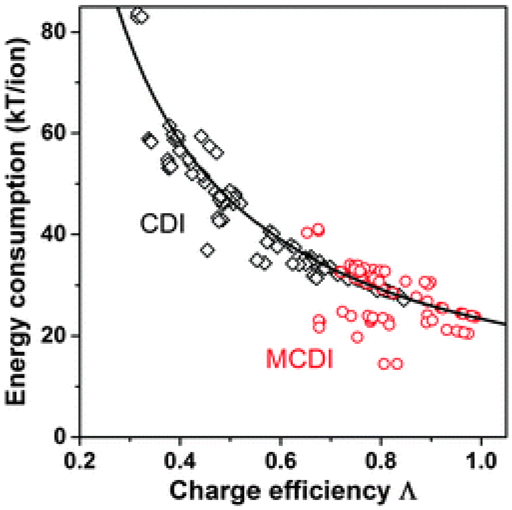


Figure 2.9 Effect of charge efficiency on the energy consumption comparing CDI and MCDI. Improving charge efficiency decreases the energy consumption. The cation and anion exchange membranes in MCDI reduce co-ion expulsion effectively improving charge efficiency. Hence, MCDI has lower energy consumption than traditional CDI architectures. Reproduced from Ref. [4] with permission from the Royal Society of Chemistry.

As a final note, these performance metrics have been used on many studies for assessing CDI. However, theses performance metrics are mostly limited to evaluating CDI and not other water desalination technologies. Patel et al. made this observation and suggested that the thermodynamic energy efficiency should be calculated when comparing CDI to other water desalination devices [3]. A method for calculating the thermodynamic energy efficiency has been developed [52]. Calculating the thermodynamic minimum energy consumption of a desalination process depends on the feed salinity, product water salinity, and water recovery ratio. The thermodynamic energy efficiency has been used to compare CDI to RO [53].

2.3.4 Electrode Materials

Most of the research effort in the field of CDI has focused on developing novel electrode materials with improved energy efficiency, salt capacity, and removal rates. The electrode materials used in capacitive deionization are highly porous materials ever since the early work of Blair and Murphy [29]. Electrode microstructure effects CDI performance [8], [49], [54], [55]. Porada et al. suggests the direct prediction of CDI performance based on pore size distribution and knowledge of geometrical properties of electrode [49]. Figure 2.10 shows SEM images of a commercial carbon-based electrode used in CDI type of application. Notice the hierarchical pore structure of the carbon electrode. Image d) zooms into the big carbon particle revealing it's microporosity.

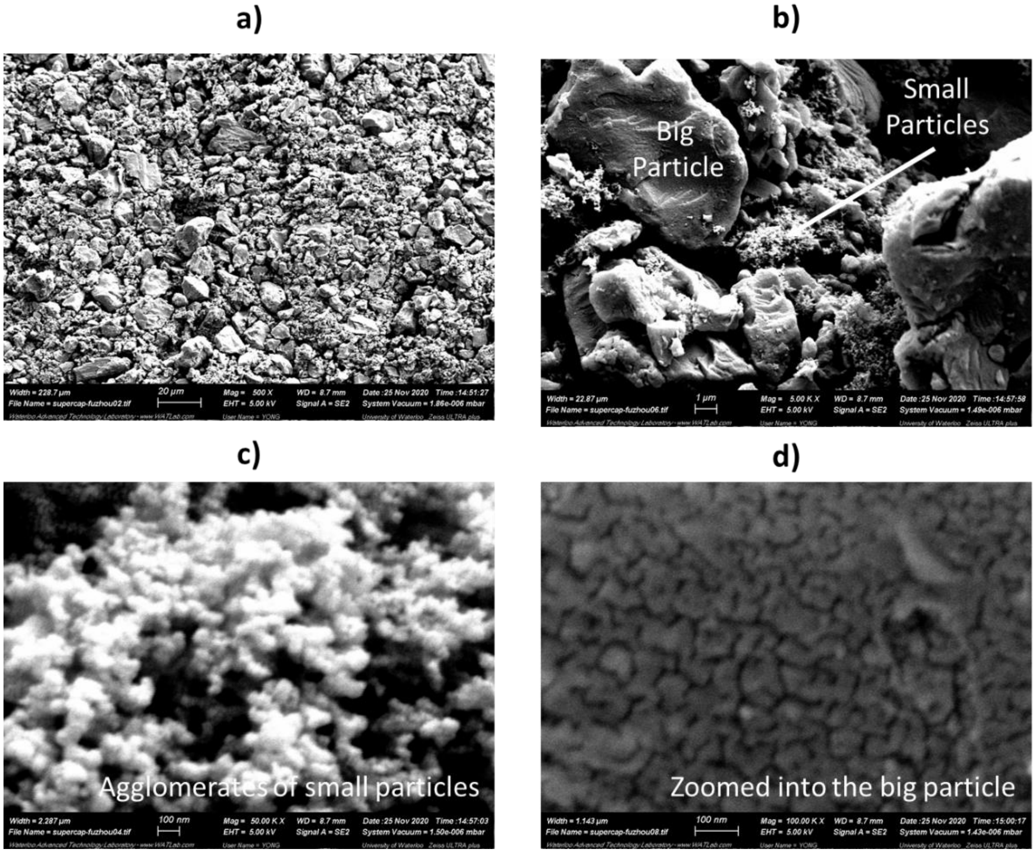


Figure 2.10 SEM images of typical carbon-based electrode used in commercial applications. The images show different sizes or widths a) 228.7 um b) 22.87 um c) 2.287 and d) 1.143. The images help to show the hierarchical pore structure of carbon-based electrodes. Image d) shows a zoomed in image of a big particle containing micropores.

The field of CDI has seen two broad types of electrodes used: capacitive and pseudocapacitive electrodes. Capacitive electrodes store ions in the electrical double layers of micropores. Examples of these electrodes are active carbon, carbon aerogel, and mesoporous carbon as well as more advance carbon materials that rely on carbon-based nanomaterials. Liu et al. writes a review of carbon-based composite materials used in capacitive deionization which includes a table of carbon-based electrode materials and their salt adsorption capacity. Activated carbon, carbon aerogels, and carbon nanotubes typically have a salt adsorption capacity of less than 10 mg g^{-1} [56]. In this work, capacitive electrodes are focused on because EDL storage is the primary mechanism for ion removal. Pseudocapacitive electrodes rely on ion intercalation as the primary mechanism for ion-removal but may also involve some EDL storage of ions. These electrodes are also known as battery electrodes. The developments of these types of electrodes has led to a sharp increase in salt adsorption capacity. For example, Cao et al. achieved a salt adsorption capacity of 137.2 mg g^{-1} from their work using a hybrid CDI cell of composite $Na_3V_2(PO_4)_3/C$ as the one electrode and activated carbon as the second achieved [57]. The review paper by Jiang et al. on battery electrodes has an excellent summary of battery electrodes tried and their salt adsorption capacity recorded [58]. Many of these recorded have a salt capacity greater than 100 mg g^{-1}. Patel et al. on the other hand, calculated and compared the thermodynamic efficiency for activated carbon, advanced carbon, and pseudocapacitive electrodes and found that the activated carbon materials generally performed better than the others with regards to thermodynamics [3].

2.4 Electrical Double Layer Theories

An electrical double layer model describes a separation of charge across an interface where excess charge in one phase is equally compensated by local charge in an adjacent phase [5]. For the case of capacitive deionization, the interface is the electrolyte/electrode surface, and the excess charge is the electronic charge in the electrode equally compensated by ionic charge in the adjacent electrolyte. Several different EDL models have been proposed to describe this phenomenon. Three different EDL models have been used in studying capacitive deionization for modelling the storage of ions in micropores. These EDL models are: Helmholtz, Gouy-Chapman-Stern (GCS), and modified Donnan (mD). It is in this section that a brief review is presented on these electrical double layer models. A description of the assumptions and model equations is provided for each of these EDL theories along with a summary of their historical use pertinent to studying capacitive deionization.

2.4.1 Helmholtz

The term ‘double layer’ was first used to describe two layers of opposite charge in 1883 by H. Helmholtz [59], [60]. The earliest description of the electrical double layer dates back even earlier, to 1853 when Helmholtz published a paper proposing a capacitor-like EDL structure where all surface charge (either electronic charge or chemically bound surface charge) is equally compensated by counterions in a condensed layer near the charged surface [5], [59]. Important for the discussion is the difference between counter- and co-ions. Counter-ions are ions with the opposite charge as the electrode surface and will be attracted by the surface charge while co-ions are ions with the same charge as the electrode surface and will be repelled by the surface charge. For the case of the Helmholtz EDL model, the condensed layer is made up only of counterions. That is, each surface (or electronic) charge is directly and equally compensated by counterion charge. This leads to the ideal case of unit charge efficiency where each electronic charge transferred removes one salt molecule (for the case of electro-driven water desalination devices). It is important to note that the Helmholtz model is not sufficient to describe the uptake rate of ions into electrical double layers for electro-driven desalination technologies such as capacitive deionization [5]. Johnson et al. in 1971 is the earliest attempt of applying Helmholtz

EDL theory to study capacitive deionization [32]. In that paper, Johnson et al. attempts a correction for unit charge efficiency. There exists a more recent study that used helmholtz EDL theory for modelling capacitive deionization but they too warned against applicability of Helmholtz for studying EDL based on it's unit charge efficiency and linear EDL capacitance [9].

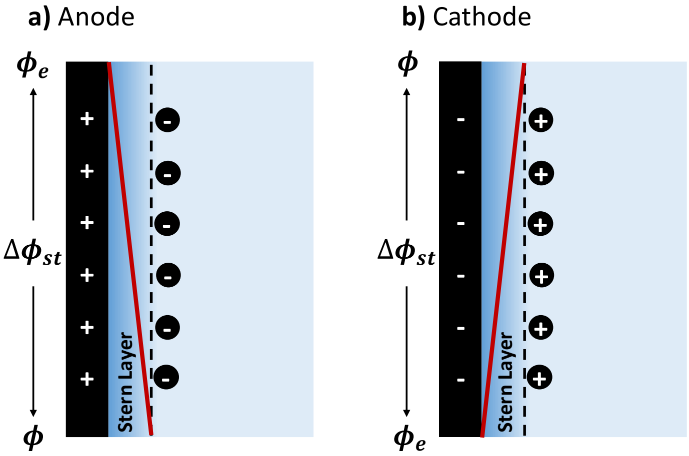


Figure 2.11 A schematic of Helmholtz EDL model for a) anode and b) cathode at steady state. Counter-ions form a fixed plan of charge adjacent to solid electrode at some finite distance. The electronic surface charge is equally compensated by the ionic charge in the electrolyte. The potential drop occurs entirely across the Stern layer.

Figure 2.11 shows the electrode/electrolyte interface assuming Helmholtz EDL structure for a) anode and b) cathode where the condensed layer of ionic charge consists only of counterions. Note that the CDI definition of anode and cathode is based on the charging step. The anode is where anions go while the cathode is where cations go during charging [5]. This definition is the opposite to cathode and anode in battery and supercapacitor applications [5]. Notice that a dielectric layer exists between surface and counterion charge (called the Stern layer in Figure 2.11) reflecting the idea that ionic charge cannot come infinitely close to electronic charge. Their separation could be due to ion size, atomic roughness, or because electronic charge may not be located directly at the edge of the carbon surface [61]. In the Helmholtz EDL structure, the potential drop occurs entirely across the dielectric or Stern layer as shown by the red line in Figure 2.11. This results in a "capacitor-like" structure and can be treated mathematically as such. To understand how Helmholtz is applied mathematically it is important to recall simple capacitor theory from electrical engineering. Figure 2.12 is a schematic of a simple electrical circuit of a capacitor in series with a voltage source. The electrode/electrolyte interface can be thought of as a capacitor in series with a voltage source.

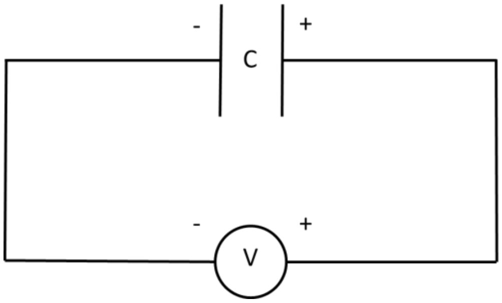


Figure 2.12 Simple electrical circuit of capacitor in series with a voltage source. Upon charging, electrons collect on the surface of the capacitor. Once the capacitor has reached it's capacity to hold charge, the voltage remains steady and electronic charge stops moving.

During charging, current flows from the voltage source and collects onto the surface of the capacitor. How much charge collects on the surface of the capacitor depends on the voltage (ϕ) and the capacitance (C). The capacitance is an intrinsic physical property of a capacitor giving the amount of charge stored per voltage. It is calculated by the amount of charge stored at steady state (zero current flow) per unit surface area. The units of capacitance are F m^{-2}. The amount of current flowing (i) can be calculated from equation 2.1. This is the basis of the source term for the uptake rate of ions in a Helmholtz EDL model as will be described in section 3.1. To determine the amount of ions accumulating on the condensed Helmholtz layer, equation 2.1 is used along with Faraday's number to convert charge to moles.

$$i = C\,\frac{d\phi}{dt} \qquad 2.1$$

2.4.2 Gouy-Chapman-Stern

As described previously, the Helmholtz EDL model is insufficient at describing the EDL structure in porous CDI electrodes because a condensed layer of counter-ions leads to unit charge efficiency. Experiments have shown that CDI cells do not achieve unit charge efficiency due to co-ion expulsion [51]. The Gouy-Chapman-Stern EDL model does not result in unit charge efficiency because it does not assume a condensed layer of counter charge but rather that charges remain diffusively distributed near the electrode surface in a diffuse layer. After, Helmholtz published his electrical double layer model, a new EDL model was developed from the combinations of works by Gouy, Chapman, and Stern. Gouy-Chapman assumed ions remain diffusively distributed in electrolyte [62], [63]. Stern expanded on the Gouy-Chapman model by adding a dielectric layer between electrode and diffuse layer that does not contain charge. Combination of a diffuse layer with a Stern layer dates back to Stern in 1924 [64]. This dielectric layer was coined the term *"Stern Layer"* (or Helmholtz layer). The representation of the EDL structure overcomes the limitation of the Helmholtz model which resulted in unit charge efficiency. Figure 2.13 is a representation of the GCS electrical double layer model. It consists of a Stern and diffuse layer. Across the Stern layer the potential drops linearly but then, in the diffuse layer, the counterion concentration progressively decays with increasing distance from electrode. There have been several example of GCS used to model capacitive deionization [65], [66].

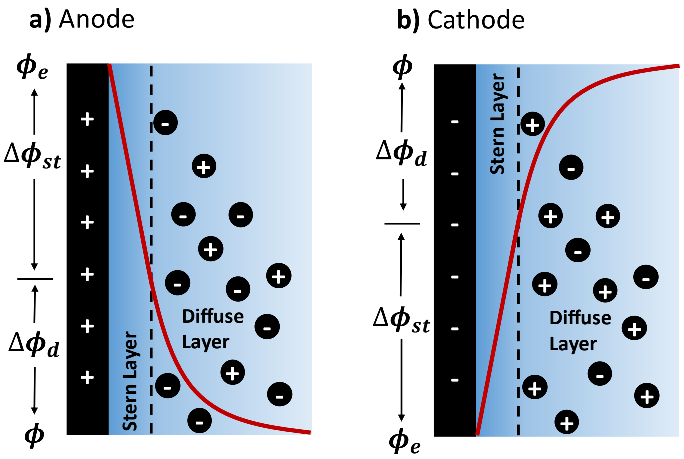


Figure 2.13 A schematic of Gouy-Chapman-Stern EDL model for a) anode and b) cathode. In this EDL model, charge remains diffusively distributed in a diffuse layer adjacent to electrode surface. A think dielectric region called the Stern layer separates charged surface and diffuse layer. The potential drops linearly across the Stern layer while it progressively decays (i.e. anode) in the diffuse layer.

The Debye length, λ_D, is characteristic length of the Diffuse layer. The diffuse layer ends 2 to 3 times the Debye length. To calculate the Debye length, equation 2.2 is used which is an expression for the inverse Debye length, $\lambda_D = {}^{1}/_{\kappa}$. It uses the macropore (or bulk) concentration C_{mA}, Avogadro's number N_{av}, and the Bjerrum length, λ_B.

$$\kappa^2 = 8\pi\lambda_B c_{mA} N_{av} \qquad 2.2$$

The Bjerrum length, λ_B, is given by equation 2.3 where R is the ideal gas constant, T the temperature, F is faraday's number, and $\varepsilon_r\varepsilon_o$ is the dielectric permittivity of water ($78\cdot 8.854\times 10^{-12}$ C V^{-1} m^{-1}). At room temperature, the Bjerrum length is 0.72 nm [5].

$$\lambda_B = F^2/(4\pi\varepsilon_r\varepsilon_o RTN_{av}) \qquad 2.3$$

Now, for example, the Debye length of an equimolar NaCl salt solution of strength 10mM at 20 °C is approximately $\lambda_D \sim 3.1\ nm$. Comparing this length to the size of a micropore (< 2nm), the diffuse layer is likely to overlap with that of an adjacent diffuse layer. Since uptake of ions occurs primarily in the micropores of porous electrodes, the GCS model is still not sufficient to model electrosorption of ions in the micropores of porous carbons. One study argues that the pore size distribution has a large effect on measured maximum salt adsorption capacity and that micropores below 2 nm contribute most significantly to mSAC [4], [49].

Now to compute the amount of charge stored in the EDLs assumed to have the GCS structure, the Poisson-Boltzmann relationship is used to describe the distribution of ions in the diffuse layer.

$$c_{mi,j} = c_{mA} \exp\left(-z_j \cdot \phi(x)\right) \qquad 2.4$$

The micropore concentration of species j is denoted $c_{mi,j}$, z_j is the valence of species j, c_{mA} is the concentration of the macropore, and $\phi(x)$ is the potential profile as a function of distance from the electrode, x. The diffuse-charge dynamics are explained in great detail in Bazant et al. [67]. The potential distribution as a function of distance from the electrode is calculated using equation 2.5 as provided in reference [65].

$$\phi(x) = 4 \cdot arctanh\left(exp(kx) \cdot \tanh\left(\frac{F}{4RT}\Delta\phi_d\right)\right) \qquad 2.5$$

Equation 2.4 is integrated across the diffuse layer using equation 2.5 yielding expression for the surface charge density, σ and the excess salt concentration, w. The surface charge density describes the amount of charge stored in the diffuse layer. Meanwhile, the excess salt concentration describes the amount of salt (not ions) captured by the micropores assuming symmetric electrodes are used. Equation 2.6 and 2.7 describe these.

$$\sigma = 4\lambda_d c_{mA} \sinh\left(\frac{F}{2RT}\Delta\phi_d\right) \qquad 2.6$$

$$w = 8\lambda_d c_{mA} \sinh^2\left(\frac{F}{4RT}\Delta\phi_d\right) \qquad 2.7$$

To determine the Donnan potential, a potential balance across the EDL is written such that equation 2.8 is formed. Here, ϕ_e is the potential of the solid electrode which is assumed to be constant and ϕ is the potential of the bulk. It is assumed that electrons transfer instantaneously to the electrode compared to the movement of ions in the electrolyte solution.

$$\Delta\phi_d + \Delta\phi_{st} = \phi_e - \phi \qquad 2.8$$

The Stern potential is calculated form the charge density as demonstrated by equation 2.9. Faraday's number is used to convert the number of moles of charge into coulombs of charge. The capacitance, C, of the Stern layer is used to calculate the voltage drop across the Stern layer.

$$\Delta\phi_{st} = \frac{\sigma F}{C} \qquad 2.9$$

2.4.3 modified Donnan

The modified Donnan electrical double layer model was developed as an attempt to model overlapping electrical double layers. The classical Donnan model is modified by adding a Stern layer and including an extra attraction term, μ_{att} which accounts for non-electrostatic forces of attraction. It is well known that non-electrostatic forces exist in porous carbons which cause ions to be stored in uncharged micropores [68]. Equation 2.10 describes the micropore concentration using the Donnan potential and extra attraction term. In this work, it was assumed that μ_{att} equals zero and therefore, non-electrostatic forces of attraction were neglected.

$$c_{mi,j} = c_{mA}\exp\left(-z_j \cdot \Delta\phi_d + \mu_{att}\right) \qquad 2.10$$

Figure 2.14 shows the structure of a modified Donnan electrical double layer model. Notice that the potential drop occurs only across the Stern layer and that the overlapping diffuse layer results in constant properties independent of distance from electrode surface. The Stern potential drop, $\Delta\phi_{st}$ occurs across the Stern layer and the Donnan potential drop, $\Delta\phi_d$, is the difference in potential between the micropore and macropore. The mD model is actually mathematically simpler than the GCS model. In fact, the mD model overcame numerical difficulties in using classical GCS EDL models in case of strong salt concentration gradients [4]. Several studies have used the mD EDL model for CDI research but only in continuum models [69], [70].

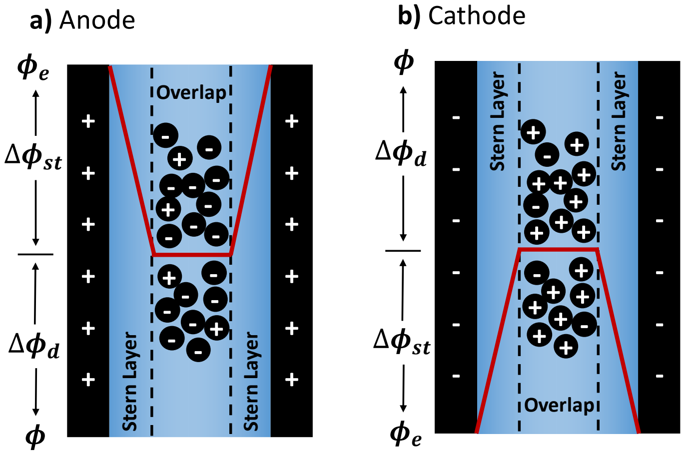


Figure 2.14 Schematic of modified Donnan EDL model for a) anode and b) cathode. The EDLs of two neighbouring electrode surfaces overlap each other resulting in constant properties within the overlapping region. The potential drops linearly across the Stern layer while there is a step change in potential, known as Donnan potential, between micro and macro pore space.

Equation 2.10 was used to obtain expressions for cation and anion micropore concentration in the modified Donnan EDL structure. The difference of cation and anion micropore concentration led to the determination of the charge concentration, $c_{charge,mi}$. This quantity represents the moles of charge (positive or negative) per micropore volume. Meanwhile, the summation of cation and anion micropore concentration led to an expression for the micropore concentration, c_{mi}. Here, the micropore concentration is the total number of moles of ions (anion and cation) per micropore volume. This quantity is intrinsically important for determining the rate of uptake of ions into the modified Donnan EDL structure as discussed more in section 3.3. The micropore charge concentration and micropore concentration are expressed below as equations 2.11 and 2.12 respectively.

$$c_{charge,mi} = -2c_{mA} \sinh\left(\frac{F}{RT}\Delta\phi_d\right) \tag{2.11}$$

$$c_{mi} = 2c_{mA} \cosh\left(\frac{F}{RT}\Delta\phi_d\right) \tag{2.12}$$

To calculate the Donnan potential, the same potential balance across the EDL is used as in equation 2.8 for the Gouy-Chapman-Stern model. The Stern potential drop is calculated using the charge micropore charge concentration and equation 2.13 where a is the specific surface area of the electrode in units of $\mathrm{m}^2\ \mathrm{m}^{-3}$.

$$\Delta\phi_{st} = -\frac{c_{charge,mi}F}{Ca} \tag{2.13}$$

Chapter 3

Model Development

In this chapter, numerical method equations (NMEs) are developed that incorporate electrical double layer theory to model the storage of ions in a pore network. The electrical double layer theories for which NMEs are developed are Helmholtz, Gouy-Chapman-Stern, and modified Donnan. Continuum model equations or partial differential equations (PDEs) are described first as the basis of developing appropriate numerical method equations (NMEs). The PDEs are used by a finite element methods (FEMs) solver (i.e. `COMSOL`) for comparison with results from the developed NMEs used on a synthesized 2D network of pores. This chapter starts with a description of the 2D pore network used for demonstrating the application of our derived NMEs and for comparison to a FEM solver.

3.1 Description of Artificial 2D Pore Network

An artificial representation of a porous material was generated to demonstrate and validate the use of the developed NMEs on a pore network. This artificial network was designed to be two dimensional so that an FEM solution of the PDEs directly on a meshed geometry equivalent to the artificial network could provide a reference solution to validate the results produced from NMEs. Figure 3.1 is the 2D pore network generated for demonstrating the use of developed NMEs and is the network that most of the simulations in this work were performed on. The FEM solver used this geometry and the sub pore scale mesh used by the FEM solver is shown in Figure 3.1. The artificial network was subdivided into three regions consisting of anode, cathode, and spacer pores in order to represent the three characteristic regions of a CDI cell. The 2D network is a square lattice of spacing $100\ \mu m$ with dimensions of 21 by 14 pores for a total of 294 pores. The anode and cathode regions consist of 112 pores each while the spacer consists of 70 pores. One third of throats in anode and cathode regions were randomly selected and removed to attempt a more realistic pore connectivity. The diameter of spacer pores was set to half of the average diameter and no spacer throats were removed. This is the geometry that was used to demonstrate the use of EDL theory in a pore network.

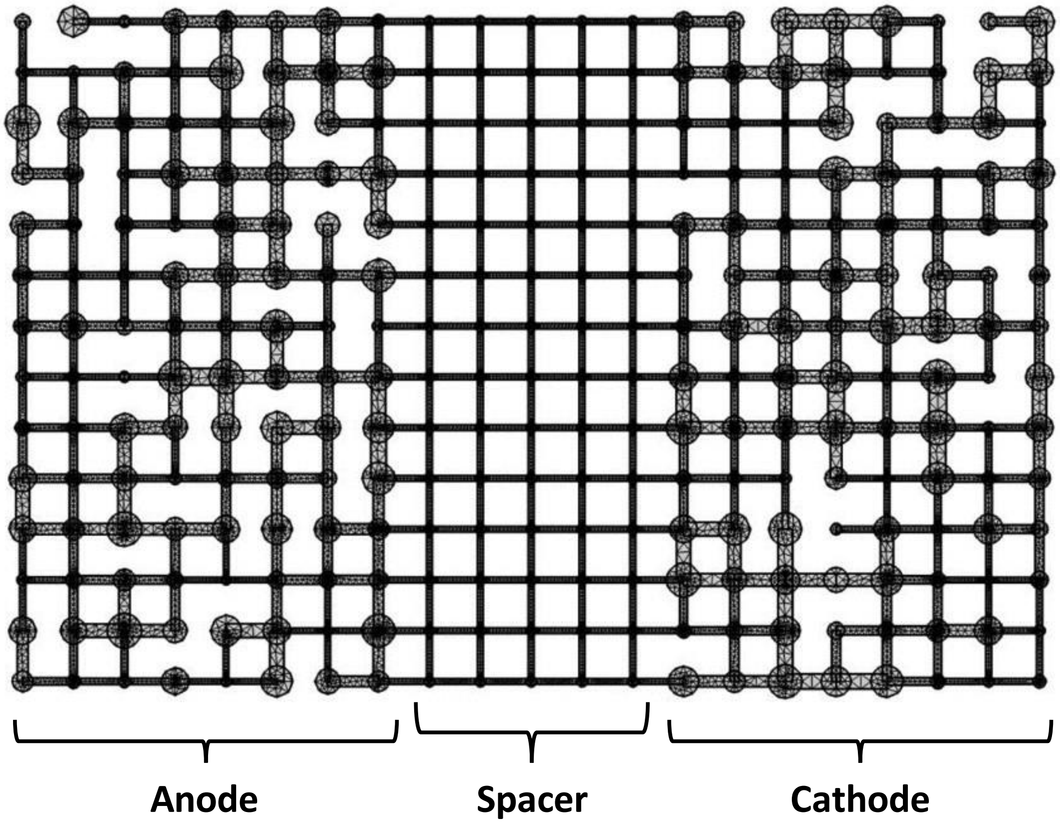


Figure 3.1 A 2D porous network made up of 294 pores on a square lattice subdivided into anode, cathode, and spacer subregions. One third of the throats in anode and cathode regions were randomly selected and removed. Pores in the spacer were assigned a uniform diameter equal to half of the average pore diameter. The sub pore scale mesh used by the FEM solver is shown.

3.2 Continuum Model Description

In this section, continuum model equations or PDEs are developed for modelling charging and flow steps of a stop flow CDI cell. These PDEs are derived by performing mass and charge balances over a differential control pore volume. Full derivation can be found in Appendix A. The Nernst-Planck relationship as described by equation 3.1 was used to calculate the molar flux of species j, $\boldsymbol{N}_j$, entering and leaving the control volume. It should be mentioned here that some simplifying assumptions (as described later on) reduce our Nernst-Planck expression to a simple Fickian diffusion problem. The reason why the Nernst-Planck expression is written and used in our development of model equations is to demonstrate the adaptability of our model approach to more complex systems.

$$\boldsymbol{N}_j = -D_j \boldsymbol{\nabla} c_{mA,j} + \boldsymbol{u} c_{mA,j} - \frac{D_j z_j F}{RT} c_{mA,j} \boldsymbol{\nabla} \phi \qquad 3.1$$

The first term of the Nernst-Planck expression represents the diffusive flux due to a concentration gradient and depends on the diffusion coefficient of species j, D_j (m^2/s). The second term is the advection term where $\boldsymbol{u}$ is the velocity vector. In this work, all vector quantities (i.e. N, $\boldsymbol{\nabla}$, and $\boldsymbol{u}$) are written in bold font. The third term calculates the migrative flux contributions due to the presence of a potential field. The potential is denoted by ϕ whereas z_j, F, R, and T are the valence of species j, faraday's constant, ideal gas constant, and temperature. The Nernst-Planck expression for molar flux depends on the macropore concentration of species j denoted by $c_{mA,j}$. The macropore concentration is used because only transport in the macropore is solved because the pore network in Figure 3.1 does not resolve micropores. Resolving micropores would significantly increase computational cost by 100 or even 1000 times.

3.2.1 Generic Mass Transport Derivation

The Nernst-Planck expression for molar flux is substituted into a transient mass conservation equation for species j (see Appendix A). For a stop flow CDI cell there is no flow during charging, therefore no flow ($\boldsymbol{u} = 0$) is assumed resulting in the advection term disappearing. In the case of a monovalent salt solution such as $NaCl$, two mass balances are written, one for each of the dissolved salt species. Electroneutrality is assumed in the macropores ($\mathrm{c_{mA} = c_{mA,\,Na} = c_{mA,\,Cl}}$) and so therefore the individual balances for Na and Cl can be summed together to arrive at a single balance for salt. In the case that we have equal anion and cation diffusivities ($\mathrm{D = D_{Na} = D_{Cl}}$) and monovalent salt solution ($z_{Cl} = -1, z_{Na} = +1$), the migration term disappears. The uptake rate of ions into the electrical double layers of micropores is modelled using a reaction term, R, with units of $mol/(m^3 \cdot s)$. This effectively reduces our problem to a transient Fickian diffusion with reaction problem as shown by equation 3.2.

$$\frac{\partial c_{mA}}{\partial t} = \mathbf{\nabla} \cdot (D\mathbf{\nabla} c_{mA}) - R \qquad 3.2$$

The reaction term of equation 3.2 was carefully considered. In the spacer of a stop flow CDI cell, there is no reaction or electrosorption of ions, therefore, the reaction term is zero, $R = 0$, and equation 3.2 reduces to equation 3.3. The reaction tern is applied to anode and cathode electrodes. The exact form of the reaction term depends on the EDL theory applied. The reaction term is described for each of the EDL theories in section 3.2.3.

$$\frac{\partial c_{mA}}{\partial t} = \mathbf{\nabla} \cdot (D\mathbf{\nabla} c_{mA}) \qquad 3.3$$

3.2.2 Generic Ionic Transport Derivation

In the following sections, it will be shown that the reaction term of equation 3.2 depends on the macropore potential. Therefore, our mass transport algorithm needs to be coupled with an ionic transport algorithm that determines the macropore potential. Hence, a charge balance is performed across a differential control pore volume. Charge conservation is imposed in a similar way as that employed by Agnaou et al. [71]. The major difference here is that, in the electrode, there is accumulation of charge in the electrical double layers. To account for the accumulation of charge, the

Stern layer in the electrical double layer is treated as a capacitor-like structure. Recall from section 2.4.1 how the current of a charging capacitor can be calculated from the rate of change of voltage. It is this relationship that is used to calculate the accumulation of charge in each macropore but with the potential drop across the Stern layer, $\Delta\phi_{st}$, used for the applied voltage. The specific surface area, a, term gives the available surface area per macropore volume. While some charge accumulates, there is current flowing in and out of the control macropore volume. Equation 3.4 is the complete charge balance with the accumulation term on the left-hand side and where $\boldsymbol{i}$ is the current density with units $\mathrm{A/m^2}$. Important to recall is that current is the net rate of positive charge transferred across an interface and so this is a balance on positive charge.

$$Ca\frac{\partial\Delta\phi_{st}}{\partial t} = -\boldsymbol{\nabla}\cdot\boldsymbol{i} \qquad 3.4$$

To calculate the current density, an expression for the ionic current density in the electrolyte was used. The ionic current density is calculated using equation 3.5 where $\boldsymbol{N}_j$ is the flux of species j calculated using the Nernst-Planck relationship in equation 3.1.

$$\boldsymbol{i} = F\sum_{n}\left(z_j\boldsymbol{N}_j\right) \qquad 3.5$$

The expression for current density can be substituted into equation 3.4 along with the Nernst-Planck expression for mass flux. Again, the advection term is removed from the Nernst-Planck expression by assuming no flow ($\boldsymbol{u} = 0$). The migration term is removed by assuming a monovalent salt solution, electroneutrality, and equal anion and cation diffusivities. By way of these assumptions, the derivation for ionic transport arrives at equation 3.6. For a full derivation refer to appendix A.

$$Ca\frac{\partial\Delta\phi_{st}}{\partial t} = -\boldsymbol{\nabla}\cdot(K\boldsymbol{\nabla}\phi) \tag{3.6}$$

Equation 3.6 is used for modelling ionic transport in the anode and cathode electrodes where there is accumulation of charge. However, in the spacer there is no electrosorption of ions. Hence, no accumulation of charge resulting in equations 3.7 where the accumulation term is removed for modelling ionic transport within the spacer pores.

$$0 = \boldsymbol{\nabla}\cdot(K\boldsymbol{\nabla}\phi) \tag{3.7}$$

The ionic conductivity of the solution is denoted by K with units of F m^{-1} s^{-1} and calculated using equation 3.8.

$$K = \frac{2F^2Dc_{mA}}{RT} \tag{3.8}$$

3.2.3 Specifying the Reaction Term, R

Up to this point, generic mass and ionic transport equations have been described. This includes equations 3.3 and 3.7 which are the model equations used for the spacer pores in the pore network where there is no storage of ions into the EDLs of the micropores associated with each macropore. In general terms, the uptake rate of ions was described using a reaction term, R, in the mass transport of equation 3.2. The exact reaction term depends on the EDL theory applied. It is in this section that the reaction term is specified for each of the electrical double layer theories: Helmholtz, Gouy-Chapman-Stern, and modified Donnan. In addition, the Stern potential is given special treatment for each of the EDL theories resulting in redefinitions of equation 3.6 for ionic charge transport.

In the case of Helmholtz, a capacitor-like EDL structure is assumed where the rate of uptake into the EDL can be calculated based on equation 2.1. Faraday's number, F, is used to convert current which is the flow of charge to flow of moles of an ion for incorporating equation 2.1 into the mass balance. The counterion valence, z^{count}, within the cathode is $+1$ while the counterion valence in anode pores is -1. In the anode, the potential will decrease with time while, in the cathode, the potential will increase with time until the system reaches steady state at approximately zero volts. Therefore, using the counterion valence, it was assured that the Helmholtz source term acts as a consumption term in both cathode and anode pores. The Helmholtz source term is divided by two as a result of assuming electroneutrality in the macropores. Therefore, the source term is $R = \frac{Ca}{2z^{count}F}\frac{\partial \phi}{\partial t}$ for the Helmholtz case. For a full derivation of this source term refer to appendix A. As for the treatment of the potential drop across the Stern layer, Helmholtz assumed that the potential drop occurs entirely across the Stern layer and therefore $\Delta\phi_{st} = \phi_1 - \phi$ where ϕ_1 is constant due to constant voltage assumption. Equations 3.9 and 3.10 are the equations for modelling storage of ions using a Helmholtz EDL structure.

$$\frac{\partial c_{mA}}{\partial t} = \nabla \cdot (D\nabla c_{mA}) - \frac{Ca}{2z^{count}F}\frac{\partial \phi}{\partial t} \qquad 3.9$$

$$Ca\frac{\partial \phi}{\partial t} = \nabla \cdot (K\nabla \phi) \qquad 3.10$$

Next the reaction term for the Gouy-Chapman-Stern EDL structure is considered. In this EDL model, the rate of change of the surface charge density, w, with time is used for the reaction term, R. The GCS model uses the surface charge density because the concentration varies within the diffuse layer. Equation 2.7 is used to calculate the surface charge density as a function of the Donnan potential. The reaction term for the GCS model contains the specific surface area, a, as a prefactor. This prefactor converts salt density which is per surface area to concentration on the basis of macropore volume. Equation 3.11 shows the mass transport equation with the GCS reaction term applied.

$$\frac{\partial c_{mA}}{\partial t} = \nabla \cdot (D \nabla c_{mA}) \; - \; a \frac{\partial w}{\partial t} \qquad 3.11$$

$$Ca \frac{\partial \phi}{\partial t} = \nabla \cdot (K \nabla \phi) - Ca \frac{\partial \Delta \phi_d}{\partial t} \qquad 3.12$$

Coupled with the mass conservation equation for the GCS model is equation 3.12, the ionic transport equation. The Stern potential is treated differently from the Helmholtz case because GCS accounts for co-ion expulsion. The additional source term arises from the fact that a potential drop occurs across the diffuse layer and does not drop entirely across the Stern layer as in the Helmholtz EDL model. The additional source term suggests that an increasing Donnan potential causes either positive charge to be consumed or negative charge to be expelled from the micropore and into the macropore. Mathematically, equation 3.12 can be obtained by substituting equation 2.8 for the Stern potential into equation 3.6.

Lastly, the reaction term for the modified Donnan EDL model is described. The reaction term used for the mD model is determined from the rate of change of micropore concentration with time. The micropore concentration is the concentration of total ions in the electrical double layer and is calculated using equation 2.12 which depends on the Donnan potential. Equation 3.13 describes the mass transport equation with a modified Donnan source term. The prefactor, $V^{mi}/_{V}$ is used for converting the source term from units of per micropore volume to per macropore volume. For a continuum model, this prefactor can be determined based on the micro and macro porosities. The $1/2$ factor is necessary to convert total ion concentration to salt concentration assuming electrode symmetry.

$$\frac{\partial c_{mA}}{\partial t} = \boldsymbol{\nabla} \cdot (D \boldsymbol{\nabla} c_{mA}) - \frac{1}{2} \frac{V^{mi}}{V} \frac{\partial c_{mi}}{\partial t} \qquad 3.13$$

Finally, the mD model uses the same ionic transport equation, that is equation 3.12, as the GCS model. Both GCS and mD models account for co-ion expulsion in the same way. Therefore, the modified Donnan model equations couple equation 3.13 for mass transport with equation 3.12 for ionic transport.

3.2.4 Advection and Diffusion Transport

In the previous sections, mass and ionic transport equations are provided for modelling the charging dynamics of a stop flow CDI cell. Special attention was given to the reaction term responsible for modelling the storage of ions into the micropore EDLs. During the flow stage, the EDLs are filled with ions and the voltage is maintained to prevent ions from leaving the micropores as treated water is collected. Therefore, the flow stage can be modelled in such a way that ignores the reaction term that accounts for the electrosorption of ions. This means that the flow stage can be modelled as a simple dispersion or advection-diffusion problem. In this section, the PDEs for an advection-diffusion problem are presented. This is already a well-defined problem, so we did not include a derivation in the appendix. However, like previous derivations it is recommended to start with the Nernst-Planck relationship for molar flux (equation 3.1). The migration term on the right-hand side can be neglected because it is assumed that potential gradients have subsided from the charging step. A transient mass balance is then carried out on a control volume using this expression for advection-diffusion molar flux. This treatment of the Nernst-Planck expression leads to the PDE for advective-diffusive transport shown in equation 3.14.

$$\frac{\partial c_{mA}}{\partial t} = \boldsymbol{\nabla} \cdot (D \boldsymbol{\nabla} c_{mA}) - \boldsymbol{\nabla} \cdot (\boldsymbol{u} c_{mA}) \qquad 3.14$$

In this work, the PDE for advective-dispersion transport is not used directly by a FEM solver because NMEs for the dispersion have been previously validated by Sadeghi et al. [72]. However, the PDE is presented here to be complete and to provide the starting point from which the NME for advection-dispersion in a pore network is derived.

3.3 Pore Network Model Description

In the previous section, PDEs were derived for modelling charging dynamics and flow stage in a stop flow CDI cell. These PDEs are used as the basis for deriving our NMEs used in our pore network model. The derived NMEs used in our pore network model are described in the following section. A pore network model uses a resistor in series like network to calculate well mixed pore properties. The resistances or conductance models are fundamental to calculating transport in a pore network model. Therefore, the conductance models used for modelling charging and flow stages are described first followed by a description of the NMEs used. The NMEs for the charging step are described first for each of the three EDL models and finished with a description of the NMEs for the flow stage.

3.3.1 Conductance Model Formulations

The conductance models used for calculating transport in a stop flow CDI cell are described first. A conductance is calculated for each pore-throat-pore conduit. Figure 3.2 is a drawing of a pore-throat-pore conduit for two spherical pores connected via a cylindrical throat. In this assembly there are three resistances: resistance of $pore\ i$, $throat\ ij$, and $pore\ j$. The conductance is the inverse of the resistance. Here, g_i^{tr} denotes the conductance of $pore\ i$ for some transport mechanism tr. For example, in the case of diffusion $tr = d$. The diameters and lengths of the pores and throat in a pore-throat-pore conduit are defined in Figure 3.2. The conductance for the entire pore-throat-pore conduit is denoted as G_{ij}^{tr} and is calculated using equation 3.17 for resistors in series. This is how the conductance of a conduit is calculated for any generic transport mechanism.

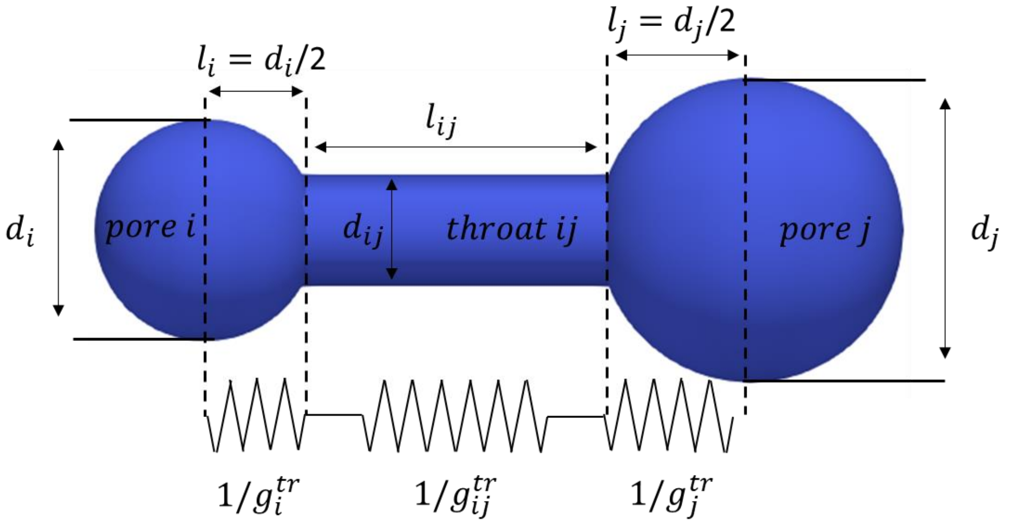


Figure 3.2 A drawing of a pore-throat-pore conduit of which conductance values are based on. Conductance values are calculated assuming a resistor in series like network where there are three resistors in series: pore *i*, throat *ij*, and pore *j* resistances.

$$G_{ij}^{tr} = \left(\frac{1}{g_i^{tr}} + \frac{1}{g_{ij}^{tr}} + \frac{1}{g_j^{tr}}\right)^{-1} \qquad 3.15$$

During the charging step, there are two transport conductance models that need to be calculated: diffusive conductance and ionic conductance. The diffusive conductance is denoted using g_i^d, while the ionic conductance is denoted using, k_i, for a single pore i. The diffusive and ionic conductance are calculated using equations 3.16 and 3.17 respectively. A_i is the cross-sectional area of pore i. In a 2D pore network, the cross-sectional area of pore i, is the diameter, d_i, of the pore. The diffusive conductance is based on fickian diffusion while the ionic conductance was derived based on equation 3.8 which assumes electroneutrality, equal cation and anion diffusivities, and monovalent salt solution.

$$g_i^d = \frac{A_i D}{l_i} \tag{3.16}$$

$$k_i = \frac{2F^2 D c_{mA,i} A_i}{RT l_i} \tag{3.17}$$

As previously mentioned, the conductance for the entire pore-throat-pore conduit is calculated assuming resistors in series. Based on the generic formulation of conduit conductance for any transport mechanism in equation 3.17, diffusive and ionic conduit conductance formulations are written as equations 3.18 and 3.19 respectively.

$$G_{ij}^d = \left(\frac{1}{g_i^d} + \frac{1}{g_{ij}^d} + \frac{1}{g_j^d}\right)^{-1} \tag{3.18}$$

$$K_{ij} = \left(\frac{1}{k_i} + \frac{1}{k_{ij}} + \frac{1}{k_j}\right)^{-1} \tag{3.19}$$

During the flow stage of a stop flow CDI cell, the transport mechanisms change from diffusive and ionic transport to advection-diffusion transport. For advection-diffusion, there are two conductance models: hydraulic conductance and diffusive conductance. The calculations for diffusive conductance were already given for diffusive transport by equations 3.16 and 3.18. The hydraulic conductance, however, is calculated using equation 3.20 and is based on the Hagen-Poiseuille for laminar flow through a tube.

$$g_i^h = \frac{\pi}{128\mu}\left(\frac{d_i^4}{l_i}\right) \tag{3.20}$$

Equation 3.20 is valid only for flow in a tube. The exact formulation of the hydraulic conductance depends on the geometry. For a 2D network, where throats are rectangles, the hydraulic conductance is calculated using equation 3.20.

$$g_i^h = \frac{\pi}{12\mu}\left(\frac{d_i^3}{l_i}\right) \qquad 3.21$$

Lastly, the hydraulic conductance for a conduit is calculated using equation 3.22 assuming a resistor in series like network.

$$G_{ij}^h = \left(\frac{1}{g_i^h} + \frac{1}{g_{ij}^h} + \frac{1}{g_j^h}\right)^{-1} \qquad 3.22$$

3.3.2 Numerical Method Equations for Charging Dynamics

Thus far, conductance formulations have been defined for diffusive, ionic, and hydraulic transport. It is these conductance formulations that are the backbone of the NMEs in a pore network model. In this section, the NMEs are presented for the charging stage of a stop flow CDI cell. The NMEs take the form of a mass balance around pore i. For the sake of brevity, semi-discrete forms of our model equations are presented starting with a discretized space term followed by discretized transient terms. Therefore, accumulation and reaction terms are discretized second as they are both transient terms. A space discretization is performed on equations 3.3 and 3.7 to arrive at the NMEs for spacer pores where there is no electrosorption of ions. Equations 3.23 and 3.24 are the numerical model equations for where there is no reaction term or uptake of ions into EDLs.

$$V_i \frac{\partial c_{mA,i}}{\partial t} = -\sum_{j=1}^{n} G_{ij}^d \left(c_{mA,i} - c_{mA,j}\right) \qquad 3.23$$

$$0 = -\sum_{j=1}^{n} K_{ij}\left(\phi_i - \phi_j\right) \qquad 3.24$$

Next, the reaction term is considered for electrode pores where there is uptake of ions into the EDL structure. The NMEs are written first for the Helmholtz model which gives equations 3.25 and 3.26 for mass and charge algorithms respectively.

$$V_i \frac{\partial c_{mA,i}}{\partial t} = -\sum_{j=1}^{n} G_{ij}^{d}\left(c_{mA,i} - c_{mA,j}\right) - \frac{CaV_i}{2z^{count}F}\frac{\partial \phi_i}{\partial t} \qquad 3.25$$

$$CaV_i \frac{\partial \phi_i}{\partial t} = -\sum_{j=1}^{n} K_{ij}\left(\phi_i - \phi_j\right) \qquad 3.26$$

The NMEs were written for the Gouy-Chapman-Stern EDL model. Recall that co-ion expulsion is considered and so therefore, the additional source term in the ionic transport equation. Equations 3.27 and 3.28 are the NMEs for mass and ionic transport for the GCS case, respectively.

$$V_i \frac{\partial c_{mA,i}}{\partial t} = -\sum_{j=1}^{n} G_{ij}^{d}\left(c_{mA,i} - c_{mA,j}\right) - aV_i \frac{\partial w_i}{\partial t} \qquad 3.27$$

$$CaV_i \frac{\partial \phi_i}{\partial t} = -\sum_{j=1}^{n} K_{ij}\left(\phi_i - \phi_j\right) - CaV_i \frac{\partial \Delta\phi_{d,i}}{\partial t} \qquad 3.28$$

The mass transport NME for the modified Donnan model is equation 3.29. The modified Donnan model uses the same ionic transport equation as the Gouy-Chapman-Stern model and therefore is coupled with equation 3.28.

$$V_i \frac{\partial c_{mA,i}}{\partial t} = -\sum_{j=1}^{n} G_{ij}^{d}\left(c_{mA,i} - c_{mA,j}\right) - \frac{V_i^{mi}}{2}\frac{\partial c_{mi,i}}{\partial t} \qquad 3.29$$

Thus far, the space terms have been discretized, but the source and accumulation terms are both transient. Therefore, it is of special interest to discretize transient terms in order to handle the reaction term. In this work, an implicit scheme is used to discretize transient terms for all EDL models. The time discretization, according to the implicit scheme, of equations 3.25 and 3.26 is given by equations 3.30 and 3.31 respectively.

$$\left[\frac{c_{mA}}{\Delta t} + \frac{Ca}{2z^{count}F}\frac{\phi}{\Delta t} - \nabla \cdot (D\nabla c_{mA})\right]^{t_1} = \left[\frac{c_{mA}}{\Delta t} + \frac{Ca}{2z^{count}F}\frac{\phi}{\Delta t}\right]^{t_0} \qquad 3.30$$

$$\left[Ca\frac{\phi}{\Delta t} - \nabla \cdot (K\nabla\phi)\right]^{t_1} = \left[Ca\frac{\phi}{\Delta t}\right]^{t_0} \qquad 3.31$$

Similarly, the transient terms of the Gouy-Chapman-Stern model equations were discretized according to the implicit scheme. Equations 3.32 and 3.33 are the resulting NMEs discretized in the time domain.

$$\left[\frac{c_{mA}}{\Delta t} + aV_i\frac{w_i}{\Delta t} - \nabla \cdot (D\nabla c_{mA})\right]^{t_1} = \left[\frac{c_{mA}}{\Delta t} + aV_i\frac{w_i}{\Delta t}\right]^{t_0} \qquad 3.32$$

$$\left[Ca\frac{\phi}{\Delta t} + CaV_i\frac{\Delta\phi_{d,i}}{\Delta t} - \nabla \cdot (K\nabla\phi)\right]^{t_1} = \left[Ca\frac{\phi}{\Delta t} + CaV_i\frac{\Delta\phi_{d,i}}{\Delta t}\right]^{t_0} \qquad 3.33$$

Lastly, the mass transport equation for the modified Donnan EDL model is discretized in the time domain. The result is equation 3.34. As before, this equation is coupled with equation 3.33.

$$\left[\frac{c_{mA}}{\Delta t} + \frac{V_i^{mi}}{2}\frac{c_{mi,i}}{\Delta t} - \nabla \cdot (D\nabla c_{mA})\right]^{t_1} = \left[\frac{c_{mA}}{\Delta t} + \frac{V_i^{mi}}{2}\frac{c_{mi,i}}{\Delta t}\right]^{t_0} \qquad 3.34$$

3.3.3 Advection-Diffusion in a Pore Network Model

During the flow stage, there is no ionic transport because it is assumed that sufficient charging has taken place such that EDLs are saturated with ions. Therefore, only advection-diffusion transport occurs during the flow stage. A steady state stokes flow algorithm is solved to obtain the pressure of each pore in the network.

$$\sum_{j=1}^{n} G_{ij}^{h}\left(p_i - p_j\right) = 0 \qquad 3.35$$

In the dispersion modelling paper of reference [72], there were several different approaches to discretizing our dispersion PDE (equation 3.14). These were: upwind, hybrid, power-law, and exponential schemes. In this work, the power-law discretization scheme was used because it had one of the lowest relative errors in comparison to finite element solvers used as the ground truth. Equation 3.36 shows the power-law discretization of the advection-dispersion PDE. The transient term was discretized using the implicit scheme similar to mass and ionic transport equations.

$$\begin{aligned}\sum_{j=1}^{n}\left(G_{ij}^{d} max\left[\left(1-\frac{\left|Pe_{ij}\right|^{5}}{10}\right),0\right]+\max\left(q_{ij},0\right)\right)c_{mA,i}\\ -\sum_{j=1}^{n}\left(G_{ij}^{d} max\left[\left(1-\frac{\left|Pe_{ij}\right|^{5}}{10}\right),0\right]+\max\left(-q_{ij},0\right)\right)c_{mA,j}=V_i\frac{\partial c_{mA,i}}{\partial t}\end{aligned} \qquad 3.36$$

3.4 Initial and Boundary Conditions

The generated network in Figure 3.1 will be used to simulate stop flow operating conditions of a CDI cell as demonstrated in reference [9]. Stop flow operating conditions consist of a charging step and a flow step. In order to simulate both charging and flow steps, different initial and boundary conditions are defined for each step. The initial and boundary conditions are defined first for the charging step followed by the flow step.

3.4.1 Initial and Boundary Conditions of the Charging Step

During the charging step, electrosorption of ions occurs in the anode and cathode pores via a reaction term while diffusive effects transport ions from spacer pores. There are no boundary conditions during the charging step because the charging step of a stop flow CDI cell has no flow, and it is assumed that there is no exchange of fluid between the cell and surrounding environment. Initial conditions, on the other hand, must be applied to determine the concentration of salt solution to undergo desalination and the applied cell voltage of the charging step. Concentration and potential initial conditions for the charging step are shown in Figure 3.3 on the artificial 2D network.

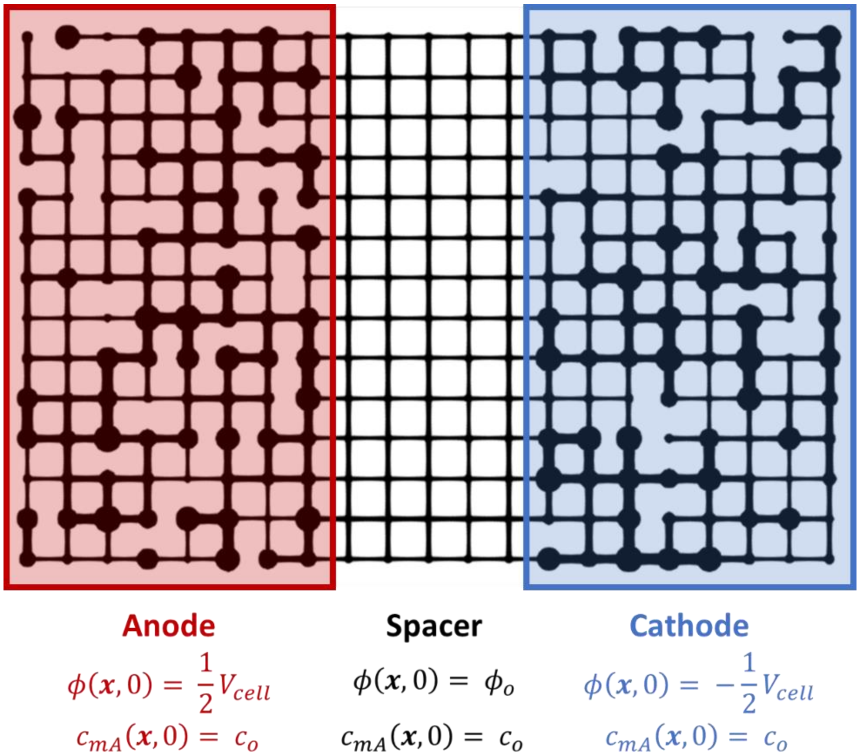


Figure 3.3 The initial conditions of the charging step for anode, spacer, and cathode subregions of artificial network. The initial condition of potential is $\frac{1}{2}V_{cell}$ in the anode, $-\frac{1}{2}V_{cell}$ in the cathode, and ϕ_o in the spacer. The initial potential of spacer pores, ϕ_o, is determined by running an ionic conductance algorithm with initial conditions of anode and cathode as boundary conditions. The initial concentration is c_o and is the same for all pores in anode, cathode, and spacer subregions.

The initial concentration of the charging step is c_o for all pores. In this work, an initial salt concentration of 100 mM was used. This concentration is equivalent to that of 5,844 mg L^{-1} and so representative of brackish water according to the US geological survey which says brackish water is between 1,000 to 10,000 mg L^{-1} [73]. Meanwhile, for initial potential, it is assumed that electric current instantaneously collects on the surface of the electrode compared to ionic current in the adjacent electrolyte. Therefore, the potential of the electrode surface must be half of the cell voltage, V_{cell}, at time zero. Furthermore, since there is not yet any capture of ions in the electrical double layers there is no potential drop across Stern layer or between micro- and macropores. Hence, for the sake of modelling, we can say that the potential in the macropores of the electrodes at time zero is the same as the potential on the adjacent electrode surface as explained in reference [74]. The initial potential of anode pores is therefore $\frac{1}{2}V_{cell}$ while the initial potential of the cathode pores is $-\frac{1}{2}V_{cell}$. The potential drops entirely across the spacer pores and so the initial potential of these spacer pores is determined by running an ionic conductance algorithm (equation 3.7) with the initial potential of anode and cathode as boundary conditions. It is the initial potential drop between anode and cathode that is responsible for catalyzing the reaction or uptake of ions into electrical double layers of micropores.

3.4.2 Initial and Boundary Conditions of the Flow Step

The flow step occurs immediately after the charging step in order to collect the treated or desalinated water. A pressure difference is applied across the CDI cell driving the treated water out of the cell and forcing new 100 mM salt solution into the cell. Therefore, pressure boundary conditions are essential for simulating flow through the cell. At the left face, a pressure of $\mathrm{P_{left}}$ is applied and at the right face, a pressure of $\mathrm{P_{right}}$ is applied. Concentration boundary conditions are applied to the pores on the left face of the network to realize the surrounding fluid concentration ($\mathrm{c_o}$) being forced into the CDI cell. Outflow boundary conditions are added to the pores on the right face of the network to simulate fluid leaving the right side. The initial concentration of the cell is the final concentration, $\mathrm{c_f}$, from the charging step simulation. These initial and boundary conditions are summarized in Figure 3.4.

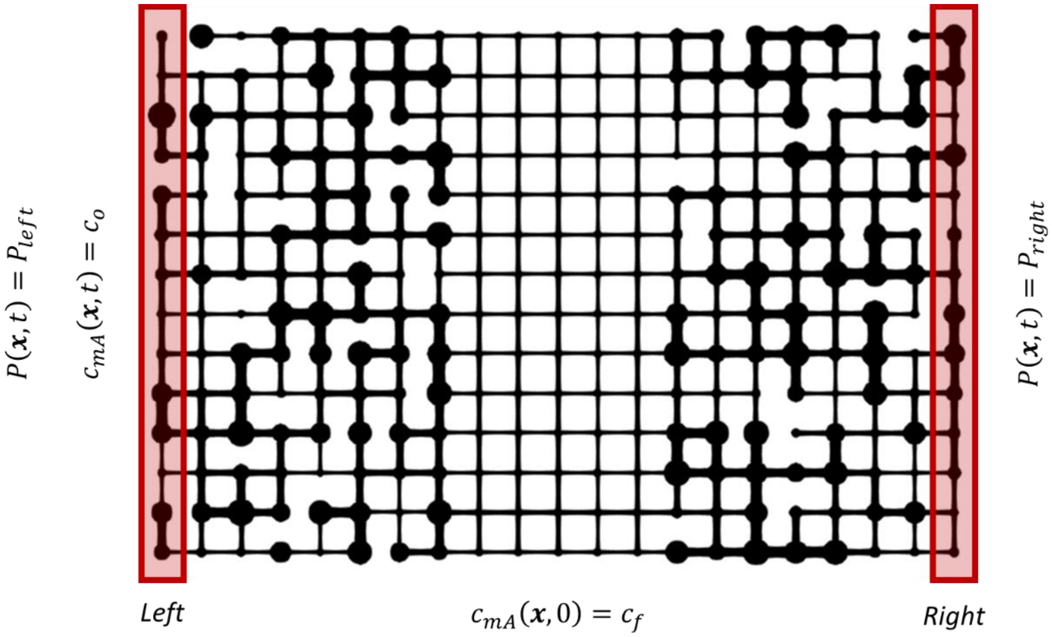


Figure 3.4 Initial conditions of the flow step shown on the artificial 2D pore network. The concentration is initially the concentration at the end of charging. Pressure boundary conditions are applied to left and right pores to induce flow. The pores on the left face are given the boundary concentration of c_o which is the concentration of the surrounding salt solution.

Chapter 4

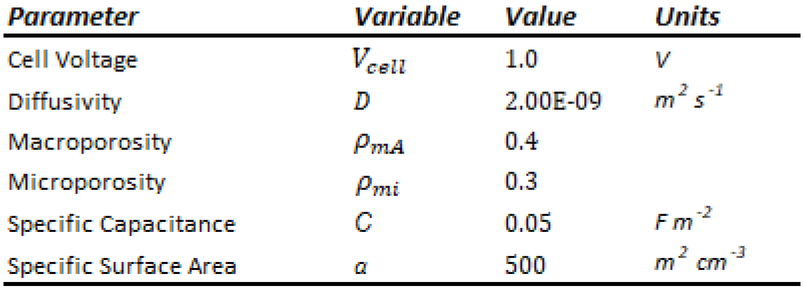
Results and Discussion

In the following, the main results and findings of the research is presented. In this work, there were two main studies, each conducted on a distinct network, to investigate the application of EDL theory within a pore network model. The first study demonstrated and validated the applicability of EDL theory to a pore network model for simulating ion storage in a CDI type of device. This study used the 2D network of pores shown in Figure 3.1. Each of the EDL theories: Helmholtz, Gouy-Chapman-Stern, and modified Donnan were tested and demonstrated on this network of pores. The dynamic charging effects of a stop flow CDI cell were simulated using each of these EDL theories and results compared to finite element method (FEM) solver. Finally, this study demonstrated how to obtain discharge curves of a stop flow CDI cell using an advection-diffusion simulation. The second study attempted to model the stop flow CDI cell found in reference [9]. To model this CDI cell, an artificial network of pores on a cubic lattice was synthesized and calibrated to match the properties of the electrode-separator assembly found in the aforementioned paper. Charging on this artificial network was simulated using the modified Donnan EDL model for a duration similar to the experiment done by Suss et al. [9]. A flow stage was simulated using previously demonstrated advection-diffusion simulation to obtain a discharge curve. The modelled discharge curve was then compared to the discharge curve in literature.

4.1 Demonstration & Validation of EDL Theories Applied to Pore Network

Each EDL theory as described in section 2.4 was applied to the network of pores shown in Figure 3.1. Charging dynamics of a stop flow CDI cell was simulated for Helmholtz, Gouy-Chapman-Stern, and modified Donnan models. The numerical method equations (NMEs) described in section 3.3.2 were used for the pore network model. These equations depend on certain key parameters such as capacitance, specific surface area, and diffusion coefficients. Table 1 lists the important parameters used to solve these equations in a pore network. By default, a cell voltage of 1.0 V was used, typical of CDI operation which is held below 1.2 V to avoid water splitting. It was assumed that sodium and chloride ions have equal diffusivity of 2×10^{-9} m^2 s^{-1} which is roughly between the actual diffusivities of each ion. The modified Donnan model requires knowing the micropore volume. To determine the micropore volume, the ratio of microporosity to macroporosity was used to estimate the micropore volume adjacent to each macropore. Therefore, micro and macro porosities of 0.3 and 0.4 were used respectively. The summation of these gives an overall porosity of 0.7 which is similar to many other porous carbon electrodes with bimodal pore size distribution. A moderate specific capacitance of 0.05 F m^{-2} was used along with a specific surface area of 500 m^2 cm^{-3}. Here, the specific surface area was regarded as the available surface area per macropore volume. The available surface area accounted for the surface area of adjacent micropores not resolved in the pore network. The surface area used is a fraction of the surface area made by BET measurements by Baumann et al. on carbon aerogels [75]. It is difficult to know precisely the total surface area involved in EDL storage of ions but it is likely less than the total available BET surface area. Therefore, the specific surface area may be treated as a fitting parameter.

Table 1 List of important parameters used in demonstration of EDL theory in a pore network. A cell voltage of 1.0 V was used with a capacitance of 0.05 F m^{-2} and specific surface area of 500 m^2 cm^{-3}. Equal anion and cation diffusivities of 2e-9 m^2 s^{-1} was used for sodium and chlorine ions. Macro and micro porosities were defined for the modified Donnan model as 0.4 and 0.3 respectively.



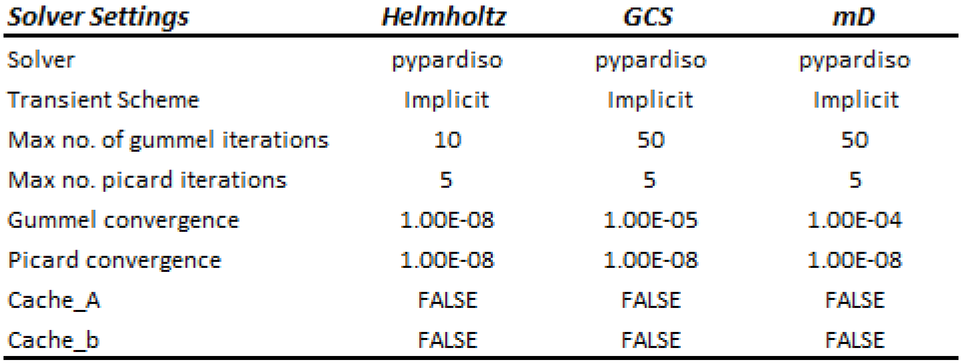
Parameter	*Variable*	*Value*	*Units*
Cell Voltage	V_{cell}	1.0	*V*
Diffusivity	D	2.00E-09	$m^2\ s^{-1}$
Macroporosity	ρ_{mA}	0.4	
Microporosity	ρ_{mi}	0.3	
Specific Capacitance	C	0.05	$F\ m^{-2}$
Specific Surface Area	a	500	$m^2\ cm^{-3}$

To model charging dynamics, a solution procedure was developed for solving coupled mass and ionic transport algorithms. Figure 4.1 shows a flow chart of a procedure for solving mass and ionic transport equations simultaneously with select EDL models. This procedure was adapted from the work by Agnaou et al. who developed a Multiphysics solver for coupling multiple Nernst Planck algorithms (one for each ionic species) with charge conservation [71]. In this work, the Multiphysics solver developed by Agnaou et al. was adapted for coupling a transient fickian diffusion algorithm with a transient ionic transport algorithm. The procedure starts by defining mass and ionic transport algorithms. Then, it proceeds by applying initial conditions. Not shown in the procedure is how the initial condition for potential in the spacer was determined by running a steady state ionic transport algorithm with fixed potential boundary conditions in anode and cathode pores. After defining initial conditions and setting up the algorithms, source terms are applied to the algorithms. These source terms depend on the EDL model selected. After setting source terms, a Multiphysics solver was set up and run to solve the algorithms in a coupled manner. Time marching occurs to solve the transient solver based on initial, final, output, and time step. The time step used for this work was $1s$. Time marching continues until either the system reached steady state according to a specified tolerance, in this case 1e-

6, or the final time step was reached. The solver returns the concentration and potential at each output time specified.

Figure 4.1 describes how the Multiphysics solver works in more detail. The multiphysics solver is made up of Gummel and Picard iterations following the solution scheme by Agnaou et al. [71]. Picard iterations refers to the iterative process of solving a single transport algorithm. A gummel iteration is the successive solving of each transport algorithm. In the case of charging dynamics of a CDI cell, there are two transport algorithms: ionic and mass. The Multiphysics solver starts by solving the mass transport algorithms until some picard convergence criteria is reached. Following picard convergence, physics models are regenerated based on the recently solved for concentration. This includes updating the ionic conductance model which depends on salt concentration (see equation 3.8). The ionic conductance is relied on heavily to solve the ionic transport system to calculate potential. A custom ionic conductance algorithm was written that assumes electroneutrality, equal cation and anion diffusivities, and monovalent salt solution. It is important that system matrices A and b are not cached but recalculated on each gummel iteration to ensure ionic conductance values are updated based on changing concentration. Following convergence of the Picard loop, model value are recalculated including source term models which depend on the potential. The Donnan potential is calculated using scipy's `fsolve` on a potential balance. The code was written with the option to use scipy's `newton_krylov` for larger systems. This process of solving mass transport followed by ionic transport is one gummel iteration and is repeated until the gummel convergence criteria is met and time step reached. Following gummel convergence, concentration and potential values are stored for calculating transient source term in the following time step.

Table 2 Comparison of solver settings used for all three EDL models. The solver settings used across all three EDL models were kept mostly consistent. Each model used the `pypardiso` solver and implicit transient scheme. Criteria for picard convergence were consistent with maximum iterations of 5 and 1e-8 tolerance across all models. Gummel convergence criteria, however, varied across each of the EDL models with mD having the highest tolerance for improved computational efficiency. In all cases, matrices A and b were not cached so that they could be updated upon each gummel iteration.



Solver Settings	*Helmholtz*	*GCS*	*mD*
Solver	pypardiso	pypardiso	pypardiso
Transient Scheme	Implicit	Implicit	Implicit
Max no. of gummel iterations	10	50	50
Max no. picard iterations	5	5	5
Gummel convergence	1.00E-08	1.00E-05	1.00E-04
Picard convergence	1.00E-08	1.00E-08	1.00E-08
Cache_A	FALSE	FALSE	FALSE
Cache_b	FALSE	FALSE	FALSE

There are certain solver settings that must be defined for solving our coupled system of ionic and mass transport equations. Table 2 compares solver settings used across all three EDL models. This includes the solver family, solver type, transient scheme, and convergence criteria. The *pypardiso* solver was used for mass and ionic transport algorithms. The implicit scheme was used for solving these transient algorithms. In general, it was easier and computationally more efficient for the Helmholtz model to reach convergence. This is represented in how the maximum number of gummel iterations used for GCS and mD was 50 but only 10 for Helmholtz. As well, gummel convergence criteria for GCS and mD were increased from 1e-8 to 1e-5 and 1e-4 respectively to help improve computational efficiency. The modified Donnan model was the slowest. That is why the convergence criteria was set highest. Picard convergence criteria was consistent for each of the models with 1e-8 convergence and 5 maximum number of iterations. The code used for each EDL model is available in appendix B.

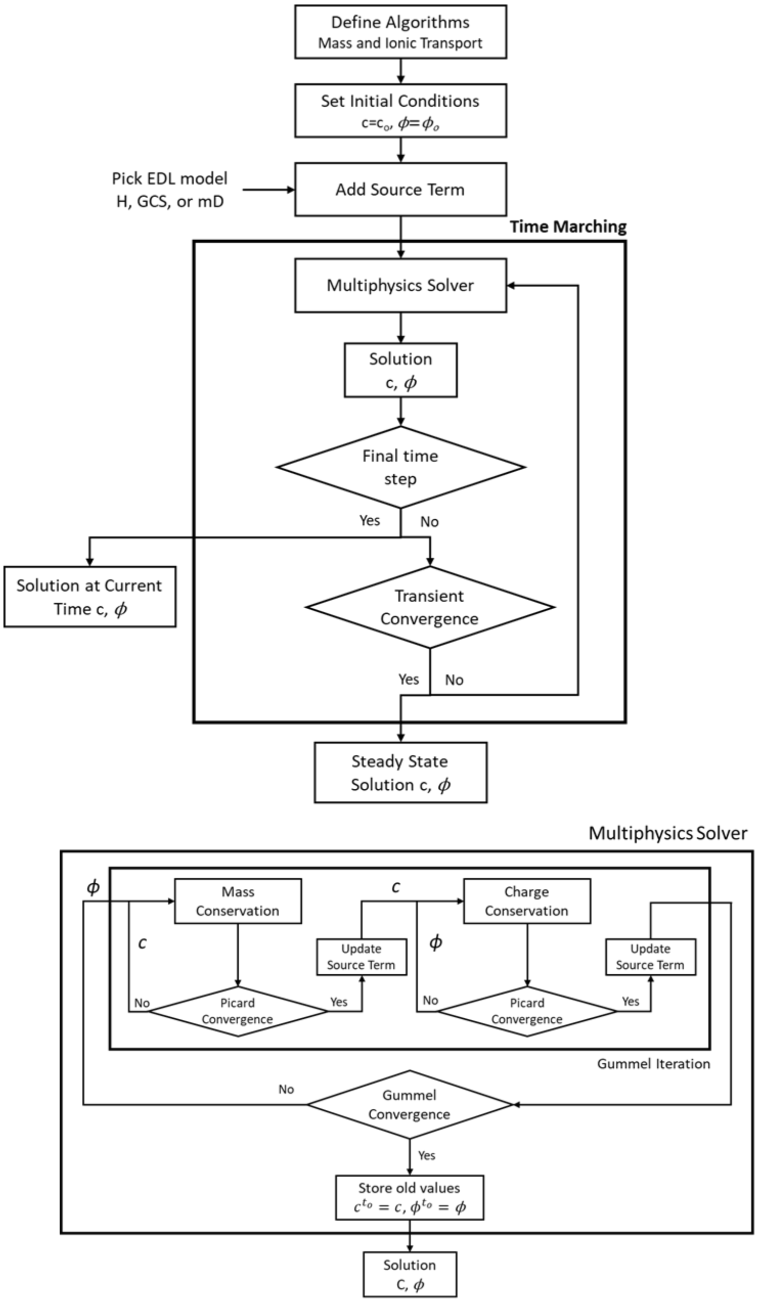


Figure 4.1 Procedure for solving charging dynamics in a pore network model. Ionic and mass transport equations are defined, initial conditions set, and source terms applied based on selected EDL model. Time marching begins around a multi-physics solver that couples ionic and mass transport equations. The procedure the multi-physics solver uses is described in more detail showing how the mass algorithm is solved followed by the charge conservation equation and repeated until gummel convergence is reached.

4.1.1 Charging Dynamics Using Electrical Double Layer Model

In this section, the charging dynamics that can be modeled using electrical double layer theory are demonstrated on a pore network. Figure 4.2 shows the charging dynamics for concentration (a) and potential (b) at times: 0s, 10s, 50s, and 250s using the Gouy-Chapman-Stern EDL model. Time zero shows the initial conditions for concentration and potential as described in section 3.4.1. The difference in potential at time zero between anode and cathode pores drives the storage of ions into electrical double layers and ions stop electrosorbing when the potential everywhere has reached $0V$. After the potential gradient has subsided, diffusive effects persist moving ions from the spacer pores into the electrode until steady state is reached and the concentration is uniform (~ 250s). The capacitance of the spacer pores was set to zero and hence no electrosorption of ions or accumulation of charge occurs within spacer pores. The volumetric capacitance of electrode pores was multiplied by the sum of the pore volume and half of neighbouring throat volumes in order to consider the electrosorption of ions in the electrode throats. Interconnecting throats, that is those throats connecting electrode to spacer pores, however, were not considered to electrosorb ions and therefore did not have volumes added to neighboring electrode pores. Here, only the charging dynamics that use the GCS EDL model are presented but the charging dynamics from all EDL models were studied. Results from all models could have been shown but for the sake of brevity only results from one of the models was presented. The charging dynamics are strikingly similar for each of the EDL models applied. The main difference is the final salt concentration and the time scale. Given the parameters used in Table 1, the average final salt concentrations for Helmholtz, GCS, and modified Donnan in all pores were 46.8 mM, 66.5 mM, and 92.7 mM respectively.

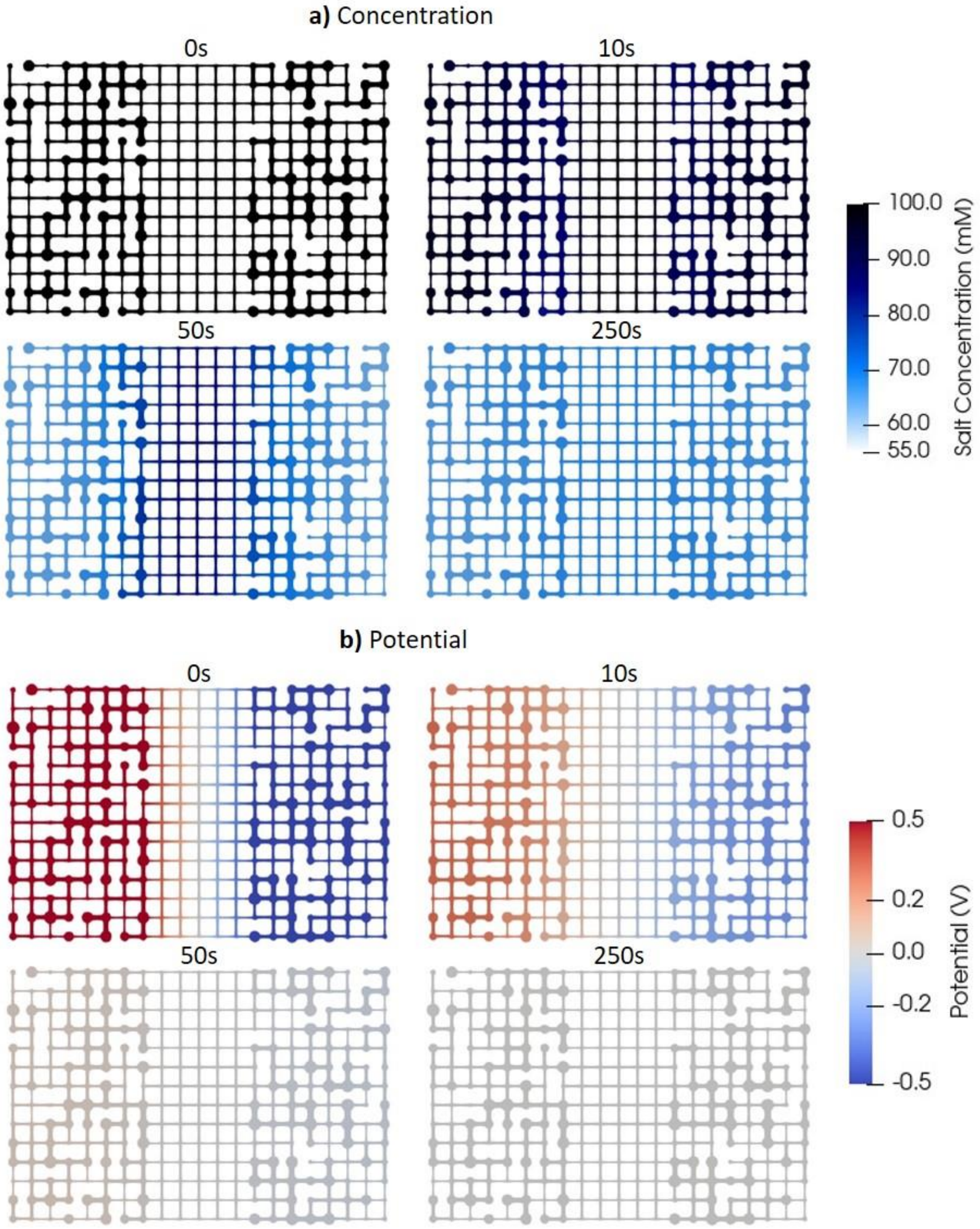


Figure 4.2 Charging dynamics of concentration (a) and potential (b) using the Gouy-Chapman-Stern EDL model at times 0s, 10s, 50s, and 250s. The potential gradient drives the uptake of ions into electrical double layers followed by diffusive effects moving ions from spacer to electrode pores.

4.1.2 Error Analysis of EDL Models

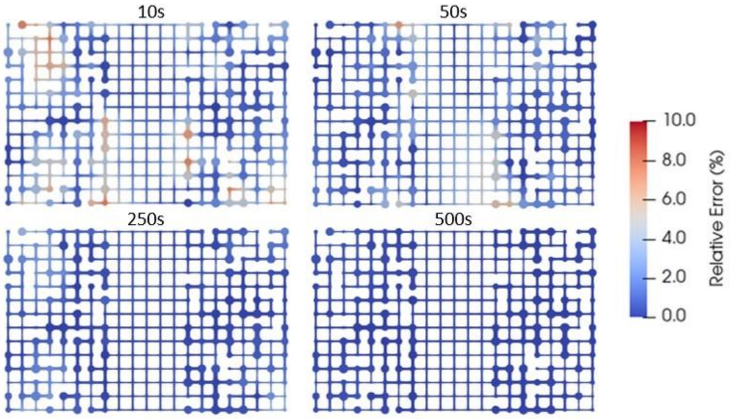
In this section, the error was calculated by comparing the PNM solution to FEM results. `COMSOL` was used as the finite element method solver and it was treated as the true solution. The pore network model relied on the NMEs derived in section 3.3 while the FEM solution was based on the continuum model equations derived in section 3.2. Equations 4.1 and 4.2 show how to calculate the absolute relative error for concentration and potential. These error calculations are similar to those calculated by Agnaou et al. except that the errors are relative to the initial condition, c_o and ϕ_o, and not the FEM solution [71]. The reason why is that, for this problem, the potential approaches zero. Hence, dividing by values approaching zero can result in poor error characterization. Therefore, this approach of dividing by the initial conditions was developed. Recall that the initial conditions were 100 mM for the concentration and 0.5 V for the potential. In the numerator, the PNM quantity is compared to the FEM solution. Pore network modelling returns a pore volume averaged quantity. For improved accuracy, half of the neighbouring throat volumes were added to each pore volume and this was referred to as the effective volume. Therefore, the FEM solver quantities (i.e. concentration and potential) were averaged over each pore and half of its neighbouring throat volumes for comparison to the PNM quantity for every pore.

$$\varepsilon_c = \left|\frac{c_{mA,i}^{PNM} - c_{mA,i}^{FEM}}{c_o}\right| \times 100\% \qquad 4.1$$

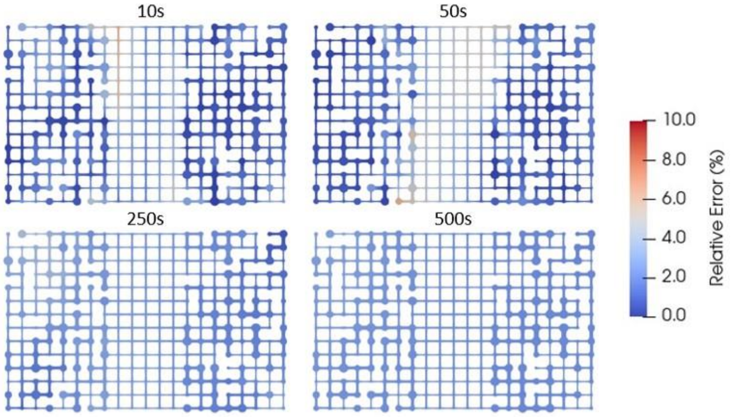
$$\varepsilon_\phi = \left|\frac{\phi_i^{PNM} - \phi_i^{FEM}}{\phi_o}\right| \times 100\% \qquad 4.2$$

Figure 4.3 and Figure 4.4 shows a colour map of relative error calculations for concentration and potential, respectively. This was done for each of the three models: Helmholtz (a), Gouy-Chapman-Stern (b), and modified Donnan (c). The concentration errors were calculated at times 10s, 50s, 250s, and 500s while potential errors were calculated at times 10s, 20s, 50s, and 250s. A larger time span was considered for concentration due to its larger time scale caused by diffusive effects that precede potential driven reactions. Notice that the concentration errors in Figure 4.3 do not exceed 10%. In fact, the maximum concentration error observed was 7.6%, 7.9%, and 1.6% for Helmholts, GCS, and mD models respectively. As for potential errors, Figure 4.4 shows that the relative error does not exceed 15%. The maximum potential error observed was 13.4%, 12.2%, and 11.5% for Helmholtz, GCS, and mD models respectively. For both potential and concentration errors, the relative error decreases as time increases. This is likely caused by a smoothing of initial conditions done in `COMSOL` to obtain better convergence. It is therefore possible that errors could be even less than what is recorded here but the actual effect of smoothing initial conditions done in `COMSOL` on the error was not investigated in this work. Another observation is the relatively lower errors in the modified Donnan case. In Figure 4.3, it appears that that the modified Donnan model has exceptionally low errors. This result was shocking because the gummel tolerance for the mD case was the largest of any of the other EDL models. It is therefore possible, but was not investigated, that a small tolerance causes the solver to converge away from the FEM solution. More work should be done to study the effect of tolerance on observed errors. In no way, as of yet, does it suggest that the mD model is more accurate when applied to a pore network. The only firm conclusion that can be made so far is that each EDL model was applied to a pore network correctly and compares reasonably well to a traditional FEM solution.

a) Helmholtz



b) Gouy-Chapman-Stern



c) modified Donnan

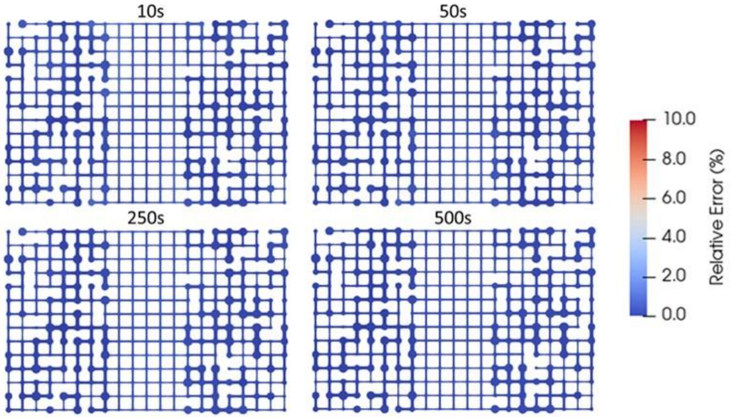
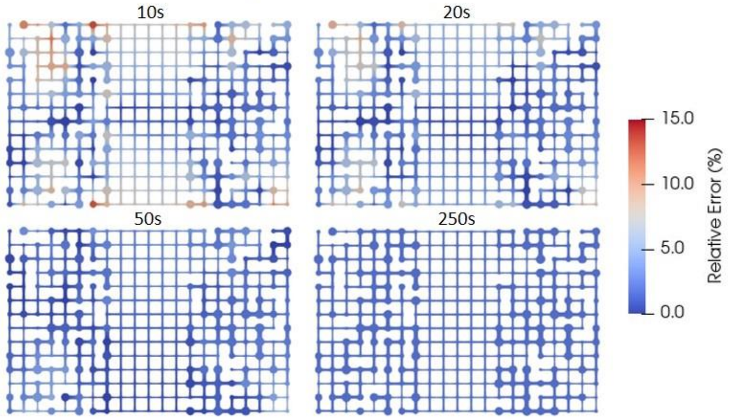
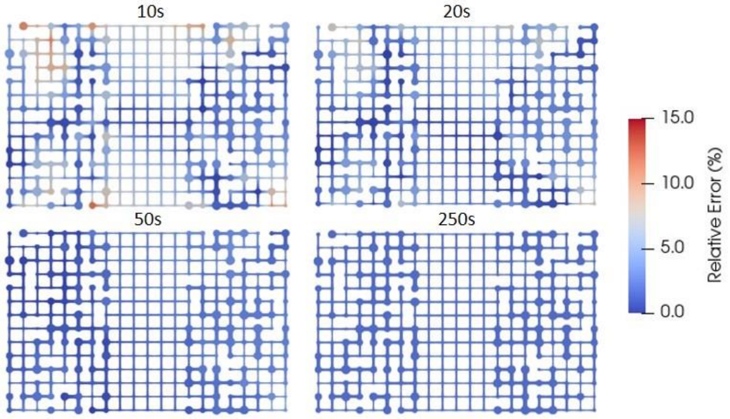


Figure 4.3 The relative error of concentration comparing the FEM solution to results from PNM model equations for the Helmholtz (a), Gouy-Chapman-Stern (b), and modifed Donnan (c) EDL models. The relative error is less than 10% for each of the models determination of concentration.

a) Helmholtz



b) Gouy-Chapman-Stern



c) modified Donnan

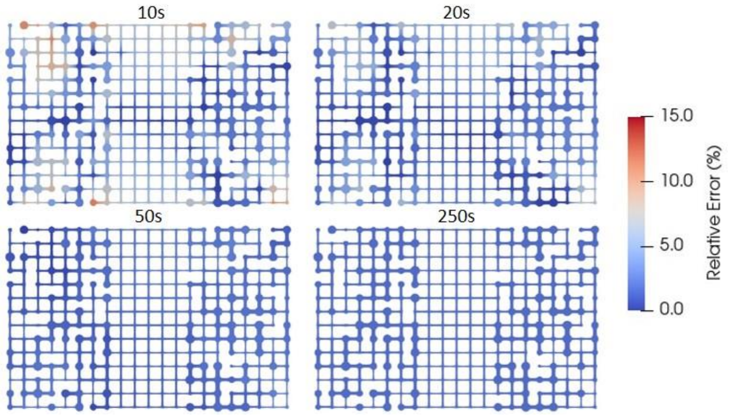


Figure 4.4 The relative error of potential comparing the FEM solution to results from PNM model equations for the Helmholtz (a), Gouy-Chapman-Stern (b), and modifed Donnan (c) EDL models. The relative error is less than 15% for each of the models determination of potential.

4.1.3 Discharge Curve Demonstration of a Pore Network

Following charging of a stop flow CDI cell there is a flow stage to collect the treated water. In the flow stage, the applied voltage is maintained to ensure stored ions are not released from electrical double layers and back into the bulk solution. A pressure gradient is applied across the cell to displace the treated water with new feed water. The pressure gradient used was only 0.001 Pa because the same 2D network was used as charging in sections 4.1.1 and 4.1.2 (to demonstrate only). The permeability of this network is unrealistically high at 95,268 darcy due to the 2D geometry explaining why such a small pressure drop is required. Discharge curves on a more realistic network are performed in section 4.2. During this flow stage a discharge curve can be obtained by measuring the outflow concentration with time. A discharge curve was simulated using an advection-diffusion algorithm described in section 3.3.3 and initial/boundary conditions in section 3.4. Figure 4.5 shows discharge curves for four different applied voltages: 0.6 V (a), 0.8 V (b), 1.0 V (c), and 1.2 V (d). The three different EDL models used for the storage of ions into the micropores are compared. The blue, yellow, and green curves represent the modified Donnan, Gouy-Chapman-Stern, and Helmholtz model respectively. These discharge curves perfectly measure the treated effluent leaving the CDI cell as opposed to most experiments where a conductivity sensor is placed some distance downstream from the outflow port of the CDI cell resulting in mixing with untreated water [9]. Modelling the experimental situation of a conductivity sensor downstream is considered in section 4.2 and proved to be tedious. The modified Donnan model exhibits the smallest amount of salt adsorption by having the highest trough in the discharge curves. This is due to the models assumption of overlapping electrical double layers. Overlapping electrical double layers decrease the amount of salt adsorption in the micropores. The Helmholtz model assumes unit charge efficiency and is therefore the ideal case for a given applied voltage, capacitance, and specific surface area. It was however possible for the GCS model to store more ions than theoretically possible and therefore careful consideration must be made when choosing parameters for the GCS model. In the figure shown, it appears that the GCS model electrosorbs less salt than the Helmholtz model. This is expected behaviour due to the consideration of co-ion expulsion by GCS theory. Adjusting the parameters caused the GCS model to deviate from expected behaviour. In fact, for a cell voltage of 1.0 V, capacitance of 0.1 F m^{-2}, and specific surface area of 300 m^2 cm^{-3}, the GCS model had a charge efficiency of 1.05 after 250s of charging. This demonstrates how the GCS model can be impractical depending on the range of parameters. However, a full parametric study was not conducted and so it is recommended that the effect of parameters on performance metrics such as charge efficiency, salt adsorption, and salt adsorption rate be completed for each of these EDL theories applied to pore networks before any major conclusions are drawn. The purpose here is to simply demonstrate the usefulness of pore networks for determining a discharge curve.

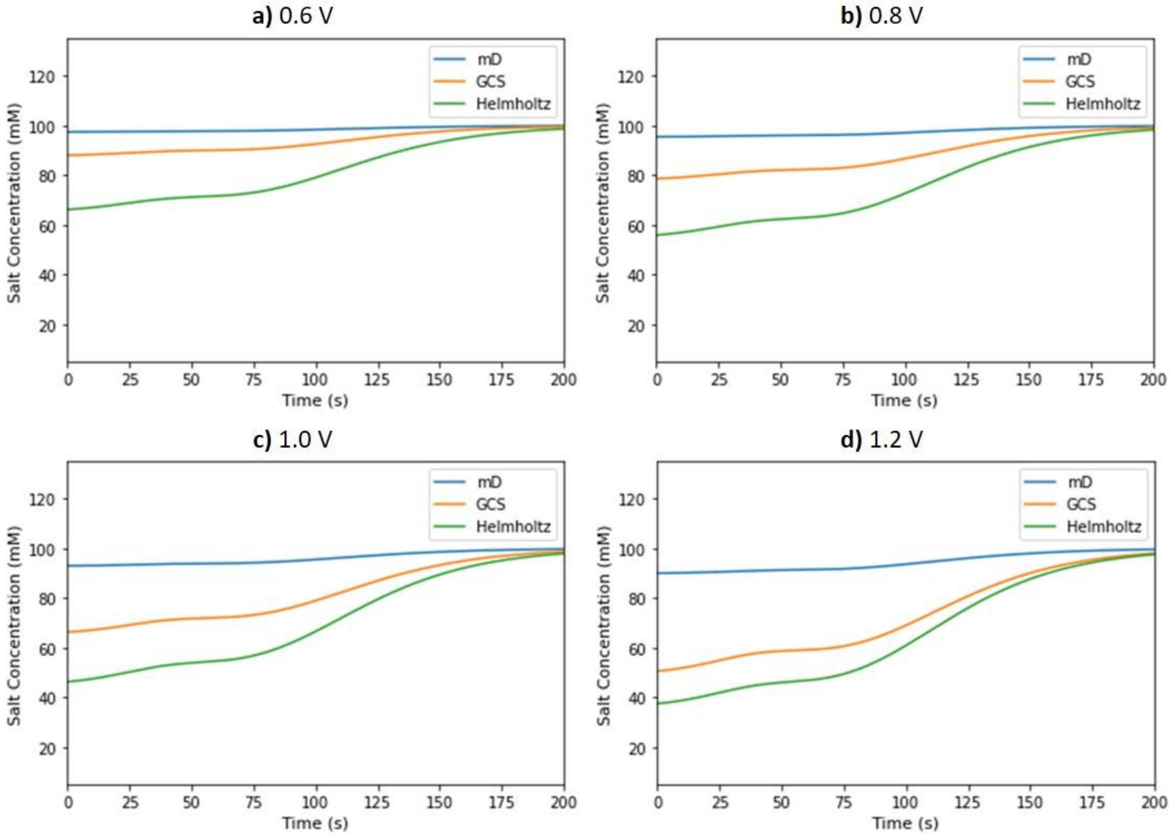


Figure 4.5 Discharge curves of flow stage in a stop flow CDI cell obtained from pore network modelling. The three EDL models are represented as different line colours: modified Donnan (blue), Gouy-Chapman-Stern (GCS), and Helmholtz (Green). The voltage is varied from 0.6 V (a), 0.8 V (b), 1.0 V (c), and 1.2 V (d)

4.2 Comparison to Experiment

In this section, an attempt was made to fit our model to experimental data obtained from a literature report on stop-flow cells [9]. In that work, a flow through architecture was used with an electrode-separator-electrode assembly. The stop flow CDI cell was constructed of two hierarchical carbon aerogel monoliths (HCAM) electrodes of thickness 1mm each and a porous polypropylene separator material of thickness 25µm. The thin separator demonstrated the benefits of using the flow through CDI cell design. On this CDI cell, a stop flow experiment was conducted where the cell was charged for $5\tau_{\mathrm{RC}}$ (see equation 4.4) with no flow. Following charging, a pressure difference, ΔP, of 10 kPa was applied to cause flow through the CDI cell. At this pressure, a flow rate, Q, of 0.5 mL min^{-1} was achieved. Based on this information, the permeability was calculated based on equation 4.3 where L is the length of the domain and μ is the viscosity of the fluid. The viscosity used was that of water which is 0.893 mPa s. The cross-sectional area, A, of the CDI cell by Suss et al. was 2 × 2 cm. The resulting permeability was therefore, 3.768 mD.

$$K = \frac{Q\mu L}{A\Delta P} \tag{4.3}$$

4.2.1 Calibration of the Artificial Network

An artificial network of pores was synthesized and calibrated to match the pore size distribution, permeability, tortuosity, and porosity of the electrode and electrode-separator assembly in Suss et al. [9]. First, the synthesized electrode network was made on a cubic lattice with average coordination number of 5. The average coordination number was chosen to achieve a tortuosity between 3 and 4 which is typical of porous carbons including aerogels [76], [77]. The tortuosity of the artificial network was 3.96 which is in the range for porous carbon electrodes. Next, the pore size distribution was calibrated by matching the simulated intrusion curve of the artificial electrode network to the actual intrusion curve of the HCAM electrodes recorded in reference [9]. Figure 4.6 shows the mercury intrusion curve for the HCAM electrode in black. The intrusion curve shows a bimodal distribution of micro and macro sized pores. The first hump represents pores of approximately 1μm. These pores contribute to 70% of the pore volume. The second hump represents nano sized pores and it's these micropores that contribute to 98% of the surface area [9]. The synthesized pore network attempts only to resolve the macro sized pores. Therefore, only the first hump of the intrusion curve is matched. The blue curve in Figure 4.6 is the simulated intrusion curve for the artificial network. Scipy's `weibull_min` percent point function was used to simulate a weibull pore size distribution and parameters manually adjusted until satisfied with the fit of the intrusion curve. The smallest pore diameters were set to be above 1.4 µm and the scale of the distribution (i.e. what controls the width) was set to 2.6 µm. A shape of 0.4 was set meaning that most of the values are clustered at the low end of the distribution with a long tail.

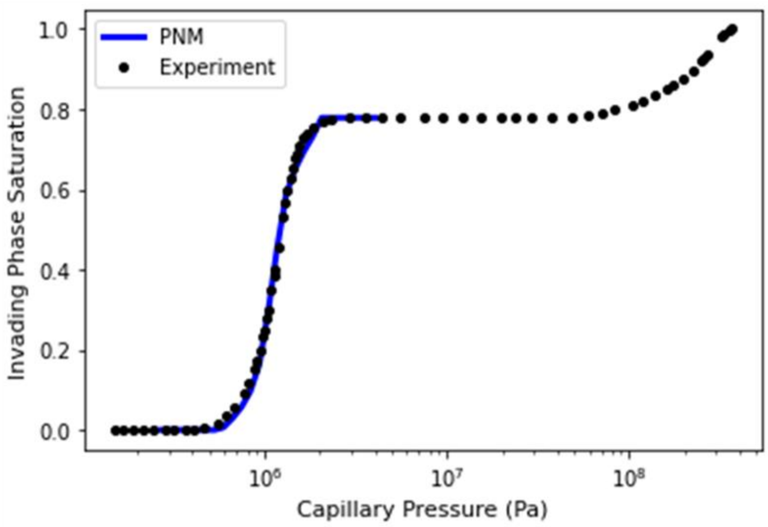


Figure 4.6 The pore size distribution of the artificial electrode network was calibrated to the actual hierarchical carbon aerogel monoliths used in experiment by comparing mercury intrusion curves. The mercury intrusion curve from experiment (black) was matched with the intrusion curve simulated on the artificial pore network (blue).

After matching the experimental intrusion curve (at least the macropore region), the pore size distribution was found. Figure 4.7 shows the resulting Weibull distribution with pore diameters ranging from 1.4 to 6 µm and throat diameters from 0.7 to 2 µm. The precise range of pore sizes recorded in Suss et al. for HCAM electrodes used in their study were 0.7 to 2 µm. However, it is important to note that the authors arrived at those pore sizes from mercury intrusion data. Although, results from intrusion data are often recorded as pore sizes, these are indeed throat sizes. That is, therefore, why our range of throat diameters perfectly match the pore size range recorded by Suss et al. [9].

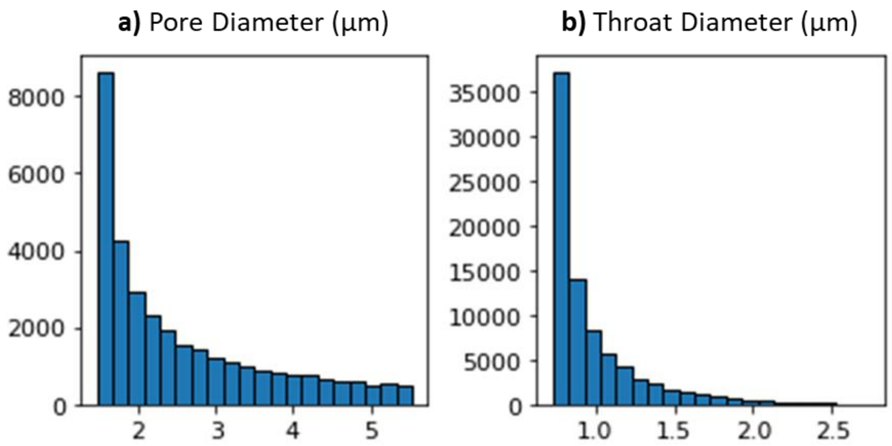
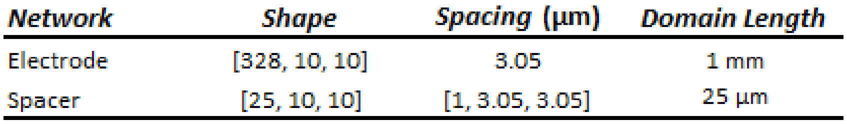


Figure 4.7 The pore and throat size distribution of the calibrated electrode network on a cubic lattice. We show a) histogram of pore diameter and b) histogram of throat diameter. The pore diameter ranges from about 1.4 to 6 µm. Meanwhile, the throat diameters range from 0.7 to 3 µm. The pore size distribution is skewed to smaller pores. This agrees with HCAM electrodes in reference [9] which recorded throat sizes primarily between 0.7 and 2 µm.

The porosity was adjusted by changing the spacing of the network. The porosity of the synthetic electrode network was 53.3% which is similar to the macroporosity of the HCAM electrodes in reference [9]. The porosity of the HCAM electrodes was about 75% but only 70% of that is macropore volume. Therefore, the macroporosity of HCAM electrodes was about 52.5% which is close to the synthesized networks porosity.

Table 3 Dimensions and spacing of electrode and spacer networks calibrated to match electrode-spacer assembly in Suss et al. [9]. The domain length was matched with experiment but only a fraction of the total 2 by 2 cm cross sectional area was modeled for computational efficiency.



Network	*Shape*	*Spacing* (µm)	*Domain Length*
Electrode	[328, 10, 10]	3.05	1 mm
Spacer	[25, 10, 10]	[1, 3.05, 3.05]	25 µm

Finally, the full electrode-spacer-electrode assembly was generated. Figure 4.8 is a visualization of the calibrated 3D network. The visualization focuses on the 25 µm spacer and only a fraction of the electrodes on either side are visualized. The electrodes used had the previous pore size distribution, porosity, and tortuosity. Table 3 lists the shape and spacing used for the electrode and spacer networks. The two electrode networks are identical in shape and spacing. While the average coordination number is the same for each electrode, throats were randomly selected from each to be removed and therefore, the electrodes are not perfect matches but good enough to call the system symmetric. The electrode and spacer had shape and domain selected in order to preserve the domain length (i.e. in the direction of flow). Only a small cross-sectional area of 10 by 10 pores was modelled in order to save on computational effort. The spacer network was stitched between the two electrode networks. The spacer pore diameter was adjusted until the permeability of the electrode-spacer-electrode assembly was close to the value recorded by Suss et al. [9]. The measured permeability of the synthetic network was 3.767 mD using a uniform spacer pore diameter of 0.67 μm. Note that the actual polypropylene spacer had 60 nm pores [9].

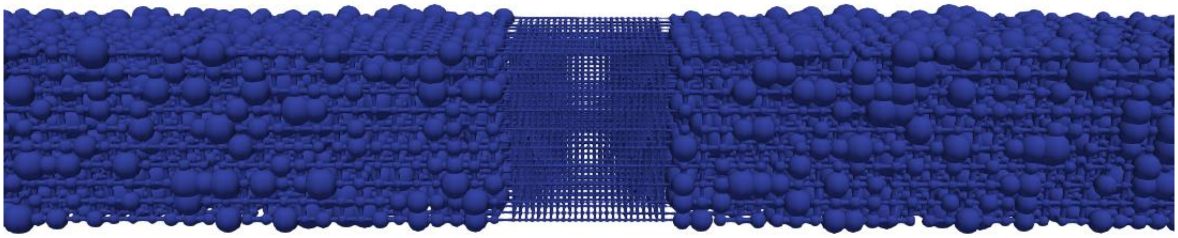


Figure 4.8 3D pore network calibrated to match pore size distribution, permeability, and porosity of actual electrode-spacer assembly in Suss et al. [9]. The visualization shows the entire 25 μm spacer but only a small part of the the 1mm length electrodes on either side. Only a 10 pore by 10 pore cross-section was modelled for sake of computational efficiency.

4.2.2 Simulated Discharge Curve

Once the synthetic network was calibrated, charging and flow stimulations were performed on the network according to initial and boundary conditions in section 3.4. The initial concentration was 100 mM and the cell voltage was 1.0 V to match charging conditions in Suss et al. [9]. In the paper by Suss et al., it was decided that charging proceeds based on a resistive-capacitive (RC) timescale and the RC time was derived. Using this approach, a RC timescale for electrode-spacer-electrode assembly with symmetric electrodes was derived. Equation 4.4 is the derived RC timescale where ε_e and ε_s is the porosity of the electrode and spacer respectively while τ_e and τ_s are the electrode and spacer tortuosity, respectively. The tortuosity of the synthetic spacer network was 4.53 and the porosity was 0.05. The ionic conductivity of the solution, K, was calculated using equation 3.8. The cross-sectional area of the domain is denoted by A while the electrode and spacer domain length are denoted by l_e and l_s respectively. In the experiment, charging was completed for $5\tau_{RC}$. The RC timescale was calculated for our synthetic network using equation 4.4 and got 74.5s. Therefore, the synthesized CDI cell was charged for 199s to match the charging time of the stop flow CDI cell in experiment [9].

$$\tau_{RC} = \left(\frac{2K\varepsilon_e A}{l_e \tau_e} + \frac{K\varepsilon_s A}{l_s \tau_s}\right)^{-1} C_{cell} \qquad 4.4$$

The cell capacitance is C_{cell} which is in terms of faraday's only and determined from the specific capacitance. The specific capacitance used here was 0.2 F m^{-2} which was multiplied by the active specific surface area. Recall that the specific surface area, a, gives the active surface area per macropore volume. Therefore, to determine the number of farads, the volumetric capacitance was multiplied by the volume of the electrode times the electrode macroporosity. The specific surface area was used as a fitting parameter to help match the experimental discharge curve to the modelled discharge curve.

The modified Donnan EDL model was used to model the charging in the synthesized cell because it was assumed that there will be overlapping electrical double layers. In HCAM electrodes, the micropores are in the nano-scale while the Debye length for a 100 mM solution was calculated to be about 1 nm using equations 2.2 and 2.3 while the diffuse layer extends about 2 to 3 times the Debye length [5]. Therefore, the modified Donnan model is most appropriate here for micropores less than 3 nm. In the HCAM electrodes, however, the sub-10nm pores were responsible for most of the electrosorption of ions and so, in this work, the liberty was taken to assume that overlapping EDLs occur in all micropores less than 10 nm and hence the modified Donnan model applies.

After charging the stop flow CDI cell, the flow stage was simulated by applying 10 kPa of pressure across the electrode-spacer-electrode assembly. Figure 4.9 shows the resulting discharge curve compared to experiment in reference [9]. One of the main difficulties was accounting for the pipe in between the outlet port of the sop flow CDI cell and the conductivity sensor. This pipe length is expected to cause mixing with untreated water and effect the time scale. Accounting for this pipe proved to be tedious. In order to account for the extra pipe length, an additional two pores were added to the outlet of the network to simulate a pipe. The diameter of these pores was scaled according to the scaled cross-sectional area of the synthetic network. It was unknown exactly what the length or distance was of the conductivity sensor from the pipe. Therefore, different pipe lengths were tried until satisfied. Eventually, a spacing of 20 cm was used between pore centers. The specific surface area used to get a satisfying match was 400 m^2 cm^{-3}. BET measurement for specific surface area of HCAM electrodes achieve specific surface area of 495 m^2 cm^{-3}. It is expected that not all of that surface area is accessible and participating in the charging dynamics. In fact, one paper concluded in their study that only 16% of the BET surface area participated in electrosorption [78]. Another paper by Zhao et al. concluded that much more of the BET area, 55% of it, was used for electrosorption [66].

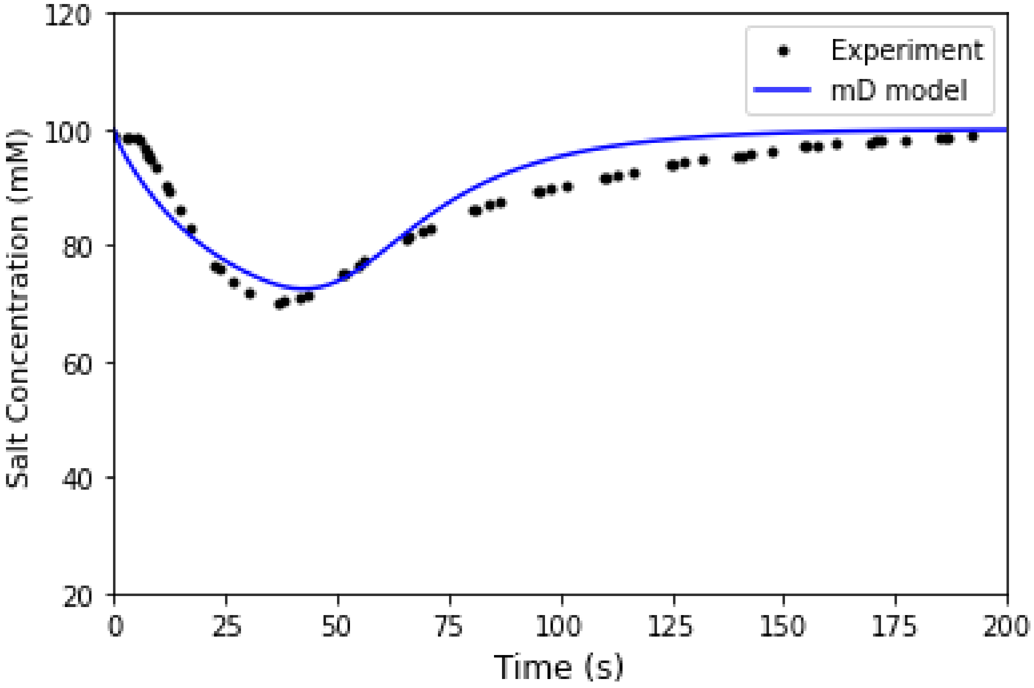


Figure 4.9 Discharge curve of flow stage of stop flow CDI cell following charging. Pore network model results using the modified Donnan model were compared to experiment and results show good agreement. A capacitance of 0.2 F m^{-2} and specific surface area of 400 m^2 cm^{-3} was used. Modelling the effect of the pipe length between the sensor and outlet port of the CDI cell was difficult.

In conclusion, attempting to match experimental observations is difficult yet. Selecting the right EDL model and the correct capacitance and specific surface area heavily influence results. It is difficult to know exactly the surface area available for electrosorption. In addition, there may be turbulence or other phenomena not being captured in modelling the discharge curve. Modelling the extra pipe length between outlet port and conductivity sensor was tedious and should have been avoided by proper experimental design. Overall, however, this was a good first attempt at using a pore network model to match real experiment results.

Chapter 5

Conclusions and Future Works

In summary, electrical double layer models were applied to a pore network for modelling the storage of ions during charging of a stop flow CDI cell. Continuum and numerical method equations were derived and compared using `COMSOL` and `OpenPNM` as FEM and PNM solvers respectively. Model equations were written for three of the most common EDL models: Helmholtz, Gouy-Chapman-Stern, and modified Donnan. Each of these EDL models have already been used in studying CDI but most often applied in a continuum model. This work was the first to apply EDL theory to a pore network model. A test 2D geometry of 294 pores on an irregular cubic lattice was used to demonstrate the use of EDL models in a pore network. The test geometry consisted of anode, cathode, and spacer subdomains. In the spacer, there was no electrosorption of ions and therefore, EDL theories were not applied here. Charging and flow stage of a stop flow CDI cell was modelled using the test network. A stop flow CDI cell was modelled because there is no flow during charging. This allowed attention to be given to the source terms of our model equations that depend on appropriate EDL theory. In addition, electroneutrality in the macropores, equal anion and cation diffusivities, and monovalent salt solution were assumed in derived PDEs and NMEs.

Results from applying EDL theory to a pore network showed good agreement with FEM solution and experimental observations. First, the charging dynamics of each EDL model were demonstrated on a pore network. Next, the errors of each were investigated and it was found that the error in concentration and potential was always less than 10% and 15% respectively showing good agreement with ground truth FEM solution. The errors progressively decayed with charging time for all EDL models likely due to smoothing of initial conditions in `COMSOL` for faster convergence. The error was the least for the modified Donnan model. However, it is important to clarify that this work means by no way to suggest that the modified Donnan model is more accurate. Discharge curve obtained from the pore network model showed how Helmholtz is the ideal case of unit charge efficiency while modified Donnan case had the smallest charge efficiency due to assumption of overlapping EDLs. Careful consideration for parameters was necessary to ensure that the GCS model did not show theoretically impossible charge efficiency greater than unity. Lastly, a network was synthesized and calibrated to match electrode-spacer assembly by Suss et al. [9]. The synthetic network was calibrated based on pore size distribution, permeability, and porosity. Charging and flow simulations were performed on synthetic network and discharge curve compared to that found in literature for a charging time of 199s, capacitance of 0.2 F m^{-2}, and specific surface area of 400 m^2 cm^{-3}. Results found good agreement with experiment, but it was challenging to model effect of extra pipe length between conductivity sensor and outflow port.

There are several future works that could be done to expand on this work. These future works have been summarized as follows:

- Revisions to the assumptions of derived model equations including incorporating flow during charging, applying unequal diffusivities, or considering more complex contaminates other than simple monovalent salt solution.
- Simulate entire charge-discharge cycle of a CDI cell by expanding upon derived NMEs. Only the charging stage of a CDI cycle was modelled in this work where ions are stored in EDLs. The next step would be to model the discharge where stored ions are released from EDLs and back into macropores.

- Conduct a full parametric study on the EDL models applied to a pore network to determine the effect of parameters on performance metrics such as charge efficiency, salt adsorption, and salt adsorption rate. This may include the effect of pore structure on performance metrics.
- Complete more rigorous comparison of PNM result to experiment observations. The experiment that was compared had a conductivity sensor placed downstream of the outlet port. This introduced other phenomena difficult to model precisely. Therefore, it is recommended to conduct experiment where the effluent concentration is more precisely determined. One idea would be to evacuate the downstream pipe leading to sensor before the flow stage.
- Investigate the effects of gummel tolerance on the error between FEM and PNM solvers. The modified Donnan model used a tolerance greater than the other methods. It is unsure whether the low error observed in the modified Donnan case was due to the nature of this EDL theory or the different solver parameters used.
- Errors diminished during charging of the CDI cell in all EDL models. It was thought that this occurs because of smoothing initial conditions done in `COMSOL`. However, more work needs to be done to investigate the effect of smoothing initial conditions on the error.

Bibliography

[1] M. M. Mekonnen and A. Y. Hoekstra, "Four billion people facing severe water scarcity," *Sci. Adv.*, vol. 2, no. 2, 2016.

[2] J. Eke, A. Yusuf, A. Giwa, and A. Sodiq, "The global status of desalination: An assessment of current desalination technologies, plants and capacity," *DESALINATION*, vol. 495, Dec. 2020.

[3] S. K. Patel *et al.*, "The relative insignificance of advanced materials in enhancing the energy efficiency of desalination technologies," *ENERGY Environ. Sci.*, vol. 13, no. 6, pp. 1694–1710, Jun. 2020.

[4] M. E. Suss, S. Porada, X. Sun, P. M. Biesheuvel, J. Yoon, and V. Presser, "Water desalination via capacitive deionization: What is it and what can we expect from it?," *Energy Environ. Sci.*, vol. 8, no. 8, pp. 2296–2319, 2015.

[5] S. Porada, R. Zhao, A. Van Der Wal, V. Presser, and P. M. Biesheuvel, "Review on the science and technology of water desalination by capacitive deionization," *Progress in Materials Science*, vol. 58, no. 8. Elsevier Ltd, pp. 1388–1442, 2013.

[6] S. K. Patel, M. Qin, W. S. Walker, and M. Elimelech, "Energy Efficiency of Electro-Driven Brackish Water Desalination: Electrodialysis Significantly Outperforms Membrane Capacitive Deionization," *Environ. Sci. Technol.*, 2020.

[7] R. Liu, S. Yao, Y. Li, and J. Cheng, "Pore-scale study of dynamic ion adsorption process in porous electrodes of capacitive deionization using lattice Boltzmann method," *Int. J. Heat Mass Transf.*, vol. 135, pp. 769–781, 2019.

[8] T. N. Baroud and E. P. Giannelis, "Role of Mesopore Structure of Hierarchical Porous Carbons on the Electrosorption Performance of Capacitive Deionization Electrodes," *ACS Sustain.*

Chem. Eng., vol. 7, no. 8, pp. 7580–7596, 2019.

[9] M. E. Suss *et al.*, “Capacitive desalination with flow-through electrodes,” *Energy Environ. Sci.*, 2012.

[10] H. H. G. Savenije, “Water scarcity indicators; the deception of the numbers,” *Phys. Chem. EARTH PART B-HYDROLOGY Ocean. Atmos.*, vol. 25, no. 3, pp. 199–204, 2000.

[11] United Nations, “Water Scarcity,” 2021. [Online]. Available: https://www.unwater.org/water-facts/scarcity/. [Accessed: 01-Jul-2021].

[12] A. E. Ercin and A. Y. Hoekstra, “Water footprint scenarios for 2050: A global analysis,” *Environ. Int.*, vol. 64, pp. 71–82, Mar. 2014.

[13] C. J. Vorosmarty, P. Green, J. Salisbury, and R. B. Lammers, “Global water resources: Vulnerability from climate change and population growth,” *Science (80-.).*, vol. 289, no. 5477, pp. 284–288, Jul. 2000.

[14] T. Oki, Y. Agata, S. Kanae, T. Saruhashi, D. W. Yang, and K. Musiake, “Global assessment of current water resources using total runoff integrating pathways,” *Hydrol. Sci. JOURNAL-JOURNAL DES Sci. Hydrol.*, vol. 46, no. 6, pp. 983–995, Dec. 2001.

[15] T. Oki and S. Kanae, “Global hydrological cycles and world water resources,” *Science (80-.).*, vol. 313, no. 5790, pp. 1068–1072, Aug. 2006.

[16] N. Hanasaki *et al.*, “An integrated model for the assessment of global water resources Part 2: Applications and assessments,” *Hydrol. EARTH Syst. Sci.*, vol. 12, no. 4, pp. 1027–1037, 2008.

[17] Y. Wada, L. P. H. van Beek, D. Viviroli, H. H. Duerr, R. Weingartner, and M. F. P. Bierkens, “Global monthly water stress: 2. Water demand and severity of water stress,” *WATER Resour. Res.*, vol. 47, Jul. 2011.

[18] Y. Wada, L. P. H. van Beek, and M. F. P. Bierkens, “Modelling global water stress of the recent

past: on the relative importance of trends in water demand and climate variability," *Hydrol. EARTH Syst. Sci.*, vol. 15, no. 12, pp. 3785–3808, 2011.

[19] A. Y. Hoekstra, M. M. Mekonnen, A. K. Chapagain, R. E. Mathews, and B. D. Richter, "Global Monthly Water Scarcity: Blue Water Footprints versus Blue Water Availability," *PLoS One*, vol. 7, no. 2, Feb. 2012.

[20] M. S. Islam, T. Oki, S. Kanae, N. Hanasaki, Y. Agata, and K. Yoshimura, "A grid-based assessment of global water scarcity including virtual water trading," in *INTEGRATED ASSESSMENT OF WATER RESOURCES AND GLOBAL CHANGE*, 2007, p. 19+.

[21] M. Kummu, P. J. Ward, H. de Moel, and O. Varis, "Is physical water scarcity a new phenomenon? Global assessment of water shortage over the last two millennia," *Environ. Res. Lett.*, vol. 5, no. 3, 2010.

[22] J. Alcamo, M. Floerke, and M. Maerker, "Future long-term changes in global water resources driven by socio-economic and climatic changes," *Hydrol. Sci. JOURNAL-JOURNAL DES Sci. Hydrol.*, vol. 52, no. 2, pp. 247–275, Apr. 2007.

[23] I. Ihsanullah, M. A. Atieh, M. Sajid, and M. K. Nazal, "Desalination and environment: A critical analysis of impacts, mitigation strategies, and greener desalination technologies," *Sci. Total Environ.*, vol. 780, Aug. 2021.

[24] GWI/DESALDATA, "DesalData," 2018. [Online]. Available: https://www.desaldata.com/. [Accessed: 07-May-2018].

[25] E. Jones, M. Qadir, M. T. H. van Vliet, V. Smakhtin, and S. Kang, "The state of desalination and brine production: A global outlook," *Sci. Total Environ.*, vol. 657, pp. 1343–1356, Mar. 2019.

[26] M. Qasim, M. Badrelzaman, N. N. Darwish, N. A. Darwish, and N. Hilal, "Reverse osmosis

desalination: A state-of-the-art review," *DESALINATION*, vol. 459, pp. 59–104, Jun. 2019.

[27] M. Al-Shammiri and M. Safar, "Multi-effect distillation plants: state of the art," *DESALINATION*, vol. 126, no. 1–3, pp. 45–59, Nov. 1999.

[28] M. A. Darwish and N. M. Al-Najem, "Energy consumption by multi-stage flash and reverse osmosis desalters," *Appl. Therm. Eng.*, vol. 20, no. 5, pp. 399–416, 2000.

[29] J. W. BLAIR and G. W. MURPHY, "Electrochemical Demineralization of Water with Porous Electrodes of Large Surface Area," 1960.

[30] J. C. Farmer, D. V Fix, G. V Mack, R. W. Pekala, and J. F. Poco, "The use of capacitive deionization with carbon aerogel electrodes to remove inorganic contaminants from water." 1995.

[31] J. C. Farmer, D. V Fix, G. V Mack, J. F. Poco, and J. K. Nielsen, "No Title." 1995.

[32] A. M. Johnson and J. Newman, "Desalting by Means of Porous Carbon Electrodes," *J. Electrochem. Soc.*, vol. 118, no. 3, pp. 510–517, 1971.

[33] M. Zhang and W. Kong, "Recent progress in graphene-based and ion-intercalation electrode materials for capacitive deionization," *J. Electroanal. Chem.*, vol. 878, Dec. 2020.

[34] M. E. Suss, S. Porada, X. Sun, P. M. Biesheuvel, J. Yoon, and V. Presser, "Water desalination via capacitive deionization: what is it and what can we expect from it?," *Energy Environ. Sci.*, vol. 8, no. 8, pp. 2296–2319, Jul. 2015.

[35] P. M. Biesheuvel, R. Zhao, S. Porada, and A. van der Wal, "Theory of membrane capacitive deionization including the effect of the electrode pore space," *J. Colloid Interface Sci.*, vol. 360, no. 1, pp. 239–248, 2011.

[36] A. Campione, L. Gurreri, M. Ciofalo, G. Micale, A. Tamburini, and A. Cipollina, "Electrodialysis for water desalination: A critical assessment of recent developments on process

fundamentals, models and applications," *Desalination*, vol. 434, pp. 121–160, 2018.

[37] J.-B. Lee, K.-K. Park, H.-M. Eum, and C.-W. Lee, "Desalination of a thermal power plant wastewater by membrane capacitive deionization," *DESALINATION*, vol. 196, no. 1–3, pp. 125–134, Sep. 2006.

[38] X. Gao, A. Omosebi, J. Landon, and K. L. Liu, "Surface charge enhanced carbon electrodes for stable and efficient capacitive deionization using inverted adsorption-desorption behavior," *ENERGY Environ. Sci.*, vol. 8, no. 3, pp. 897–909, 2015.

[39] J. Lee, S. Kim, C. Kim, and J. Yoon, "Hybrid capacitive deionization to enhance the desalination performance of capacitive techniques," *ENERGY Environ. Sci.*, vol. 7, no. 11, pp. 3683–3689, Nov. 2014.

[40] S. Jeon *et al.*, "Desalination via a new membrane capacitive deionization process utilizing flow-electrodes," *ENERGY \& Environ. Sci.*, vol. 6, no. 5, pp. 1471–1475, May 2013.

[41] R. Zhao, P. M. Biesheuvel, and A. van der Wal, "Energy consumption and constant current operation in membrane capacitive deionization," *Energy Environ. Sci.*, vol. 5, no. 11, pp. 9520–9527, Oct. 2012.

[42] L. Wang and S. Lin, "Membrane Capacitive Deionization with Constant Current vs Constant Voltage Charging: Which Is Better?," *Environ. Sci. Technol.*, vol. 52, no. 7, pp. 4051–4060, Apr. 2018.

[43] J.-H. Choi, "Comparison of constant voltage (CV) and constant current (CC) operation in the membrane capacitive deionisation process," *Desalin. WATER Treat.*, vol. 56, no. 4, pp. 921–928, Oct. 2015.

[44] J. Kang, T. Kim, K. Jo, and J. Yoon, "Comparison of salt adsorption capacity and energy consumption between constant current and constant voltage operation in capacitive

deionization," *DESALINATION*, vol. 352, pp. 52–57, Nov. 2014.

[45] Y. Qu *et al.*, "Energy consumption analysis of constant voltage and constant current operations in capacitive deionization," *Desalination*, vol. 400, pp. 18–24, 2016.

[46] Y.-W. Chen, J.-F. Chen, C.-H. Lin, and C.-H. Hou, "Integrating a supercapacitor with capacitive deionization for direct energy recovery from the desalination of brackish water," *Appl. Energy*, vol. 252, p. 113417, 2019.

[47] A. SOFFER, "ELECTRICAL DOUBLE-LAYER OF HIGH SURFACE POROUS CARBON ELECTRODE," *J. Electroanal. Chem.*, vol. 38, no. 1, pp. 25-\&, 1972.

[48] R. Zhao, O. Satpradit, H. H. M. Rijnaarts, P. M. Biesheuvel, and A. van der Wal, "Optimization of salt adsorption rate in membrane capacitive deionization," *Water Res.*, vol. 47, no. 5, pp. 1941–1952, 2013.

[49] S. Porada *et al.*, "Direct prediction of the desalination performance of porous carbon electrodes for capacitive deionization," *ENERGY Environ. Sci.*, vol. 6, no. 12, pp. 3700–3712, Dec. 2013.

[50] E. Avraham, M. Noked, Y. Bouhadana, A. Soffer, and D. Aurbach, "Limitations of charge efficiency in capacitive deionization: II. on the behavior of CDI cells comprising two activated carbon electrodes," *J. Electrochem. Soc.*, vol. 156, no. 10, pp. P157–P162, 2009.

[51] E. Avraham, Y. Bouhadana, A. Soffer, and D. Aurbach, "Limitation of Charge Efficiency in Capacitive Deionization," *J. Electrochem. Soc.*, vol. 156, no. 6, pp. P95–P99, 2009.

[52] L. Wang, J. E. Dykstra, and S. Lin, "Energy Efficiency of Capacitive Deionization," *Environ. Sci. Technol.*, vol. 53, no. 7, pp. 3366–3378, Apr. 2019.

[53] M. Qin *et al.*, "Comparison of energy consumption in desalination by capacitive deionization and reverse osmosis," *Desalination*, vol. 455, pp. 100–114, 2019.

[54] W. Dianbudiyanto and S.-H. Liu, "Outstanding performance of capacitive deionization by a

hierarchically porous 3D architectural graphene," *Desalination*, vol. 468, 2019.

[55] F. Zhang *et al.*, "Multiscale Pore Network Boosts Capacitance of Carbon Electrodes for Ultrafast Charging," *Nano Lett.*, 2017.

[56] Y. Liu, C. Nie, X. Liu, X. Xu, Z. Sun, and L. Pan, "Review on carbon-based composite materials for capacitive deionization," *RSC Adv.*, vol. 5, no. 20, pp. 15205–15225, 2015.

[57] J. Cao, Y. Wang, L. Wang, F. Yu, and J. Ma, "Na3V2(PO4)3@C as Faradaic Electrodes in Capacitive Deionization for High-Performance Desalination," *Nano Lett.*, vol. 19, no. 2, pp. 823–828, Feb. 2019.

[58] Y. Jiang, S. I. Alhassan, D. Wei, and H. Wang, "A Review of Battery Materials as CDI Electrodes for Desalination," *WATER*, vol. 12, no. 11, Nov. 2020.

[59] H. Helmholtz, "Ueber einige Gesetze der Vertheilung elektrischer Ströme in körperlichen Leitern, mit Anwendung auf die thierisch-elektrischen Versuche (Schluss.)," *Ann. Phys.*, vol. 165, no. 7, pp. 353–377, 1853.

[60] H. Helmholtz, "Helmholtz's theory of double electric layers," *J. Franklin Inst*, vol. 115, p. 310, 1883.

[61] S. Porada *et al.*, "Water desalination using capacitive deionization with microporous carbon electrodes," *ACS Appl. Mater. Interfaces*, 2012.

[62] D. L. Chapman, "LI. A contribution to the theory of electrocapillarity," *London, Edinburgh, Dublin Philos. Mag. J. Sci.*, vol. 25, no. 148, pp. 475–481, 1913.

[63] G. Gouy, *Sur la constitution de la charge electrique a la surface d'un electrolyte*, vol. 9. .

[64] O. Stern, "The theory of the electrolytic double shift," *ZEITSCHRIFT FUR ELEKTROCHEMIE UND Angew. Phys. CHEMIE*, vol. 30, pp. 508–516, 1924.

[65] P. M. Biesheuvel, "Thermodynamic cycle analysis for capacitive deionization," *J. Colloid*

Interface Sci., vol. 332, no. 1, pp. 258–264, 2009.

[66] R. Zhao, P. M. Biesheuvel, H. Miedema, H. Bruning, and A. van der Wal, "Charge efficiency: A functional tool to probe the double-layer structure inside of porous electrodes and application in the modeling of capacitive deionization," *J. Phys. Chem. Lett.*, 2010.

[67] M. Z. Bazant, K. Thornton, and A. Ajdari, "Diffuse-charge dynamics in electrochemical systems.," *Phys. Rev.*, vol. 70, no. 2 Pt 1.

[68] H. A. Arafat, M. Franz, and N. G. Pinto, "Effect of salt on the mechanism of adsorption of aromatics on activated carbon," *LANGMUIR*, vol. 15, no. 18, pp. 5997–6003, 1999.

[69] E. N. Guyes, A. N. Shocron, A. Simanovski, P. M. Biesheuvel, and M. E. Suss, "A one-dimensional model for water desalination by flow-through electrode capacitive deionization," *Desalination*, vol. 415, pp. 8–13, 2017.

[70] A. Hemmatifar, M. Stadermann, and J. G. Santiago, "Two-Dimensional Porous Electrode Model for Capacitive Deionization," *J. Phys. Chem. C*, vol. 119, no. 44, pp. 24681–24694, 2015.

[71] M. Agnaou, M. A. Sadeghi, T. G. Tranter, and J. T. Gostick, "Modeling transport of charged species in pore networks: Solution of the Nernst–Planck equations coupled with fluid flow and charge conservation equations," *Comput. Geosci.*, 2020.

[72] M. A. Sadeghi, M. Agnaou, J. Barralet, and J. Gostick, "Dispersion modeling in pore networks: A comparison of common pore-scale models and alternative approaches," *J. Contam. Hydrol.*, p. 103578, 2019.

[73] S. Honarparvar, X. Zhang, T. Chen, C. Na, and D. Reible, "Modeling technologies for desalination of brackish water — toward a sustainable water supply," *Curr. Opin. Chem. Eng.*, vol. 26, pp. 104–111, 2019.

[74] P. M. Biesheuvel and M. Z. Bazant, "Nonlinear dynamics of capacitive charging and

desalination by porous electrodes," *Phys. Rev. E - Stat. Nonlinear, Soft Matter Phys.*, vol. 81, no. 3, 2010.

[75] T. F. Baumann, M. A. Worsley, T. Y.-J. Han, and J. H. Satcher Jr., "High surface area carbon aerogel monoliths with hierarchical porosity," *J. Non. Cryst. Solids*, vol. 354, no. 29, pp. 3513–3515, Jul. 2008.

[76] D. Jayne, Y. Zhang, S. Haji, and C. Erkey, "Dynamics of removal of organosulfur compounds from diesel by adsorption on carbon aerogels for fuel cell applications," *Int. J. Hydrogen Energy*, vol. 30, no. 11, pp. 1287–1293, 2005.

[77] R. LEYVARAMOS and C. J. GEANKOPLIS, "DIFFUSION IN LIQUID-FILLED PORES OF ACTIVATED CARBON .1. PORE VOLUME DIFFUSION," *Can. J. Chem. Eng.*, vol. 72, no. 2, pp. 262–271, 1994.

[78] P. M. Biesheuvel, B. van Limpt, and A. van der Wal, "Dynamic Adsorption/Desorption Process Model for Capacitive Deionization," *J. Phys. Chem. C*, vol. 113, no. 14, pp. 5636–5640, Apr. 2009.

Appendix

Appendix A Derivation of Model Equations

Derivations for the model equations used in this study are presented here. Generic mass and charge conservation equations are derived and then applied to each of the three electrical double layer (EDL) models: Helmholtz, Gouy-Chapman-Stern, and modified Donnan. The derived models assume electroneutrality in the macropores, no flow, and equal anion and cation diffusivities. For simplicity and conciseness, we consider one dimension only and then after extend to three dimensions.

Mass Conservation

Figure A.1 shows a control volume of dimensions Δx, Δy, and Δz with mass flux in the x-direction and reaction occurring inside the control volume.

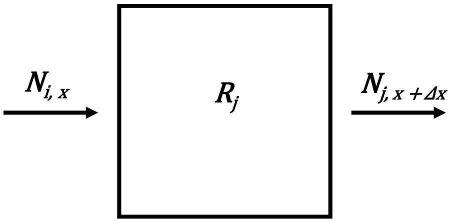


Figure A.1 Side view of control volume showing mass flux in x-direction

Equation A.1 is the mass balance for the above system where $N_{j,x}$ is the mass flux of species j going into the differential volume element at x and $N_{j,x+\Delta\mathrm{x}}$ is the mass flux of species j leaving the differential volume element at $x + \Delta x$. The concentration for species j is denoted as $\mathrm{c}_{\mathrm{mA,j}}$.

$$\Delta x \Delta y \Delta z \frac{\partial c_{mA,j}}{\partial t} = \Delta y \Delta z N_{j,x} - \Delta y \Delta z N_{j,x+\Delta x} - \Delta x \Delta y \Delta z \, R_j \qquad \text{A.1}$$

Electrosorption of species j is considered using a volumetric reaction term R_j with units of $mol/(m^3 s)$. This term is defined later when we specify the EDL model. Equation A.1 is divided by ΔxΔyΔz followed by taking the limit as $\Delta x \rightarrow 0$ to obtain equation A.2.

$$\frac{\partial c_{mA,j}}{\partial t} = -\frac{\partial N_j}{\partial x} - R_j \qquad \text{A.2}$$

Equation A.3 is the Nernst Planck expression for mass flux which considers diffusive, advective, and migrative forces. We apply Nernst Planck to determine the molar flux, N_j. D_j is the diffusion coefficient and z_j is the valence for species j. The variables R, F, and T are the ideal gas constant, faraday number and temperature, respectively. The macropore potential is given by ϕ and the velocity is given by u. For simplicity and conciseness, only the x-direction is considered for this derivation.

$$N_j = -D_j \frac{\partial c_{mA,j}}{\partial x} + u c_{mA,j} - \frac{D_j z_j F}{RT} c_{mA,j} \frac{\partial \phi}{\partial x} \qquad \text{A.3}$$

Equation A.3 is substituted into equation A.2 to give us equation A.4.

$$\frac{\partial c_{mA,j}}{\partial t} = D_j \frac{\partial^2 c_{mA,j}}{\partial x^2} - u \frac{\partial c_{mA,j}}{\partial x} + \frac{D_j z_j F c_{mA,j}}{RT} \frac{\partial^2 \phi}{\partial x^2} - R_j \qquad \text{A.4}$$

For the case of a monovalent salt solution such as NaCl, we can apply equation A.4 to come up with two equations, one for each of the dissolved species' sodium and chlorine, as shown in equation A.5. In deriving these two equations we assume equal anion and cation diffusivities ($D = D_{Na} = D_{Cl}$) as well as set $z_{Cl} = -1$ and $z_{Na} = +1$.

$$\frac{\partial c_{mA,Na}}{\partial t} = D_{Na}\frac{\partial^2 c_{mA,Na}}{\partial x^2} - u\frac{\partial c_{mA,Na}}{\partial x} + \frac{D_{Na}Fc_{mA,Na}}{RT}\frac{\partial^2 \phi}{\partial x^2} - R_{Na}$$

A.5

$$\frac{\partial c_{mA,Cl}}{\partial t} = D_{Cl}\frac{\partial^2 c_{mA,Cl}}{\partial x^2} - u\frac{\partial c_{mA,Cl}}{\partial x} - \frac{D_{Cl}Fc_{mA,Cl}}{RT}\frac{\partial^2 \phi}{\partial x^2} - R_{Cl}$$

Adding these two equations together we get equation A.6. In doing so electroneutrality was assumed such that $c_{mA} = c_{mA,Na} = c_{mA,Cl}$. Note that this assumption as well as assuming equal anion and cation diffusivities leads to the elimination of the migration term. No flow was also assumed resulting in $u = 0$ and the subsequent removal of the advection term. This reduces our system to a transient fickian diffusion with reaction problem. Note that the reaction term R in equation A.6 is the sum of individual species reaction terms such that $R = \sum_j R_j$.

$$\frac{\partial c_{mA}}{\partial t} = \frac{\partial}{\partial x}\left(D\frac{\partial c_{mA}}{\partial x}\right) - R \qquad \text{A.6}$$

Lastly, we extend equation A.6 from one dimension to three dimensions using the Laplacian operator giving equation A.7.

$$\frac{\partial c_{mA}}{\partial t} = \boldsymbol{\nabla} \cdot (D\boldsymbol{\nabla} c_{mA}) - R \qquad \text{A.7}$$

Charge Conservation

Figure A.2 shows a side view of a control volume of dimensions Δx, Δy, and Δz with current flowing in the x-direction only.

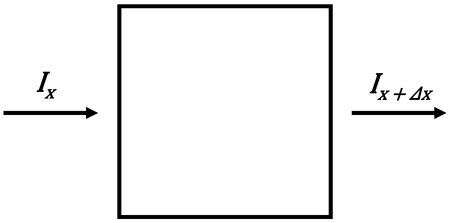


Figure A.2 Side view of control volume showing current flow in x-direction

Equation A.8 is a charge balance on the above system where I_x is the current going into the differential volume element at x and $I_{x+\Delta x}$ is the current leaving the differential volume element at $x + \Delta$x. The accumulation of charge is determined by the voltage drop across the capacitor-like Stern layer denoted by $\Delta\phi_{st}$. The exact determination of $\Delta\phi_{st}$ depends on the EDL model applied but in general it is defined such that decreasing Stern potential represents increasing storage of positive charge.

$$-\Delta x\Delta y\Delta z Ca\frac{\partial\Delta\phi_{st}}{\partial t} = \Delta y\Delta z I_x - \Delta y\Delta z I_{x+\Delta x} \qquad \text{A.8}$$

The volumetric capacitance, Ca, is used to relate the voltage drop across the Stern layer to the amount of charge stored in the electrical double layer. Equation A.8 is divided by $\Delta x\Delta y\Delta z$ and the limit as $\Delta x \rightarrow 0$ is taken to arrive at equation A.9.

$$Ca\frac{\partial\Delta\phi_{st}}{\partial t} = -\frac{\partial I}{\partial x} \qquad \text{A.9}$$

Now we determine an expression for the ionic current. The ionic current can be determined from $I = F\sum_j(z_jN_j)$ where N_j is the mass flux of ionic species j and can be determined from using the Nernst Planck expression in equation A.3. Combining these expressions yield equation A.10 for determining the ionic current.

$$I = -F\sum_j z_jD_j\frac{\partial c_{mA,j}}{\partial x} + Fu\sum_j z_jc_{mA,j} - \frac{F^2}{RT}\frac{\partial\phi}{\partial x}\sum_j z_j^2D_jc_{mA,j} \qquad \text{A.10}$$

For a monovalent salt solution ($z_{Cl} = -1, z_{Na} = +1$) and assuming electroneutrality in the macropores ($c_{mA} = c_{mA,Na} = c_{mA,Cl}$) as well as assuming equal cation and anion diffusivities ($D = D_{Na} = D_{Cl}$) equation A.10 reduces to equation A.11.

$$I = -\frac{2F^2Dc_{mA}}{RT}\frac{\partial\phi}{\partial x} \qquad \text{A.11}$$

Equation A.11 is substituted into equation A.9 for the ionic current to give equation A.12.

$$Ca\frac{\partial\Delta\phi_{st}}{\partial t} = -\frac{\partial}{\partial x}\left(\frac{2F^2Dc}{RT}\frac{\partial\phi}{\partial x}\right) \qquad \text{A.12}$$

From equation A.12 an ionic conductance, K, is derived such that $K = \frac{2F^2Dc_{mA}}{RT}$. After applying this ionic conductance and extending to three dimensions using the Laplacian operator we get equation A.13.

$$Ca\frac{\partial\Delta\phi_{st}}{\partial t} = -\boldsymbol{\nabla}\cdot(K\boldsymbol{\nabla}\phi) \qquad \text{A.13}$$

Lastly, if we assume that there is no accumulation of charge, then the left hand side of equation A.13 disappears resulting in equation A.14.

$$0 = \boldsymbol{\nabla} \cdot (K \boldsymbol{\nabla} \phi) \qquad \text{A.14}$$

Reaction Term, R

Here the mass and ionic transport equations previously derived are applied to each of the electrical double layer models. Each model used relies on a specific reaction term, R. The explanation for the reaction term used for each model is recorded here along with treatment of the Stern layer potential for each EDL theory. The electrical double layer models being considered are: Helmholtz, Gouy-Chapman-Stern, and modified Donnan.

a) Helmholtz

The Helmholtz model assumes a capacitor like EDL structure where every electrode surface charge is equally compensated by counter charge in the electrolyte. This model consists of a Helmholtz or Stern layers separating the charges in water and solid electrode phases. The potential drop occurs entirely across the Stern layer. The solid electrode and liquid brine solution acts as two sides to a capacitor. The rate of Electrosorption of ions into the electrical double layer is therefore modeled using simple electrical engineering capacitor theory. Equation A.15 shows how to calculate current flow in a charging or discharging simple capacitor circuit.

$$i = C\frac{d\phi}{dt} \qquad \text{A.15}$$

Recalling that current is just the flow of charge and so this simple expression for current can be extended to the rate of flow of ions into the electrical double layer. Equation A.16 gives an expression for the rate of flow of counterions into the electrical double layer using Faraday's constant, F, to relate moles of charge to moles of ions.

$$R = \frac{a}{2z^{count}F} i \qquad \text{A.16}$$

Substituting equation A.15 into equation A.16 gives an expression for the rate of ions into an electrical double layer in terms of potential yielding equation A.17.

$$R = \frac{Ca}{2z^{count}F} \frac{d\phi}{dt} \qquad \text{A.17}$$

Substituting equation A.17 into equation A.7 for the rate of Electrosorption of ions into the electrical double layer gives equation A.18. This is our mass conservation equation for the Helmholtz model.

$$\frac{\partial c_{mA}}{\partial t} = \boldsymbol{\nabla} \cdot (D \boldsymbol{\nabla} c_{mA}) - \frac{Ca}{2z^{count}F} \frac{\partial \phi}{\partial t} \qquad \text{A.18}$$

Next, the accumulation term of the ionic transport equation is considered. This term contains the Stern layer potential drop. Treatment of the Stern layer potential drop depends on the EDL model. For Helmholtz, the Stern layer potential drop is equal to $\phi_1 - \phi$ as described by equation A.19 where ϕ_1 is the potential of the solid electrode.

$$\Delta\phi_{st} = \phi_1 - \phi \qquad \text{A.19}$$

Equation A.19 is substituted into equation A.13 to get equation A.20. In the case of constant voltage, ϕ_1 does not change with respect to time. Therefore, equation $\text{A}.20$ leads to equation A.21 as ϕ_1 is removed and negative signs cancel.

$$Ca\frac{\partial(\phi_1 - \phi)}{\partial t} = -\boldsymbol{\nabla}\cdot(K\boldsymbol{\nabla}\phi) \qquad \text{A.20}$$

$$Ca\frac{\partial\phi}{\partial t} = \boldsymbol{\nabla}\cdot(K\boldsymbol{\nabla}\phi) \qquad \text{A.21}$$

That concludes the derivation of the Helmholtz model equations. The mass transport equation (A.18) is coupled with the ionic transport equation (A.21).

b) Gouy-Chapman-Stern

Next we derive the reaction term for the Gouy-Chapman-Stern model. Since the diffuse layer is characteristically defined as a region of varying concentration, a surface charge density is calculated representing the amount of charge in the EDL per electrode surface area. The surface charge density is calculated using equation A.22

$$w = 8\lambda_d C_{mA}\sinh^2\left(\frac{F}{4RT}\Delta\phi_d\right) \qquad \text{A.22}$$

To incorporate the surface charge density into the reaction term, R, the rate of change of surface charge density is used. A prefactor of the specific surface area is used to convert units from per electrode area to per macropore volume. Equation A.23 is therefore the reaction term used for the GCS model.

$$R = a\frac{dw}{dt} \qquad \text{A.23}$$

This reaction term is substituted into the generic mass transport expression to arrive at equation A.24. This is the mas transport equation for the GCS model.

$$\frac{\partial c_{mA}}{\partial t} = \mathbf{\nabla} \cdot (D\mathbf{\nabla} c_{mA}) - a\frac{dw}{dt} \qquad \text{A.24}$$

Next, the ionic transport equation to be used with the GCS model is derived. To do this, we start with the generic ionic transport expression in equation A.13. Special treatment of the Stern layer potential is considered. Here the Stern layer potential is calculated using equation A.25. This expression was derived from a potential balance across the EDL structure.

$$\Delta\phi_{st} = \phi_1 - \phi - \Delta\phi_d \qquad \text{A.25}$$

Equation A.25 was substituted into equation A.13 to arrive at equation A.26. Assuming constant voltage operation again, ϕ_1 does not change with time. Expanding and moving the Donnan potential term to the right-hand side equation A.27 is formed. Note that the second term on the right-hand side accounts for co-ion expulsion of ions and can be thought of as a reaction term.

$$Ca\frac{\partial(\phi_1 - \phi - \Delta\phi_d)}{\partial t} = -\mathbf{\nabla} \cdot (K\mathbf{\nabla}\phi) \qquad \text{A.26}$$

$$Ca\frac{\partial\phi}{\partial t} = \mathbf{\nabla} \cdot (K\mathbf{\nabla}\phi) - Ca\frac{\partial\Delta\phi_d}{\partial t} \qquad \text{A.27}$$

That concludes the derivation of the GCS model equations. The mass transport equation (A.24) is coupled with the ionic transport equation (A.27).

c) modified Donnan

Lastly, we derive the reaction term and model equations for the modified Donnan EDL model. The reaction term for the modified Donnan model depends on the micropore concentration. The micropore concentration can be calculated using equation A.28. The micropore concentration is the total concentration of ions in the micropore.

$$c_{mi} = 2c_{mA} \cosh\left(\frac{F}{RT}\Delta\phi_d\right) \tag{A.28}$$

The micropore concentration is used to derive a reaction term for the modified Donnan model. The reaction term, R, will be defined as the rate of change of micropore concentration. Equation A.29 calculates the reaction term. The prefactor $\frac{V^{mi}}{V}$ is used to convert units from per micropore volume to per macropore volume where V^{mi} is micropore volume and V is macropore volume.

$$R = \frac{1}{2}\frac{V^{mi}}{V}\frac{\partial c_{mi}}{\partial t} \tag{A.29}$$

The $^1/_2$ term is used to reflect that the number of salt molecules electrosorbed is half of the total ion concentration in the micropore. It is important to note that, in doing so, electrode symmetry is assumed such that the number of counter-ions in the anode plus the number of co-ions in the cathode equals the number of co-ions in the anode plus the number of counter-ions in the cathode. The expression for reaction term, R, is substituted into equation A.7 to arrive at equation A.30

$$\frac{\partial c_{mA}}{\partial t} = \mathbf{\nabla} \cdot (D\mathbf{\nabla} c_{mA}) - \frac{1}{2}\frac{V^{mi}}{V}\frac{\partial c_{mi}}{\partial t} \tag{A.30}$$

Next, the ionic transport algorithm is defined for the modified Donnan model. In fact, the ionic transport algorithm for modified Donnan is identical to the ionic transport algorithm for GCS. The potential balance across the EDL is the same. The difference is how the Stern and Donnan potentials are calculated. Equation A.31 is the ionic transport equation coupled with equation A.30.

$$Ca\frac{\partial \phi}{\partial t} = \boldsymbol{\nabla} \cdot (K\boldsymbol{\nabla}\phi) - Ca\frac{\partial \Delta\phi_d}{\partial t} \qquad \text{A.31}$$

Appendix B Python Code

The following code was written as a demonstration of applying EDL theories to a pore network. Code was written for: Helmholtz, Gouy-Chapman-Stern, and modified Donnan EDL models. The code includes importing FEM solver data from excel file for comparison with pore network model results and error calculations.

Helmholtz

```
import pandas as pd
import openpnm as op
import helmholtz_models as cdi
import numpy as np

np.random.seed(0)

# network
spacer = 5 # pore width
[Npx, Npy, Npz] = [21, 14, 1]
spacing = 1e-4
net = op.network.Cubic(shape=[Npx, Npy, Npz], spacing=spacing)

net['pore.anode'] = net['pore.coords'][:, 0] < (Npx - spacer)*spacing/2
net['pore.cathode'] = net['pore.coords'][:, 0] > (Npx + spacer)*spacing/2
net['pore.spacer'] = np.logical_and(~net['pore.anode'], ~net['pore.cathode'])
Ps_a = net.pores(labels='anode')
Ps_c = net.pores(labels='cathode')
Ps_s = net.pores(labels='spacer')
Ts_a = net.find_neighbor_throats(pores=Ps_a, mode='xnor')
Ts_c = net.find_neighbor_throats(pores=Ps_c, mode='xnor')
inter_thrt = net.find_neighbor_throats(net.pores(labels='spacer'), mode='xor')

# remove throats
remove = np.random.choice(np.concatenate((Ts_a, Ts_c)), size=int(Ts_a.size/3))
op.topotools.trim(net, throats=remove)
op.topotools.plot_connections(net, throats=net.throats('all'))

# geometry
Ps = net.pores(labels='all')
Ts = net.throats(labels='all')
geo = op.geometry.StickAndBall2D(network=net, pores=Ps, throats=Ts)
geo['pore.diameter'][Ps_s] = geo['pore.diameter'].mean() * 0.5
geo.regenerate_models(exclude=['pore.diameter'])
# add effective volume model
geo['throat.volume'][inter_thrt] = 0  # don't add volume of inter throats
eff_vol = op.models.geometry.pore_volume.effective
geo.add_model(propname='pore.effective_volume', model=eff_vol)
# add pore.capacitance prop without interconnecting throat volume
C = 0.05 # F/m^2
a = 500000000 # m^2/m^3
V = geo['pore.effective_volume']
geo['pore.capacitance'] = C * a * V
geo['pore.capacitance'][Ps_s] = 0
# add back interconecting throat volume
```

```
geo.regenerate_models(exclude=['pore.diameter'])

# phase
salt = op.phases.Water(network=net)
salt['pore.diffusivity'] = 2e-9
salt['pore.valence'] = 1
salt['pore.valence'][Ps_a] = -1
salt['pore.concentration'] = 100

# physics
phys = op.physics.GenericPhysics(network=net, phase=salt, geometry=geo)
phys['pore.capacitance'] = geo['pore.capacitance']

current = op.models.physics.ionic_conductance.myelectroneutrality_2D
phys.add_model(propname='throat.ionic_conductance', model=current,
               regen_mode='normal')

diffusion = op.models.physics.diffusive_conductance.ordinary_diffusion_2d
phys.add_model(propname='throat.diffusive_conductance',
               model=diffusion,
               regen_mode='normal')

# settings for algorithms
settings = {'solver_max_iter': 5, 'solver_tol': 1e-08, 'solver_rtol': 1e-08,
            'nlin_max_iter': 10, 'cache_A': False, 'cache_b': False}

# algorithms
ic = op.algorithms.IonicConduction(network=net, phase=salt, settings=settings)
tfd = op.algorithms.TransientFickianDiffusion(network=net, phase=salt,
                                              settings=settings)
p = op.algorithms.TransientIonicConduction(network=net, phase=salt,
                                           settings=settings)
pnp = op.algorithms.TransientMultiphysicsSolver(network=net, phase=salt)
pnp.setup(algorithms=[tfd.name, p.name])

# update tfd settings
tfd.settings['pore_volume'] = 'pore.effective_volume'

# update charge settings
p.settings['charge_conservation'] = 'laplace'
p.settings['pore_volume'] = 'pore.capacitance'
p.settings['old_prop'] = 'pore.potential'

# run ic algorithm to get initial conditions for potential
ic.set_value_BC(pores=Ps_a, values=0.5)
ic.set_value_BC(pores=Ps_c, values=-0.5)
ic.run()

# initial conditions
tfd.set_IC(values=100)
P_IC = ic['pore.potential']
p.set_IC(values=P_IC)

# time step
```

```
dt = 1
tmax = 2

# Fix time step problem
p.settings['t_step'] = dt
tfd.settings['t_step'] = dt

# multiphysics solver settings
pnp.settings['t_initial'] = 0
pnp.settings['t_final'] = tmax
pnp.settings['t_step'] = dt
pnp.settings['t_output'] = 1
pnp.settings['t_scheme'] = 'implicit'

# add model which fetches 'pore.potential' from potential algorithm
p['pore.potential'] = P_IC
phys.add_model(propname='pore.potential_new',
               model=cdi.fetch_property,
               quantity='pore.potential',
               algorithm=p.name)
# add helmholtz source term model
helmholtz = cdi.helmholtz
phys['pore.potential_old'] = P_IC
phys.add_model(propname='pore.helmholtz',
               model=helmholtz,
               capacitance='pore.capacitance',
               valence='pore.valence',
               potential='pore.potential',
               algorithm=pnp.name)
tfd.set_source(propname='pore.helmholtz', pores=[Ps_a, Ps_c])

# run
pnp.run()

# update results
salt.update(tfd.results())
salt.update(p.results())

# update C and P
# mask = net['pore.coords'][:, 1] == 1.5*spacing  # only want certain pores
mask = net.pores('all')
C = tfd['pore.concentration'][mask]
P = p['pore.potential'][mask]

# Charge Efficiency
V = geo['pore.effective_volume']
c = geo['pore.capacitance']
F = 96485.3321233100184
gamma_salt = np.dot(100-C, V.T)
sigma_charge = np.sum((c*(0.5-P)/F)[net.pores('anode')])
lamda = gamma_salt/sigma_charge
print('salt adsorption (mol):', gamma_salt)
print('charge efficiency:', lamda)
```

```
# get comsol data for comparison, these are the times I am exporting
df_c = pd.read_excel('COMSOL-data-updated.xlsx',
                     sheet_name='Concentration')
df_p = pd.read_excel('COMSOL-data-updated.xlsx',
                     sheet_name='Potential')
columns = [1, 2, 6, 11, 21, 51, 101, 251, 501, 751, 1001]
times = np.arange(0, 1001, 1)
for c in columns:
    salt['pore.COMSOL_potential@'+str(times[c-1])] = df_p.values[:, c]
    salt['pore.COMSOL_concentration@'+str(times[c-1])] = df_c.values[:, c]
    Cfem = salt['pore.COMSOL_concentration@'+str(times[c-1])]
    Pfem = salt['pore.COMSOL_potential@'+str(times[c-1])]
    Cpnm = salt['pore.concentration@'+str(times[c-1])]
    Ppnm = salt['pore.potential@'+str(times[c-1])]
    # Error
    salt['pore.concentration_error@'+str(times[c-1])] = np.abs((Cpnm - Cfem)/100)*100
    salt['pore.potential_error@'+str(times[c-1])] = np.abs((Ppnm - Pfem)/0.5)*100

# Export to XDMF for paraview
net['throat.radius'] = net['throat.diameter'] / 2
op.io.XDMF.save(network=net, phases=[salt], filename="interpore-conference-1")
```

Gouy-Chapman-Stern

```
import pandas as pd
import openpnm as op
import gcs_models as cdi
import numpy as np

np.random.seed(0)

# network
spacer = 5 # pore width
[Npx, Npy, Npz] = [21, 14, 1]
spacing = 1e-4
net = op.network.Cubic(shape=[Npx, Npy, Npz], spacing=spacing)

net['pore.anode'] = net['pore.coords'][:, 0] < (Npx - spacer)*spacing/2
net['pore.cathode'] = net['pore.coords'][:, 0] > (Npx + spacer)*spacing/2
net['pore.spacer'] = np.logical_and(~net['pore.anode'], ~net['pore.cathode'])
Ps_a = net.pores(labels='anode')
Ps_c = net.pores(labels='cathode')
Ps_s = net.pores(labels='spacer')
Ts_a = net.find_neighbor_throats(pores=Ps_a, mode='xnor')
Ts_c = net.find_neighbor_throats(pores=Ps_c, mode='xnor')
inter_thrt = net.find_neighbor_throats(net.pores(labels='spacer'), mode='xor')

# remove throats
remove = np.random.choice(np.concatenate((Ts_a, Ts_c)), size=int(Ts_a.size/3))
op.topotools.trim(net, throats=remove)
op.topotools.plot_connections(net, throats=net.throats('all'))

# geometry
Ps = net.pores(labels='all')
Ts = net.throats(labels='all')
geo = op.geometry.StickAndBall2D(network=net, pores=Ps, throats=Ts)
geo['pore.diameter'][Ps_s] = geo['pore.diameter'].mean() * 0.5
geo.regenerate_models(exclude=['pore.diameter'])
# add effective volume model
geo['throat.volume'][inter_thrt] = 0  # don't add volume of inter throats
eff_vol = op.models.geometry.pore_volume.effective
geo.add_model(propname='pore.effective_volume', model=eff_vol)
# add pore.capacitance prop without interconnecting throat volume
C = 0.05 # F/m^2
a = 500000000 # m^2/m^3
V = geo['pore.effective_volume']
geo['pore.capacitance'] = C * a * V
geo['pore.capacitance'][Ps_s] = 0
# Surface area
geo['pore.surface_area'] = a * V
# add back interconecting throat volume
geo.regenerate_models(exclude=['pore.diameter'])

# phase
salt = op.phases.Water(network=net)
```

```
salt['pore.diffusivity'] = 2e-9
salt['pore.valence'] = 1
salt['pore.valence'][Ps_a] = -1
salt['pore.concentration'] = 100

# physics
phys = op.physics.GenericPhysics(network=net, phase=salt, geometry=geo)
phys['pore.capacitance'] = geo['pore.capacitance']
phys['pore.surface_area'] = geo['pore.surface_area']
phys['pore.capacitance_specific'] = C

current = op.models.physics.ionic_conductance.myelectroneutrality_2D
phys.add_model(propname='throat.ionic_conductance', model=current,
               regen_mode='normal')

diffusion = op.models.physics.diffusive_conductance.ordinary_diffusion_2d
phys.add_model(propname='throat.diffusive_conductance',
               model=diffusion,
               regen_mode='normal')

# settings for algorithms
settings = {'solver_max_iter': 5, 'solver_tol': 1e-08, 'solver_rtol': 1e-08,
            'nlin_max_iter': 10, 'cache_A': False, 'cache_b': False}

# algorithms
ic = op.algorithms.IonicConduction(network=net, phase=salt, settings=settings)
tfd = op.algorithms.TransientFickianDiffusion(network=net, phase=salt,
                                              settings=settings)
p = op.algorithms.TransientIonicConduction(network=net, phase=salt,
                                           settings=settings)
pnp = op.algorithms.TransientMultiphysicsSolver(network=net, phase=salt)
pnp.settings['g_max_iter'] = 50
pnp.settings['g_tol'] = 1e-5
pnp.setup(algorithms=[tfd.name, p.name])

# update tfd settings
tfd.settings['pore_volume'] = 'pore.effective_volume'
tfd.settings['old_prop'] = 'pore.salt_surface_density'

# update charge settings
p.settings['charge_conservation'] = 'laplace'
p.settings['pore_volume'] = 'pore.capacitance'
p.settings['old_prop'] = 'pore.donnan_potential'

# run ic algorithm to get initial conditions for potential
ic.set_value_BC(pores=Ps_a, values=0.5)
ic.set_value_BC(pores=Ps_c, values=-0.5)
ic.run()

# initial conditions
tfd.set_IC(values=100)
P_IC = ic['pore.potential']
p.set_IC(values=P_IC)
```

```
# time step
dt = 1
tmax = 2

# Fix time step problem
p.settings['t_step'] = dt
tfd.settings['t_step'] = dt

# multiphysics solver settings
pnp.settings['t_initial'] = 0
pnp.settings['t_final'] = tmax
pnp.settings['t_step'] = dt
pnp.settings['t_output'] = 1
pnp.settings['t_scheme'] = 'implicit'

# add debye length model
phys['pore.bjerrum_length'] = 7.189145219266839e-10 # at room temperature

phys.add_model(propname='pore.debye_length',
               model=cdi.debye_length,
               pore_bjerrum_length='pore.bjerrum_length',
               pore_concentration='pore.concentration')

# add donnan potential model
salt['pore.potential'] = P_IC
salt['pore.potential_electrode'] = P_IC
phys.add_model(propname='pore.donnan_potential',
               model=cdi.donnan_potential,
               pore_capacitance='pore.capacitance_specific',
               pore_concentration='pore.concentration',
               pore_debye_length='pore.debye_length',
               pore_potential='pore.potential',
               pore_potential_electrode='pore.potential_electrode',
               pore_temperature='pore.temperature')

# add micropore concentration model
phys.add_model(propname='pore.salt_surface_density',
               model=cdi.salt_surface_density,
               pore_concentration='pore.concentration',
               pore_debye_length='pore.debye_length',
               pore_donnan_potential='pore.donnan_potential',
               pore_temperature='pore.temperature')

# add mass source term
phys['pore.salt_surface_density_old'] = np.zeros(net.Np)
phys.add_model(propname='pore.gcs_source_mass',
               model=cdi.gcs_source_mass,
               pore_surface_area='pore.surface_area',
               pore_salt_surface_density='pore.salt_surface_density',
               algorithm=pnp.name)
tfd.set_source(propname='pore.gcs_source_mass', pores=[Ps_a, Ps_c])

# add potential source term (really same as donnan)
phys['pore.donnan_potential_old'] = np.zeros(net.Np)
```

```
phys.add_model(propname='pore.gcs_source_potential',
               model=cdi.gcs_source_potential,
               pore_capacitance='pore.capacitance',
               pore_donnan_potential='pore.donnan_potential',
               algorithm=pnp.name)
p.set_source(propname='pore.gcs_source_potential', pores=[Ps_a, Ps_c])

# run
pnp.run()

# update results
salt.update(tfd.results())
salt.update(p.results())

# update C and P
# mask = net['pore.coords'][:, 1] == 1.5*spacing  # only want certain pores
mask = net.pores('all')
C = tfd['pore.concentration'][mask]
P = p['pore.potential'][mask]

# Charge Efficiency
V = geo['pore.effective_volume']
c = geo['pore.capacitance']
F = 96485.3321233100184
gamma_salt = np.dot(100-C, V.T)
sigma_charge = np.sum((c*(0.5-P)/F)[net.pores('anode')])
lamda = gamma_salt/sigma_charge
print('salt adsorption (mol):', gamma_salt)
print('charge efficiency:', lamda)

# get comsol data for comparison, these are the times I am exporting
df_c = pd.read_excel('COMSOL-data-updated.xlsx',
                     sheet_name='Concentration')
df_p = pd.read_excel('COMSOL-data-updated.xlsx',
                     sheet_name='Potential')
columns = [1, 2, 3, 6, 11, 21, 26, 51, 76, 101]
times = np.arange(0, 1001, 10)
for c in columns:
    salt['pore.COMSOL_potential@'+str(times[c-1])] = df_p.values[:, c]
    salt['pore.COMSOL_concentration@'+str(times[c-1])] = df_c.values[:, c]
    Cfem = salt['pore.COMSOL_concentration@'+str(times[c-1])]
    Pfem = salt['pore.COMSOL_potential@'+str(times[c-1])]
    Cpnm = salt['pore.concentration@'+str(times[c-1])]
    Ppnm = salt['pore.potential@'+str(times[c-1])]
    # Error
    salt['pore.concentration_error@'+str(times[c-1])] = np.abs((Cpnm - Cfem)/100)*100
    salt['pore.potential_error@'+str(times[c-1])] = np.abs((Ppnm - Pfem)/0.5)*100

# Export to XDMF for paraview
net['throat.radius'] = net['throat.diameter'] / 2
op.io.XDMF.save(network=net, phases=[salt], filename="capacitive_deionization_gcs")
```

modified Donnan

```
import pandas as pd
import openpnm as op
import donnan_models as cdi
import numpy as np

np.random.seed(0)

# network
spacer = 5 # pore width
[Npx, Npy, Npz] = [21, 14, 1]
spacing = 1e-4
net = op.network.Cubic(shape=[Npx, Npy, Npz], spacing=spacing)

net['pore.anode'] = net['pore.coords'][:, 0] < (Npx - spacer)*spacing/2
net['pore.cathode'] = net['pore.coords'][:, 0] > (Npx + spacer)*spacing/2
net['pore.spacer'] = np.logical_and(~net['pore.anode'], ~net['pore.cathode'])
Ps_a = net.pores(labels='anode')
Ps_c = net.pores(labels='cathode')
Ps_s = net.pores(labels='spacer')
Ts_a = net.find_neighbor_throats(pores=Ps_a, mode='xnor')
Ts_c = net.find_neighbor_throats(pores=Ps_c, mode='xnor')
inter_thrt = net.find_neighbor_throats(net.pores(labels='spacer'), mode='xor')

# remove throats
remove = np.random.choice(np.concatenate((Ts_a, Ts_c)), size=int(Ts_a.size/3))
op.topotools.trim(net, throats=remove)
op.topotools.plot_connections(net, throats=net.throats('all'))

# geometry
Ps = net.pores(labels='all')
Ts = net.throats(labels='all')
geo = op.geometry.StickAndBall2D(network=net, pores=Ps, throats=Ts)
geo['pore.diameter'][Ps_s] = geo['pore.diameter'].mean() * 0.5
geo.regenerate_models(exclude=['pore.diameter'])
# add effective volume model
geo['throat.volume'][inter_thrt] = 0  # don't add volume of inter throats
eff_vol = op.models.geometry.pore_volume.effective
geo.add_model(propname='pore.effective_volume', model=eff_vol)
# add pore.capacitance prop without interconnecting throat volume
C = 0.05 # F/m^2
a = 500000000 # m^2/m^3
V = geo['pore.effective_volume']
geo['pore.capacitance'] = C * a * V
geo['pore.capacitance'][Ps_s] = 0
# calculate micropore volume
rho_mi = 0.3
rho_ma = 0.4
geo['pore.micropore_volume'] = rho_mi / rho_ma * V
# add back interconecting throat volume
geo.regenerate_models(exclude=['pore.diameter'])
```

```
# phase
salt = op.phases.Water(network=net)
salt['pore.diffusivity'] = 2e-9
salt['pore.valence'] = 1
salt['pore.valence'][Ps_a] = -1
salt['pore.concentration'] = 100

# physics
phys = op.physics.GenericPhysics(network=net, phase=salt, geometry=geo)
phys['pore.capacitance'] = geo['pore.capacitance']
phys['pore.micropore_volume'] = geo['pore.micropore_volume']

current = op.models.physics.ionic_conductance.myelectroneutrality_2D
phys.add_model(propname='throat.ionic_conductance', model=current,
               regen_mode='normal')

diffusion = op.models.physics.diffusive_conductance.ordinary_diffusion_2d
phys.add_model(propname='throat.diffusive_conductance',
               model=diffusion,
               regen_mode='normal')

# settings for algorithms
settings = {'solver_max_iter': 5, 'solver_tol': 1e-08, 'solver_rtol': 1e-08,
            'nlin_max_iter': 10, 'cache_A': False, 'cache_b': False}

# algorithms
ic = op.algorithms.IonicConduction(network=net, phase=salt, settings=settings)
tfd = op.algorithms.TransientFickianDiffusion(network=net, phase=salt,
                                              settings=settings)
p = op.algorithms.TransientIonicConduction(network=net, phase=salt,
                                           settings=settings)
pnp = op.algorithms.TransientMultiphysicsSolver(network=net, phase=salt)
pnp.settings['g_max_iter'] = 50
pnp.settings['g_tol'] = 1e-4
pnp.setup(algorithms=[tfd.name, p.name])

# update tfd settings
tfd.settings['pore_volume'] = 'pore.effective_volume'
tfd.settings['old_prop'] = 'pore.concentration_micropore'

# update charge settings
p.settings['charge_conservation'] = 'laplace'
p.settings['pore_volume'] = 'pore.capacitance'
p.settings['old_prop'] = 'pore.donnan_potential'

# run ic algorithm to get initial conditions for potential
ic.set_value_BC(pores=Ps_a, values=0.5)
ic.set_value_BC(pores=Ps_c, values=-0.5)
ic.run()

# initial conditions
tfd.set_IC(values=100)
P_IC = ic['pore.potential']
p.set_IC(values=P_IC)
```

```
# time step
dt = 1
tmax = 2

# Fix time step problem
p.settings['t_step'] = dt
tfd.settings['t_step'] = dt

pnp.settings['t_final'] = tmax
pnp.settings['t_step'] = dt# multiphysics solver settings
pnp.settings['t_initial'] = 0

pnp.settings['t_output'] = 1
pnp.settings['t_scheme'] = 'implicit'

# add donnan potential model
salt['pore.potential'] = P_IC
salt['pore.potential_electrode'] = P_IC
salt['pore.capacitance_vol'] = C * a
phys.add_model(propname='pore.donnan_potential',
               model=cdi.donnan_potential,
               pore_capacitance='pore.capacitance_vol',
               pore_concentration='pore.concentration',
               pore_potential='pore.potential',
               pore_potential_electrode='pore.potential_electrode',
               pore_temperature='pore.temperature')

# add micropore concentration model
phys.add_model(propname='pore.concentration_micropore',
               model=cdi.micropore_concentration,
               pore_concentration='pore.concentration',
               pore_donnan_potential='pore.donnan_potential',
               pore_temperature='pore.temperature')

# add mass source term
phys['pore.concentration_micropore_old'] = np.ones(net.Np)*200
phys.add_model(propname='pore.donnan_source_mass',
               model=cdi.donnan_source_mass,
               micropore_volume='pore.micropore_volume',
               micropore_concentration='pore.concentration_micropore',
               algorithm=pnp.name)
tfd.set_source(propname='pore.donnan_source_mass', pores=[Ps_a, Ps_c])

# add potential source term
phys['pore.donnan_potential_old'] = np.zeros(net.Np)
phys.add_model(propname='pore.donnan_source_potential',
               model=cdi.donnan_source_potential,
               pore_capacitance='pore.capacitance',
               pore_donnan_potential='pore.donnan_potential',
               algorithm=pnp.name)
p.set_source(propname='pore.donnan_source_potential', pores=[Ps_a, Ps_c])

# run
pnp.run()
```

```
# update results
salt.update(tfd.results())
salt.update(p.results())

# update C and P
# mask = net['pore.coords'][:, 1] == 1.5*spacing  # only want certain pores
mask = net.pores('all')
C = tfd['pore.concentration'][mask]
P = p['pore.potential'][mask]

# Charge Efficiency
V = geo['pore.effective_volume']
c = geo['pore.capacitance']
F = 96485.3321233100184
gamma_salt = np.dot(100-C, V.T)
sigma_charge = np.sum((c*(0.5-P)/F)[net.pores('anode')])
lamda = gamma_salt/sigma_charge
print('salt adsorption (mol):', gamma_salt)
print('charge efficiency:', lamda)

# get comsol data for comparison, these are the times I am exporting
df_c = pd.read_excel('COMSOL-data.xlsx',
                     sheet_name='Concentration')
df_p = pd.read_excel('COMSOL-data.xlsx',
                     sheet_name='Potential')
columns = [1, 2, 3, 6, 11, 21, 26, 51, 76, 101]
times = np.arange(0, 1001, 10)
for c in columns:
    salt['pore.COMSOL_potential@'+str(times[c-1])] = df_p.values[:, c]
    salt['pore.COMSOL_concentration@'+str(times[c-1])] = df_c.values[:, c]
    Cfem = salt['pore.COMSOL_concentration@'+str(times[c-1])]
    Pfem = salt['pore.COMSOL_potential@'+str(times[c-1])]
    Cpnm = salt['pore.concentration@'+str(times[c-1])]
    Ppnm = salt['pore.potential@'+str(times[c-1])]
    # Error
    salt['pore.concentration_error@'+str(times[c-1])] = np.abs((Cpnm - Cfem)/100)*100
    salt['pore.potential_error@'+str(times[c-1])] = np.abs((Ppnm - Pfem)/0.5)*100

# Export to XDMF for paraview
net['throat.radius'] = net['throat.diameter'] / 2
op.io.XDMF.save(network=net, phases=[salt], filename="capacitive_deionization_donnan")
```

Models were written in separate .py files and imported for use in the previous demonstration of applying EDL theory to a pore network. Those models are recorded as follows for each of the EDL theories: Helmholtz, Gouy-Chaoman-Stern, and modified Donnan.

Helmholtz

```
def helmholtz(target, capacitance='', valence='', potential='', algorithm=''):
    r"""
    Calculates the rate of electrosorption using the Helmholtz capacitor-like
    model for electrical double layers. This model assumes electroneutrality
    in the macropores and electrode symmetry. Therefore, the rate calculated
    is the rate of salt adsorbed not the rate of an individual ion.

        .. math::
            r = - C / z_{counter} / 2 / F * dphi/dt

    Parameters
    ----------
    capacitance : string
        The dictionary key on the target object containing coefficient C, the
        capacitance with units C/V or F

    valence : string
        The dictionary key on the target object containing coefficient
        z_{counter}, the counterion valence

    potential : string
        The dictionary key on the target object containing the potential (phi)

    algorithm : string
        The algorithm name of which the time step is stored

    Returns
    -------
    A dictionary containing the following three items:

        **'rate'** - The value of the source term function at the given X.

        **'S1'** - The slope of the source term function at the given X.

        **'S2'** - The intercept of the source term function at the given X.

    The slope and intercept provide a linearized source term equation about the
    current value of X as follow:

        .. math::
            rate = S_{1}   X  +  S_{2}

    """

    project = target.project
    algorithm = project.algorithms()[algorithm]
```

```
    phase = project.find_phase(target)

    F = 96485.3321233100184

    Ca = target[capacitance]
    phi_new = target[potential + '_new']
    phi_old = target[potential + '_old']
    z = phase[valence]
    dt = algorithm.settings['t_step']

    r = - Ca / z / 2 / F / dt * (phi_new - phi_old)
    S1 = 0
    S2 = - Ca / z / 2 / F / dt * (phi_new - phi_old)

    values = {'S1': S1, 'S2': S2, 'rate': r}

    return values


def fetch_property(target, quantity='', algorithm=''):
    r"""
    The purpose of this method is to update a hard coded property on the
    target object by fetching the property from the algorithm object where
    it's most recent value is stored

    Parameters
    ----------
    quantity : string, required
        The dictionary key on the target object containing the quantity to be
        fetched
    algorithm : string, required
        The name of the algorithm by which the quantity will be fetched

    Returns
    -------
    values : data
        The values of the property fetched from the algorithm object

    """
    project = target.project
    algorithm = project.algorithms()[algorithm]

    target[quantity + '_new'] = algorithm[quantity]
    values = target[quantity + '_new']

    return values
```

Gouy-Chapman-Stern

```
import numpy as np
from scipy.optimize import fsolve
from scipy.optimize import newton_krylov


def gcs_source_mass(target,
                    pore_surface_area='',
                    pore_salt_surface_density='',
                    algorithm=''):
    r"""
    Calculates the rate of electrosorption using the gouy-chapman-stern model
    for electrical double layers. This model assumes electroneutrality
    in the macropores and electrode symmetry. Therefore, the rate calculated
    is the rate of salt adsorbed not the rate of an individual ion.

        .. math::
            r = - A * dw/dt

    Parameters
    ----------
    pore_surface_area : string
        The dictionary key on the target object containing the specific
        surface area, A in units m^{2}

    pore_salt_surface_density : string
        The dictionary key on the target object containing the salt surface
        density. We automatically append '_old' to find the old value.

    algorithm : string
        The algorithm name of which the time step is stored

    Returns
    -------
    A dictionary containing the following three items:

        **'rate'** - The value of the source term function at the given X.

        **'S1'** - The slope of the source term function at the given X.

        **'S2'** - The intercept of the source term function at the given X.

    The slope and intercept provide a linearized source term equation about the
    current value of X as follow:

        .. math::
            rate = S_{1}   X  +  S_{2}

    """
    project = target.project
    algorithm = project.algorithms()[algorithm]
```

```
    A = target[pore_surface_area]
    w_new = target[pore_salt_surface_density]
    w_old = target[pore_salt_surface_density + '_old']
    dt = algorithm.settings['t_step']

    r = - A / dt * (w_new - w_old)
    S1 = 0
    S2 = - A / dt * (w_new - w_old)
    values = {'S1': S1, 'S2': S2, 'rate': r}

    return values


def gcs_source_potential(target,
                         pore_capacitance='',
                         pore_donnan_potential='',
                         algorithm=''):
    r"""
    Source term for considering the effects of co-ion expulsion from micropores
    into macropores. For example, a decrease in the donnan potential in the
    anode signifies positive ions (co-ion) leaving the micropore and entering
    macropore resulting in a gernation of positive ions in the ionic transport
    algorithm. This source term is to be applied to an ionic transport
    algorithm.

        .. math::
            r = - C * d{\Delta\phi}_{d}/dt

    Parameters
    ----------
    pore_capacitance : string
        The dictionary key on the target object containing coefficient C, the
        capacitance with units C/V or F

    pore_donnan_potential : string
        The dictionary key on the target object containing the donnan
        potential denoted as {\Delta\phi}_{d}. We automatically append '_old'
        to find the old value.

    algorithm : string
        The algorithm name of which the time step is stored

    Returns
    -------
    A dictionary containing the following three items:

        **'rate'** - The value of the source term function at the given X.

        **'S1'** - The slope of the source term function at the given X.

        **'S2'** - The intercept of the source term function at the given X.

    The slope and intercept provide a linearized source term equation about the
    current value of X as follow:
```

```
        .. math::
            rate = S_{1}   X  +  S_{2}

    """
    project = target.project
    algorithm = project.algorithms()[algorithm]

    C = target[pore_capacitance]
    phi_d = target[pore_donnan_potential]
    phi_d_old = target[pore_donnan_potential + '_old']
    dt = algorithm.settings['t_step']

    r = - C / dt * (phi_d - phi_d_old)
    S1 = 0
    S2 = - C / dt * (phi_d - phi_d_old)
    values = {'S1': S1, 'S2': S2, 'rate': r}

    return values


def debye_length(target,
                 pore_bjerrum_length='',
                 pore_concentration=''):
    r"""
    A method for calculating the debye length. The debye length is a function
    of concentration and bjerrum length. The debye length gives us information
    about the size of the electrical double layer.

        .. math::
            {lambda}_{d} = 1 / {kappa}
            {kappa}^{2} = 8{pi}{lambda}_{b}c_{mA}N_{av}

    Parameters
    ----------
    pore_bjerrum_length : string
        The dictionary key on the target object containing the bjerrum length,
        {lambda}_{b}

    pore_concentration : string
        The dictionary key on the target object containing the macropore
        concentration, c_{mA}

    """

    Nav = 6.02214076e23

    lamda_B = target[pore_bjerrum_length]
    C = target[pore_concentration]

    k = np.sqrt(8 * np.pi * lamda_B * C * Nav)
    lamda_D = 1/k

    return lamda_D
```

```
def salt_surface_density(target,
                         pore_concentration='',
                         pore_debye_length='',
                         pore_donnan_potential='',
                         pore_temperature=''):
    r"""
    A method for calculating the salt surface density. The rate of change of
    the salt surface desnity is used to determine the uptake rate of salt into
    the electrical double layers of micropores.

        .. math::
            w = 8{lambda}_{d}c_{mA}sinh^{2}(F/4/R/T{\Delta\phi}_{d})

    Parameters
    ----------
    pore_concentration : string
        The dictionary key on the target object containing the macropore
        concentration, c_{mA}

    pore_debye_length : string
        The dictionary key on the target object containing the debye length,
        {lambda}_{d}

    pore_donnan_potential : string
        The dictionary key on the target object containing the donnan
        poential, {\Delta\phi}_{d}

    pore_temperature : string
        The dictionary key on the target object containing the temperature, T

    """

    F = 96485.3321233100184
    R = 8.31446261815324

    CmA = target[pore_concentration]
    phi_d = target[pore_donnan_potential]
    T = target[pore_temperature]
    lamda_D = target[pore_debye_length]

    w = 8 * lamda_D * CmA * np.sinh(1/4 * phi_d * F / R / T) ** 2

    return w


def charge_surface_density(target,
                           pore_donnan_potential,
                           pore_concentration='',
                           pore_debye_length='',
                           pore_temperature=''):
    r"""
    A method for calculating the concentration of charge stored in the
```

```
    micropore. The amount of charge stored in the micropore is used for
    determining the stern potential.

        .. math::
            {sigma} = 4{lambda}_{d}c_{mA}sinh(F/2/R/T{\Delta\phi}_{d})

    Parameters
    ----------
    pore_donnan_potential : array-like
        The pore length array of donnan potential values

    pore_concentration : string
        The dictionary key on the target object containing the macropore
        concentration

    pore_debye_length : string
        The dicionary key on the target object containing the debye length

    pore_temperature : string
        The dicionary key on the target object containing the temperature

    """
    F = 96485.3321233100184
    R = 8.31446261815324

    phi_d = pore_donnan_potential
    CmA = target[pore_concentration]
    T = target[pore_temperature]
    lamda_D = target[pore_debye_length]

    sigma = 4 * lamda_D * CmA * np.sinh(1/2 * phi_d * F / R / T)

    return sigma


def stern_potential(target,
                    pore_charge_density,
                    pore_capacitance=''):
    r"""
    A method for calculating the micropore concentration. The change in
    micropore concentration is used as the source term of our mass algorithm.

        .. math::
            {\Delta\phi}_{st} = c_{charge,mi} * F / Ca

    Parameters
    ----------
    pore_charge_density : array-like
        The pore length array of micropore charge values

    pore_capacitance : string
        The dictionary key on the target object containing the volumetric
        capacitance, Ca
```

```
    """
    F = 96485.3321233100184

    sigma = pore_charge_density
    C = target[pore_capacitance]

    phi_st = sigma * F / C

    return phi_st


def _potential_balance_gcs(phi_d,
                           target,
                           pore_capacitance='',
                           pore_concentration='',
                           pore_debye_length='',
                           pore_potential='',
                           pore_potential_electrode='',
                           pore_temperature=''):
    r"""
    Hidden method used for calculating donnan potential. Fsolve is used to
    find the root of this potential balance. The root is the donnan potential.
    The potential balance across the EDL must be zero.
    """

    phi = target[pore_potential]
    phi_1 = target[pore_potential_electrode]

    sigma = charge_surface_density(target,
                                   phi_d,
                                   pore_concentration,
                                   pore_debye_length,
                                   pore_temperature)

    phi_st = stern_potential(target,
                             sigma,
                             pore_capacitance)

    fun = phi_d + phi_st + phi - phi_1

    return fun


def donnan_potential(target,
                     pore_capacitance='',
                     pore_concentration='',
                     pore_debye_length='',
                     pore_potential='',
                     pore_potential_electrode='',
                     pore_temperature='',
                     solver='fsolve'):
    r"""
    A method for calculating the donnan potential. To calculate the donnan
    potential, we use scipy's fsolve on a charge balance around the electrical
```

```
double layer. Here is a look at the charge balance:

    .. math::
        0 = {\Delta\phi}_{d} + {\Delta\phi}_{st} + {phi} - {phi}_1

Parameters
----------
pore_capacitance : string
    The dictionary key on the target object containing the volumetric
    capacitance, Ca in units F/m3

pore_concentration : string
    The dictionary key on the target object containing the macropore
    concentration, c_{mA}

pore_potential : string
    The dictionary key on the target object containing the macropore
    potential, {phi}

pore_potential_electrode : string
    The dictionary key on the target object containing the electrode
    potential, {phi}_{1} We assume that electrode is fixed and that the
    electronic conductance is much greater than the ionic conductance.

pore_temperature : string
    The dictionary key on the target object containing the temperature, T

solver : string
    The name of scipy's solver. Either fsolve or newton_krylov. The default
    solver is fsolve. Use newton_krylov for large systems.

"""
n = target.project.network.Np

if solver == 'newton_krylov':

    f = _potential_balance_gcs
    phi_d = newton_krylov(lambda phi_d: f(phi_d,
                                          target,
                                          pore_capacitance,
                                          pore_concentration,
                                          pore_debye_length,
                                          pore_potential,
                                          pore_potential_electrode,
                                          pore_temperature),
                          np.zeros(n))

elif solver == 'fsolve' :

    phi_d = fsolve(_potential_balance_gcs, np.zeros(n),
               args=(target,
                     pore_capacitance,
                     pore_concentration,
                     pore_debye_length,
```

```
                              pore_potential,
                              pore_potential_electrode,
                              pore_temperature))

    return phi_d
```

modified Donnan

```
import numpy as np
from scipy.optimize import fsolve
from scipy.optimize import newton_krylov


def donnan_source_mass(target,
                       micropore_volume='',
                       micropore_concentration='',
                       algorithm=''):
    r"""
    Calculates the rate of electrosorption using the modified Donnan model
    for electrical double layers. This model assumes electroneutrality
    in the macropores and electrode symmetry. Therefore, the rate calculated
    is the rate of salt adsorbed not the rate of an individual ion.

        .. math::
            r = - V_{mi} / 2 * dc_{mi}/dt

    Parameters
    ----------
    micropore_volume : string
        The dictionary key on the target object containing the micropore
        volume, V_{mi}

    micropore_concentration : string
        The dictionary key on the target object containing the micropore
        concentration. We automatically append '_old' to find the old value.

    algorithm : string
        The algorithm name of which the time step is stored

    Returns
    -------
    A dictionary containing the following three items:

        **'rate'** - The value of the source term function at the given X.

        **'S1'** - The slope of the source term function at the given X.

        **'S2'** - The intercept of the source term function at the given X.

    The slope and intercept provide a linearized source term equation about the
    current value of X as follow:

        .. math::
            rate = S_{1}   X  +  S_{2}

    """

    project = target.project
    algorithm = project.algorithms()[algorithm]
```

```
    V_mi = target[micropore_volume]
    C_new = target[micropore_concentration]
    C_old = target[micropore_concentration + '_old']
    dt = algorithm.settings['t_step']

    r = - V_mi / 2 / dt * (C_new - C_old)
    S1 = 0
    S2 = - V_mi / 2 / dt * (C_new - C_old)

    values = {'S1': S1, 'S2': S2, 'rate': r}

    return values


def donnan_source_potential(target,
                            pore_capacitance='',
                            pore_donnan_potential='',
                            algorithm=''):
    r"""
    Source term for considering the effects of co-ion expulsion from micropores
    into macropores. For example, a decrease in the donnan potential in the
    anode signifies positive ions (co-ion) leaving the micropore and entering
    macropore resulting in a gernation of positive ions in the ionic transport
    algorithm. This source term is to be applied to an ionic transport
    algorithm.

        .. math::
            r = - C * d{\Delta\phi}_{d}/dt

    Parameters
    ----------
    pore_capacitance : string
        The dictionary key on the target object containing coefficient C, the
        capacitance with units C/V or F

    pore_donnan_potential : string
        The dictionary key on the target object containing the donnan
        potential denoted as {\Delta\phi}_{d}. We automatically append '_old'
        to find the old value.

    algorithm : string
        The algorithm name of which the time step is stored

    Returns
    -------
    A dictionary containing the following three items:

        **'rate'** - The value of the source term function at the given X.

        **'S1'** - The slope of the source term function at the given X.

        **'S2'** - The intercept of the source term function at the given X.
```

```
    The slope and intercept provide a linearized source term equation about the
    current value of X as follow:

        .. math::
            rate = S_{1}   X  +  S_{2}

    """

    project = target.project
    algorithm = project.algorithms()[algorithm]

    C = target[pore_capacitance]
    phi_d_new = target[pore_donnan_potential]
    phi_d_old = target[pore_donnan_potential + '_old']
    dt = algorithm.settings['t_step']

    r = - C / dt * (phi_d_new - phi_d_old)
    S1 = 0
    S2 = - C / dt * (phi_d_new - phi_d_old)
    values = {'S1': S1, 'S2': S2, 'rate': r}

    return values


def micropore_concentration(target,
                            pore_concentration='',
                            pore_donnan_potential='',
                            pore_temperature=''):
    r"""
    A method for calculating the micropore concentration. The change in
    micropore concentration is used as the source term of our mass algorithm.

        .. math::
            c_{mi} = 2c_{mA}cosh(F/R/T{\Delta\phi}_{d})

    Parameters
    ----------
    pore_concentration : string
        The dictionary key on the target object containing the macropore
        concentration

    pore_donnan_potential : string
        The dictionary key on the target object containing the donnan
        potential

    pore_temperature : string
        The dicionary key on the target object containing the temperature

    """
    F = 96485.3321233100184
    R = 8.31446261815324

    CmA = target[pore_concentration]
    phi_d = target[pore_donnan_potential]
```

```
    T = target[pore_temperature]

    Cmi = 2 * CmA * np.cosh(phi_d * F / R / T)

    return Cmi


def micropore_charge(target,
                     pore_donnan_potential,
                     pore_concentration='',
                     pore_temperature=''):
    r"""
    A method for calculating the concentration of charge stored in the
    micropore. The amount of charge stored in the micropore is used for
    determining the stern potential.

        .. math::
            c_{charge,mi} = 2c_{mA}sinh(F/R/T{\Delta\phi}_{d})

    Parameters
    ----------
    pore_donnan_potential : array-like
        The pore length array of donnan potential values

    pore_concentration : string
        The dictionary key on the target object containing the macropore
        concentration

    pore_temperature : string
        The dicionary key on the target object containing the temperature

    """

    F = 96485.3321233100184
    R = 8.31446261815324

    CmA = target[pore_concentration]
    T = target[pore_temperature]
    phi_d = pore_donnan_potential

    C_charge_mi = 2 * CmA * np.sinh(phi_d * F / R / T)

    return C_charge_mi


def stern_potential(target,
                    micropore_charge,
                    pore_capacitance=''):
    r"""
    A method for calculating the micropore concentration. The change in
    micropore concentration is used as the source term of our mass algorithm.

        .. math::
            {\Delta\phi}_{st} = c_{charge,mi} * F / Ca
```

```
    Parameters
    ----------
    micropore_charge : array-like
        The pore length array of micropore charge values

    pore_capacitance : string
        The dictionary key on the target object containing the volumetric
        capacitance, Ca

    """

    F = 96485.3321233100184

    C_charge_mi = micropore_charge
    Ca = target[pore_capacitance]

    phi_st = C_charge_mi * F / Ca

    return phi_st


def _potential_balance(phi_d,
                       target,
                       pore_capacitance='',
                       pore_concentration='',
                       pore_potential='',
                       pore_potential_electrode='',
                       pore_temperature=''):
    r"""
    Hidden method used for calculating donnan potential. Fsolve is used to
    find the root of this potential balance. The root is the donnan potential.
    The potential balance across the EDL must be zero.

    """
    phi = target[pore_potential]
    phi_1 = target[pore_potential_electrode]

    C_charge_mi = micropore_charge(target,
                                   phi_d,
                                   pore_concentration,
                                   pore_temperature)

    phi_st = stern_potential(target,
                             C_charge_mi,
                             pore_capacitance)

    fun = phi_d + phi_st + phi - phi_1

    return fun


def donnan_potential(target,
                     pore_capacitance='',
```

```
                 pore_concentration='',
                 pore_potential='',
                 pore_potential_electrode='',
                 pore_temperature='',
                 solver='fsolve'):
r"""
A method for calculating the donnan potential. To calculate the donnan
potential, we use scipy's fsolve on a charge balance around the electrical
double layer. Here is a look at the charge balance:

    .. math::
        0 = {\Delta\phi}_{d} + {\Delta\phi}_{st} + {phi} - {phi}_1

Parameters
----------
pore_capacitance : string
    The dictionary key on the target object containing the volumetric
    capacitance, Ca in units F/m3

pore_concentration : string
    The dictionary key on the target object containing the macropore
    concentration, c_{mA}

pore_potential : string
    The dictionary key on the target object containing the macropore
    potential, {phi}

pore_potential_electrode : string
    The dictionary key on the target object containing the electrode
    potential, {phi}_{1} We assume that electrode is fixed and that the
    electronic conductance is much greater than the ionic conductance.

pore_temperature : string
    The dictionary key on the target object containing the temperature, T

solver : string
    The name of scipy's solver. Either fsolve or newton_krylov. The default
    solver is fsolve. Use newton_krylov for large systems.

"""
n = target.project.network.Np

if solver == 'newton_krylov':

    f = _potential_balance
    phi_d = newton_krylov(lambda phi_d: f(phi_d,
                                          target,
                                          pore_capacitance,
                                          pore_concentration,
                                          pore_potential,
                                          pore_potential_electrode,
                                          pore_temperature),
                          np.zeros(n))
```

```
elif solver == 'fsolve' :

    phi_d = fsolve(_potential_balance, np.zeros(n),
               args=(target,
                     pore_capacitance,
                     pore_concentration,
                     pore_potential,
                     pore_potential_electrode,
                     pore_temperature))

return phi_d
```