การเพิ่มความเข้มข้นของกรดคาร์บอกซิลิกโดยกระบวนการฟอร์เวิร์ดออสโมซิส

นายธรรม รู้ประกอบกิจ

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรดุษฎีบัณฑิต
สาขาวิชาวิศวกรรมสิ่งแวดล้อม ภาควิชาวิศวกรรมสิ่งแวดล้อม
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2559
ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

CONCENTRATION OF CARBOXYLIC ACIDS BY FORWARD OSMOSIS PROCESS

Mr. Tam Ruprakobkit

A Dissertation Submitted in Partial Fulfillment of the Requirements

for the Degree of Doctor of Philosophy Program in Environmental Engineering

Department of Environmental Engineering

Faculty of Engineering

Chulalongkorn University

Academic Year 2016

Thesis Title   CONCENTRATION OF CARBOXYLIC ACIDS BY
                      FORWARD OSMOSIS PROCESS
By   Mr. Tam Ruprakobkit
Field of Study   Environmental Engineering
Thesis Advisor   Associate Professor Dr. Chavalit Ratanatamskul

Accepted by the Faculty of Engineering, Chulalongkorn University in Partial Fulfillment of the Requirements for the Doctoral Degree

....Dean of the Faculty of Engineering

(Associate Professor Dr. Supot Teachavorasinskun)

THESIS COMMITTEE

....Chairman

(Associate Professor Dr. Sutha Khaodhiar)

....Thesis Advisor

(Associate Professor Dr. Chavalit Ratanatamskul)

....Examiner

(Associate Professor Dr. Pisut Painmanakul)

....Examiner

(Assistant Professor Dr. On-anong Larpparisudthi)

....External Examiner

(Dr. Chalatip Ratasuk)

ธรรม รู้ประกอบกิจ : การเพิ่มความเข้มข้นของกรดคาร์บอกซิลิกโดยกระบวนการฟอร์เวิร์ด
ออสโมซิส (CONCENTRATION OF CARBOXYLIC ACIDS BY FORWARD OSMOSIS
PROCESS) อ.ที่ปรึกษาวิทยานิพนธ์หลัก: รศ. ดร. ชวลิต รัตนธรรมสกุล, 348 หน้า.

เพื่อสร้างความเข้าใจความสัมพันธ์เชิงพฤติกรรมของกระบวนการฟอร์เวิร์ดออสโมซิสกับ
พารามิเตอร์ต่างๆที่เกี่ยวข้องกับกระบวนการในระหว่างการเพิ่มความเข้มข้นของกรดคาร์บอกซิลิค
แบบจำลองทางคณิตศาสตร์จึงถูกพัฒนาขึ้นในงานวิจัยนี้ โดยวิธีการแจกแจงและรวบรวมสมการทาง
คณิตศาสตร์ของตัวแปรต่างๆที่เกี่ยวข้องอย่างเป็นระบบและถี่ถ้วน โดยอาศัยระเบียบวิธีของ
Levenberg-Marquardt แบบจำลองที่ถูกพัฒนาขึ้นจะสามารถกำหนดค่าของตัวแปรตามที่จุดต่างๆ
ของเวลาระหว่างการจำลองสภาวะการทำงานของกระบวนการฟอร์เวิร์ดออสโมซิส (Dynamic
Simulation Model) งานวิจัยนี้เลือกสารละลาย 2 ชนิดคือ NH_4Cl และ $NaCl$ เป็น Draw solution
ในการศึกษาถึงกลไกการกรองกรดคาร์บอกซิลิคในกระบวนการฟอร์เวิร์ดออสโมซิส เพื่อนำไปสู่การ
พัฒนาแบบจำลองทางคณิตศาสตร์ของกระบวนการที่สอดคล้องกับชนิด Draw solution ที่ใช้
แบบจำลองที่ถูกพัฒนาขึ้นนี้จะได้รับการตรวจสอบความแม่นยำกับข้อมูล จากผลทดลองของการเพิ่ม
ความเข้มข้นของกรดคาร์บอกซิลิกโดยใช้กรดคาร์บอกซิลิคสี่ชนิดและสารละลายผสมของกรดคาร์บอก
ซิลิคสองชนิด เป็น Feed solution ในขณะที่เมมเบรนเป็นแบบ Thin-Film Composite จากการ
เปรียบเทียบค่าตัวแปรที่ทำนายจากแบบจำลองกับค่าที่ได้จากผลการทดลองนั้น มีความสอดคล้อง
และใกล้เคียงกันเป็นอย่างดี ในกรณีของ NH_4Cl เป็น Draw solution ค่าการทำนายคลาดเคลื่อน
มาตราฐาน ของค่าน้ำหนักที่เปลี่ยนไปของ Draw solution และค่าพีเอช ของ Feed solution มีค่า
4.20%-6.99%และ 0.52%-1.13% ตามลำดับสำหรับกรดชนิดเดียว และมีค่า 5.88%-9.34% และ
0.68%-0.88% ตามลำดับ สำหรับสารละลายผสมของกรด 2 ชนิด ส่วนในกรณีของ $NaCl$ เป็น Draw
solution ค่าของค่าน้ำหนักที่เปลี่ยนไปของ Draw solution และ ค่าพีเอชของ Feed solution มีค่า
4.76%-6.36% และ 0.48%-0.68% ตามลำดับสำหรับกรดชนิดเดียว และมีค่า 5.31%-5.90% และ
0.66%-1.15% ตามลำดับสำหรับสารละลายผสมของกรด 2 ชนิด

ภาควิชา   วิศวกรรมสิ่งแวดล้อม   ลายมือชื่อนิสิต ....
สาขาวิชา   วิศวกรรมสิ่งแวดล้อม   ลายมือชื่อ อ.ที่ปรึกษาหลัก ....
ปีการศึกษา 2559

5471408921 : MAJOR ENVIRONMENTAL ENGINEERING

KEYWORDS: FORWARD OSMOSIS; CARBOXYLIC ACID; CONCENTRATION; DYNAMIC MODELING; CO2 PERMEATION

TAM RUPRAKOBKIT: CONCENTRATION OF CARBOXYLIC ACIDS BY FORWARD OSMOSIS PROCESS. ADVISOR: ASSOC. PROF. DR. CHAVALIT RATANATAMSKUL, 348 pp.

To profoundly comprehend the forward osmosis performance, related to the process operation parameters during concentrating carboxylic acid, the mathematical model was developed by thoroughly enumerating and integrating each single logical phenomena equation with pertinent variables during the operation of FO process. By means of Levenberg-Marquardt algorithm, the precise dependent process variables as a function of time were simultaneously determined at each point in simulating time (Dynamic Simulation Model). In this research, the two selected draw solution, NH_4Cl and NaCl, were also used to investigate the filtration mechanisms of carboxylic acid in forward osmosis so as to develop consistent mathematical process models. Forward osmosis experimental process was carried out to verify the developed model by using Thin-Film Composite (TFC) FO membrane whereas feed solution was varied types of carboxylic acids and a mixture of them. In case of NH_4Cl draw solution, the standard error of prediction (*SEP*) for weight change of draw solution and pH of feed solution were in the range from 4.20% to 6.99% and 0.52% to 1.13% of single carboxylic acid feed solution and in the range from 5.88% to 9.34% and 0.68% to 0.88% of a mixture of two carboxylic acids, respectively. In case of NaCl draw solution, the standard error of prediction (*SEP*) for weight change of draw solution and pH of feed solution were in the range from 4.76% to 6.36% and 0.48% to 0.68% of single carboxylic acid feed solution and in the range from 5.31% to 5.90% and 0.66% to 1.15% of a mixture of two carboxylic acids, respectively.

Department:	Environmental Engineering	Student's Signature	....
		Advisor's Signature	....
Field of Study:	Environmental Engineering		
Academic Year:	2016		

ACKNOWLEDGEMENTS

CONTENTS

LIST OF FIGURE

Content   Page

LIST OF TABLE

Content	Pages

CHAPTER I

INTRODUCTION

1.1 Introduction

Carboxylic acids are widely used as basis of chemical in many industries, such as food beverages, pharmaceutical and chemical industries. On the other hand, petrochemical and wood pulping industries usually discharge wastewater, polluted with carboxylic acids, especially acetic acid. Acetic acid is an important chemical in many industries, such as synthesis of cellulose acetate, printing, textile, dyeing and also food industries (Acetic acid, 2015). In pharmaceutical production, the carboxylic acid is used in form of lactic acid in order to produce soluble lactates from insoluble active composition. Because of acidity and antiseptic properties of lactic acid, it also can be used in topical preparation and cosmetics. Moreover, in polylactic acid (PLA) production, lactic acid performs as a monomer, also recognized as a raw material of biodegradable plastic (Lactic acid, 2015). To produce several types of butyrate esters, butyric acid is certainly involved. As a result of low molecular weight of butyric acid ester, methyl butyrate attains pleasant aromatic or taste properties. Therefore, it can be utilized as an additive in food and perfume productions (Butyric acid, 2015). Another form of carboxylic acid, which is generally used as a chemical intermediate, is valeric acid. For instance, it can be used as vinyl stabilizer for plasticizers and another benefit of veralic acid is to be odorant constituent in agricultural pesticide and lubricants (Valeric acid, 2015). The acid waste streams discharged from these manufacturing processes generally contain low levels of organic acid such as acetic acid concentration less than 5% from terephthalic acid process and valeric acid concentration in the range of 0.5-2.5 g/l from waste stream of nylon manufacturing (Jung et al., 2008; Rodríguez

et al., 2000) Typically carboxylic acids, which consist of carbon between two and five, are known as fermentation products (acidogenesis) in anaerobic digestion. They are created by acidogenic bacteria as a result of converting of less soluble organic compounds to organic acids. These acids are known as volatile fatty acids (VFAs), which mainly contain acetic acid. Unlike acidogenic bateria, methanogenic bacteria, creates a variety of by-products which are water, methane and carbon dioxide. Balancing the presences of these two types of bacteria, one for acid ferment and another for methanogen, would accomplish the overall anaerobic digestion rate of waste stabilization in the digester system (Yang et al., 2003). However, most of wastewater treatment plants have discharged usable VFA effluent from incomplete anaerobic digestion because the growth rate of methanogens can be maintained only in a narrow pH range and their growth rates are relatively lower than acidogens. Concentration, recovery and reuse these valuable materials from waste streams therefore are certainly more preferable than to treat them and then form undesired sludge for disposal, regarding to environments and economics.

Recently the membrane processes, such as ultra-infiltration (UF), reverse osmosis (RO), electro-dialysis (ED) and nano-filtration (NF), are recognized as effectively successful processes to treat organic waste streams and to retrieve the diluted organic pollutant (Timmer et al., 1994; Vertova et al., 2009). Based on desalination, membrane technology, that becomes more globally recognized and continuously developed, is forward osmosis (FO) or direct osmosis (DO) as a result of the depletion of fossil fuel leading to the escalating of fuel price (McCutcheon et al., 2006; McGinnis and Elimelech, 2008). Unlike the pressure-driven membrane processes, forward osmosis or direct osmosis requires less energy because it operates under low hydraulic pressure. The driving force is developed by differential osmotic pressure between two sides, intrinsically generated by difference of concentration gradient between low

concentration of feed solution side and high concentration of draw solution side. Certainly, this process consumes less energy than pressure-driven membrane processes, such as reverse osmosis. With less energy consumption, if the draw solution can be discharged without any post treatments or easily recovered by less energy processes, overall energy cost of forward osmosis process will be definitely less than pressure-driven membrane processes. The pressure-driven membrane processes, such as reverse osmosis, are typically driven by hydraulic pressure, created from high pressure pump. Using a high hydraulic pressure leads to high tendency of fouling on membrane surface since all unselective substances in feed stream are pushed against the membrane surface. In a different manner, the forward osmosis is driven by natural osmotic pressure produced by the circulated draw solution. This pressure pulls molecules of water but leaves suspended solids or foulants in the concentrated feed solution. Hence the current processes to remove and recover organic contaminants, such as ultra-filtration (UF), reverse osmosis (RO), known as the pressure-driven membrane processes, may possibly be replaced by the forward osmosis membrane recently. As stated above, if the forward osmosis requires no osmotic agent separation or recovery, it will draw more attentions. In some remote irrigated areas, where fresh water is shortage, the forward osmosis process can extract water from feed solution and also produces the diluted fertilizer which can be utilized in irrigation immediately (Phuntsho et al., 2011). Therefore, to simulate this application of forward osmosis process, this research selects one molar of NH_4Cl as draw solution to represent the fertilizer. This advantage can also be applied for concentrated brine, rejected from seawater reverse osmosis. The concentrated brine draw solution after diluting from the forward osmosis process can directly be discharged into the sea. In the aspect of environmental conservation, the diluted brine which has identical concentration to the seawater will also not damage or affect to marine fauna and flora (Latorre, 2005).

Subsequently, the current study also selects one molar of NaCl draw solution to represent the RO concentrated brine of seawater desalination.

Forward osmosis has been tested to reuse human wastewater for portable water in spacecraft (Cath et al., 2005a; Cath et al., 2006). Due to high contaminant rejection and low membrane fouling, the forward osmosis process is superior to the traditional pressure-driven membrane process (Achilli et al., 2009). The treatment of nutrient-rich liquid stream, anaerobic digester centrate from dewatering of digested biomass, has been researched both batch and continuous operations of FO process (Holloway et al., 2007). In food processing, forward osmosis application for concentrating a variety of liquid food shows the potential membrane technology to be applied in food industries (Nayak & Rastogi, 2010; Sant'Anna et al., 2012). Both effects of pH and temperature on forward osmosis membrane flux by using rainwater as the makeup water source for cooling water dilution have been investigated (Latorre, 2005). Forward osmosis desalination is also applied in oil and gas wastewater which has been addressed in the impacts of membrane selection and operating conditions (hydrodynamic) on the performance of FO process (Wang et al., 2014). As a result of their large specific membrane area, the performance of CTA HF membranes has been evaluated with regard to various operating conditions such as draw solution concentration, membrane orientation, cross flow velocity and temperature (Shibuya et al., 2015).

In each application, the process performance which is evaluated by water recovery, permeate water flux and solute rejection are clearly governed by the inherent membrane properties itself, anti-ICP and anti-fouling along with water permeability, solute rejection and the structure parameter of the membrane (Zhao et al., 2012). The another essentially external factor is process operation parameters that definitely affect to system efficiency such as the characteristic of feed and draw

solutions (type, concentration and capacity), the liquid temperature in the process and the flow velocity in the membrane module as well as the module design and the operation time of the process which also play the significant role in the performance of FO process. However, no such a research has specifically described these parameters in aspect of dynamic modeling to express or model their behaviors to the system performance over time. For this reason, the current study is intended to develop dynamic process model as the useful tools that can help to predict how the system efficiency or flux behavior will change over time by varying operating conditions and therefore assist to optimize performance with no experiment time consuming. This study can also implement as a guideline to configure the initial operating parameters of the FO process so that the optimal performance can be obtained by considering both operational and economical aspects as a whole.

The main objective of this research is to develop the dynamic process model, developed by systematically investigating and compiling each single logical phenomenon equation and associated variables during the operation of forward osmosis process, simulated by using low concentration of single carboxylic acid (10 mM of acetic acid, butyric acid, valeric acid and lactic acid) and a mixture of two carboxylic acids as feed solution. 1 M NH_4Cl and 1 M NaCl were prepared as draw solution. By means of Levenberg-Marquardt algorithm, the developed model is able to determine all precise dependent process variables varying over time during simulating the carboxylic acid concentration in forward osmosis process. In order to achieve this goal, the relevant performance membrane constants which are water permeability coefficient, NH_4Cl and NaCl permeability coefficient and structure parameter, have to be determined and evaluated from corresponding water flux and solute flux models, derived from FO mode configuration (AL-facing-FS) (Tiraferri et al., 2013) while acid permeability coefficients are defined from proposed water flux and

acid flux models which are derived from PRO mode configuration (AL-facing-DS). After accomplishment of model development, the model is validated against the well-controlled FO experiment under the same operating conditions where pH of feed solution and weight change of draw solution are employed as experimental data for model validation. The developed model of this research can also forecast the behaviors of other carboxylic acids and thus offers the further profound understanding and advancing knowledge in this area which can be contributed to future application of wastewater reuse for a large-scale FO process in more pragmatic manner.

1.2 Objectives of the study

1. To investigate the concentrating carboxylic acid performance by forward osmosis process.
2. To investigate the effect of chemical properties of carboxylic acids on membrane water flux and carboxylic acid retention.
3. To develop dynamic mathematical models of forward osmosis process during the concentration of carboxylic acids.
4. To validate the mathematical models by using controlled laboratory experiments.

1.3 Scope of the study

1. The experiment uses a synthetic wastewater as feed solution.
2. The variable parameters during operation study are carboxylic acid types, a mixture of two carboxylic acids and draw solution types.
3. Development of mathematical models is used to explain concentrating carboxylic acids by forward osmosis process.

4. This research is performed at controlled temperature at $28 \pm 0.5°C$.

1.4 Expected benefits

1. To know about concentrating performance of carboxylic acids by forward osmosis process.
2. To know about chemical properties of carboxylic acids effect on water flux and carboxylic acid retention during concentration process.
3. To know about the simulation of mathematical models to predict acid concentration performance as a function of time during the concentration of carboxylic acid

CHAPTER II

LITERATURE REVIEW

2.1 Introduction

Since the early age of mankind, osmosis, natural phenomenon, has been used by human being. In early period, for long term preservation, salt could be used to dehydrate foods. This makes most bacteria, fungi or any possibly harmful organisms to briefly inactivate or dry up and then decease due to osmosis. In general, osmosis is described as the net movement of water which transfers across a particular porous membrane. This movement is caused by a difference of osmotic pressure across the membrane. Only water can pass through this particular porous membrane while dissolved substance molecules or ions are rejected. Many applications of the driving force, created by osmotic pressure, will be examined and discussed later.

However, in the area of water treatment, osmosis is typically a less recognizable process than reverse osmosis. For this reason, a concise RO process narrative is given before proceeding with any discussion about osmosis. RO uses hydraulic pressure to oppose, and exceed, the osmotic pressure of an aqueous feed solution to produce purified water (Sourirajan, 1970). RO uses the applied pressure as a driving force to transport mass through the membrane. On the other hand, osmosis used intrinsic osmotic pressure as a driving force. Several publications on the application of RO in the area of both water treatment and wastewater reclamation show in the literature. A small number of publications about water treatment engineering applications, using forward osmosis (FO) or osmosis or direct osmosis (DO) show in the literature. Yet, at bench scale, FO has been utilized to treat wastewater from industries (Anderson & University of Rhode, 1977; Holloway et al., 2007) (Votta et al., 1974) to condense

landfill leachate at both pilot and full-scale (Beaudry & Herron, 1997) and to treat fluid foods in food industry at bench scale (Jiao et al., 2004). Moreover, FO is being assessed to reclaim wastewater for mobile reuse in life aid systems at demonstration scale (Beaudry & Herron, 1997; Cath et al., 2005a; Cath et al., 2006; Cath et al., 2005b) to desalinate seawater (McCutcheon & Elimelech, 2006), and purify water in urgent alleviation situations (Cohen, 2004). In field of material science, recent development has accepted the application of FO to control drug releasing in the body (Bhatt, 2004). Since the 1960s, Pressure-retarded osmosis (PRO), a familiarly associated process, has been examined and evaluated as a possible process for power generation (Aaberg, 2003[Loeb, 2001 #52; Loeb, 2002)] (Aaberg, 2003; Loeb, 2001; Loeb, 2002). PRO utilizes the osmotic pressure variation between seawater or concentrated brine and clean water to pressurize the saline flow, thus transforming the osmotic pressure of seawater into a hydrostatic pressure, used to generate electricity. The major advantages of application of FO are that it runs at no or low hydraulic pressure, has great rejection of broad spectrum of contaminants, and probably has a lesser membrane fouling tendency than processes that use pressure-driven membrane (Holloway et al., 2007). As, in FO process, the only concerned pressure, a few bars, is resulted from flow resistance in the membrane module, the required equipment is very straightforward and the membrane support is not problematic. In addition, FO is also useful in food and pharmaceutical processing due to the ability to concentrate the feed stream in the absence of high pressures and temperatures that may be able to damage the feed solution. FO application can use for medical purpose as well as a result of the ability to facilitate the slow and precise discharge of drugs which have low oral bioavailability because of their limitation on dissolvability or permeability (Bhatt, 2004).

2.2 Osmotic Processes

2.2.1 Classification of osmotic processes

Osmosis is defined as the transport of water across a particular porous membrane. This water is transported from a higher water chemical potential area to a lower water chemical potential area, which is forced by a difference in solute concentrations across the membrane that allows water passage, but virtually disallows solute molecules or ions. If osmotic pressure (π) was applied to the greater concentrated, it would avert water transport through the membrane. Unlike RO, using hydraulic pressure, FO utilizes the differential of osmotic pressure, defined as $\Delta\pi$, as a driving force to transport water across the membrane. The FO processes cause a feed stream to concentrates and extensively concentrated stream, known as the draw solution, to dilute. In PRO, pressure is oppositely applied to the osmotic pressure gradient direction, same as RO, while the net water flux is still in the concentrated draw solution direction, same as FO. Therefore, PRO can be considered as intermediate process of both.

The typical water transport equation for FO, RO and PRO is

$$J_w = A(\sigma\Delta\pi - \Delta P) \tag{2.1}$$

where J_w the water flux, A is the water permeability constant of the membrane, σ is the reflection coefficient, and ΔP is the applied pressure. $\Delta P = 0$ in case of FO, $\Delta P > \Delta\pi$ in case of RO and $\Delta\pi > \Delta P$ in case of PRO. The permeating water flux directions for FO, PRO and RO are shown in Fig. 2.1 In early 1980s, Lee and his peer (Lee et al., 1981) described flux directions and also driving force for these three processes. Fig. 2.2 shows the FO point, PRO region and RO region as well as the flux reversal point.
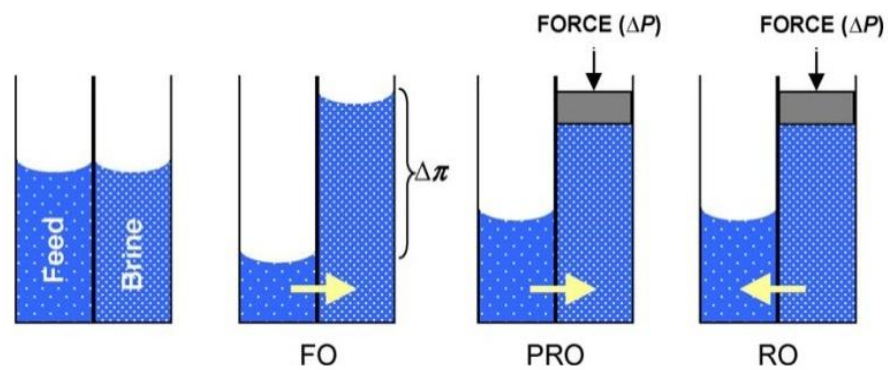
Fig. 2.1 Solvent flow in FO, PRO and RO Solvent flow in FO, PRO and RO

In FO, ΔP is zero and water transportacross the membrane to the more saline side. In PRO, water disperses across the membrane to the more saline side under positive pressure ($\Delta\pi > \Delta P$). In RO, water disperses across membrane to the less side because of hydraulic pressure. (Lee et al., 1981)
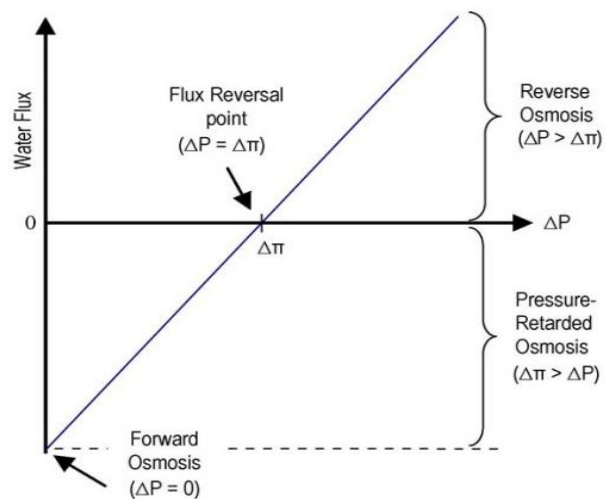
Fig. 2.2 Water flux direction and magnitude, corresponding to applied pressure in FO, PRO and RO.

FO occurs whereas the hydraulic pressure difference is zero. PRO take places whereas the applied hydraulic pressure is between zero and flux reversal point and

RO happens whereas the applied pressure difference is more than the osmotic pressure difference. (Lee et al., 1981)
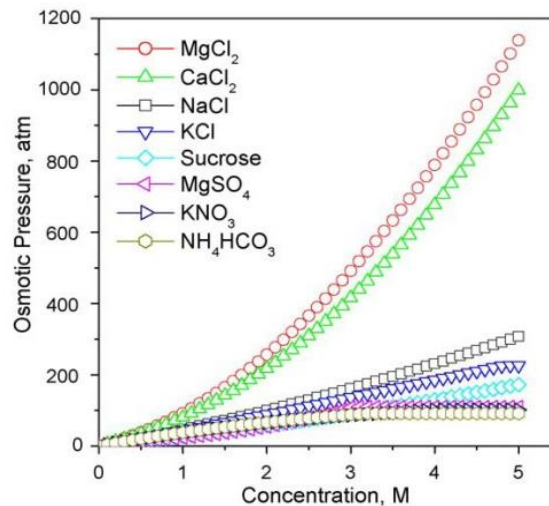
Fig. 2.3 Osmotic pressures correspond to solution concentrations at 25°C for several types of draw solution. Data were computed by OLI Stream Analyzer 2.0 **(OLI Stream Analyzer 2.0, 2005)**

2.3 Draw Solutions

On the permeable side of the membrane, the concentrated solution is the cause of the driving force in the FO processes. In the literature, this solution is named by different terms such as draw solution, sample solution, brine, osmotic agent, osmotic media, osmotic engine or driving solution. This paper selectively uses the term draw solution to name this solution. The key requirement to select the draw solution is the greater osmotic pressure than the feed solution. By utilizing OLI Stream Analyzer 2.0 (OLI Systems Inc., Morris Plains, NJ), the solution osmotic pressures can be determined in order to select as proper draw solution. The osmotic pressures, calculated by this software, as a function of solution concentration are illustrated in Fig. 2.3 To forecast the characteristics of solutions upon a broad range of concentrations and temperatures, this software applies thermodynamic modeling

which is constructed on published experimental data. Selecting a proper process to reconcentrate the draw solution, which has been already diluted after the FO processes, is another critical requirement in some FO application. A sodium chloride solution is usually exploited due to high solubility and ease of reconcentrating to high concentration with RO in the absence of threat of scaling. Furthermore, study of solute diffusion from the draw solution across the membrane is also mandatory. In case of high rejection, which is a requirement in some applications, multivalent ion solution may be a suitable choice. Seawater may be utilized as the draw solution in some applications in PRO. Formerly, seawater (Aaberg, 2003), Dead Sea Water (Loeb, 1998) and Salt Lake Water have been selected or considered as draw solution in several investigations of PRO and FO. Especially, in seawater desalination application, a variety of other chemicals have been considered and examined as solutes for draw solution as well. Batchelder (1965) proposed sulfur dioxide as solute for draw solution for this application. Glew (1965) studied furthermore on this concept and proposed the FO draw solution which consisted of water and another gas (e.g., sulfur dioxide) or liquid (e.g., aliphatic alcohols). Frank (1972) proposed an aluminum sulfate solution, Kravath and Davis (1975) proposed a glucose solution in FO application of seawater desalination, Kessler and Moody (1976) proposed a mixture of glucose and fructose for desalination of seawater and Stache (1989) proposed a concentrated fructose solution to produce a nutritious drink for the course of FO of seawater. McGinnis (2002) proposed a two-stage FO process due to an advantage of temperature-depended solubility of the solutes. Definitely, McGinnis suggested potassium nitrate (KNO_3) and sulfur dioxide (SO_2) solutions to be draw solution for seawater desalination. Later, McGinnis and colleague also suggested the innovative FO application which was shown that mixture of carbon dioxide and ammonia gases in definite ratio generated substantially concentrated draw solution of ammonium salts which were thermally removable. In this process, FO draw solution was created with osmotic pressure over

250 atm. This caused original high recoveries of clean water from concentrated saline feeds and significant decrease of brine ejections from desalination. In a modern nanotechnological method, original non-toxic magnetoferritin (NanoMagnetics, 2005) is being examined as a promising solute for draw solutions. By using a magnetic field, this substance can be promptly isolated form aqueous streams.

2.4 Concentration Polarization in Osmotic Processes

In osmotic–driven membrane processes, the water flux is expressed by Equation (2.1). The osmotic pressure difference through the active layer of the membrane is represented by $\Delta\pi$ in this equation. In these processes, because the osmotic pressure difference through the active layer is relatively considerably low, compared with the bulk osmotic pressure difference, water flux is significantly lower than expected (McCutcheon et al., 2006[Loeb, 1998 #51]) In many membrane-related transport phenomena, such water flux attribute is often noticeable. In osmotic-driven membrane processes, there are two types of concentration polarization (CP) phenomena. Those are external CP and internal CP. Following sections will discuss about both these phenomena.

2.4.1 External concentration polarization

An accumulation of solute at the surface of active layer of membrane is resulted from convective permeate flow in pressure-driven membrane processes. According to concentration polarization (CP), permeate water flux is decreased by this incident as a consequence of enlarged osmotic pressure. To deal with this increased osmotic pressure, hydraulic pressure is necessarily introduced (Sablani et al., 2001; Song & Elimelech, 1995). CP can take place in both pressure-driven membrane processes and osmotic-driven membrane processes, on both the feed and permeates sides of the membrane, because of water permeation. Solutes accumulate at the

membrane's active layer, whereas like RO process, the feed solution permeates across the active layer of the membrane. Similar to CP in pressure–driven processes, this phenomenon can be defined as concentrative external CP. Instantaneously, while contacting with the membrane's permeate side, the draw solution is being diluted whereas the permeate membrane interacts with the permeate water. This phenomenon can be described as dilutive external CP. The effectiveness of osmotic driving force is hindered by both phenomena. By controlling the water flux or enhancing velocity and turbulence at the surface of membrane, this unfavorable effect can be reduced (Mulder, 1997). Nevertheless, in FO, the capability to decrease external CP by reducing flux is constrained due to a ready low water flux. The equations to define external CP phenomena in FO can use those modeled for CP of pressure-driven membranes (Elimelech & Bhattacharjee, 1998).

Compared to the effects in the pressure-driven membrane processes, the effects of membrane fouling caused by external CP on water flux is weaker because FO process uses the low hydraulic pressure. This can be clarified that external CP has a minimal influence in osmotic-driven processes and also has a minor effect to lower the expected water flux in FO process.

2.4.2 Internal concentration polarization

Once an osmotic pressure incline is crated through an entirely rejecting symmetric dense membrane, as shown in Fig. 2.4a, without external CP, the driving force is the difference of osmotic pressures of the bulk solutions. On the other hand, the CP phenomena in FO processes are more complicated as the result of the asymmetric composite membranes consisted of a dense dividing layer and a porous support layer. Couple phenomena can appear in such membranes which depend on the orientation of membrane. Like PRO when an asymmetric membrane's porous support layer confronts the feed solution, whereas solute and water disseminate

across the porous layer, a polarized layer is created next to the dense active layer inside as shown in Fig. 2.4b. This phenomenon likes concentrative internal CP (McCutcheon et al., 2006), but it occurs inside the porous layer, and so, cannot be diminished by cross flow. In desalination and water treatment applications of FO, the feed solution confronts the active layer of the membrane and the draw solution confronts the porous support layer. It is referred to as dilutive internal CP as shown in Fig. 2.4c (McCutcheon et al., 2006), the draw solution inside the porous base turns out to be diluted, while water spreads through the active layer.
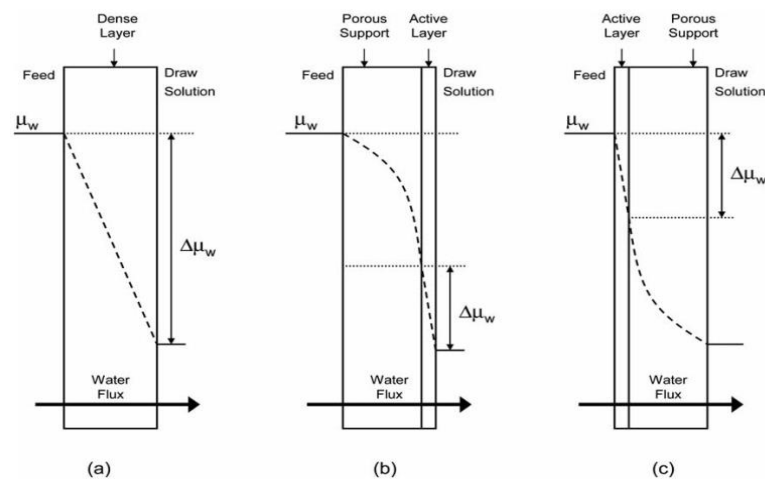
Fig. 2.4 Driving force profiles for osmosis across the membrane types and orientation in term of water chemical potential (μ_w) (McCutcheon et al., 2006) (a) A rejecting symmetric dense membrane. (b) An asymmetric membrane with porous support layer confronting the feed solution; the profile depicts concentrative internal CP. (c) An asymmetric membrane with dense active layer confronting the feed solution; the profile depicts dilutive internal CP. $\Delta\mu_w$ expresses the actual/effective driving force. The effect of external CP is assumed to be insignificant in this diagram.

Fig. 2.5 clearly depicted that the osmotic pressure difference between the bulk feed and bulk draw solution ($\Delta\pi_{bulk}$) is greater than the osmotic pressure through the membrane ($\Delta\pi_m$) because of external CP. It also shows the effective osmotic

pressure-driving force ($\Delta\pi_{eff}$) is smaller because of internal CP. Additionally, like heat exchanger operation, FO operates in a counter-current flow pattern where draw and feed solution flow tangentially to the membrane but in the opposite directions. Such this pattern offers constant $\Delta\pi$ along the membrane module and cause the process to be more efficient (Loeb, 1971).
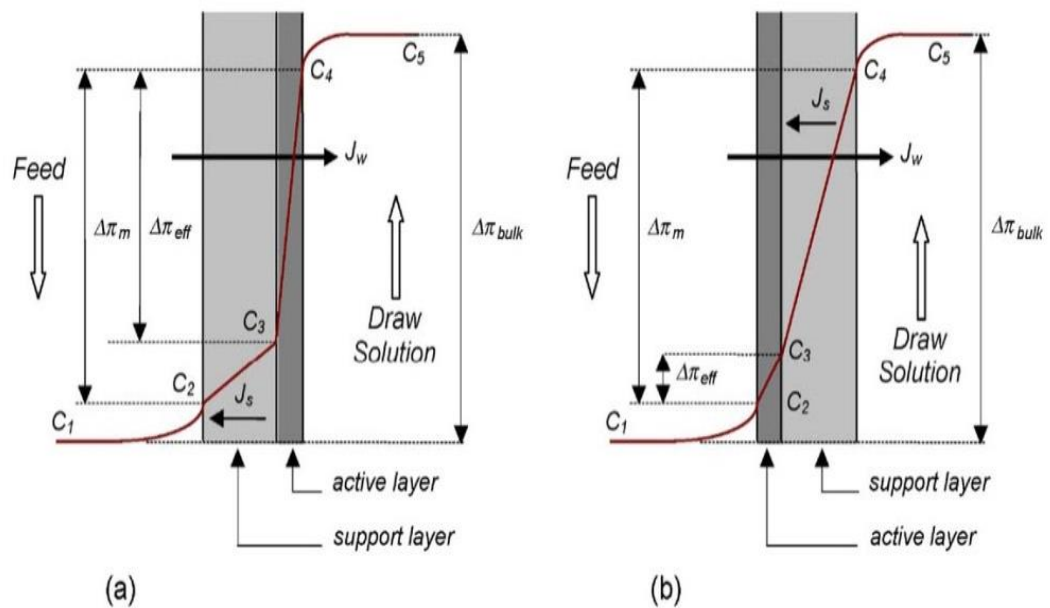
Fig. 2.5 (a) Concentrative internal CP. (b) Dilutive internal CP through a FO composite or asymmetric membrane(a) Concentrative internal CP. (b) Dilutive internal CP through a FO composite or asymmetric membrane.

2.4.3 Modeling concentrative internal concentration polarization

In RO, the hydraulic resistance, caused by the membrane structure, predominantly affects water passage across the membrane. In contrast, in FO, this water passage across the membrane is also affected by internal CP in the porous support layer. Fig.2.5a depicted the profile of concentration through an asymmetric/composite FO membrane for concentrative internal CP, whereas C_1 and C_5 are respectively the concentrations of the bulk feed and draw solutions; C_2 and C_4

are respectively the concentrations of the feed membrane and draw solution membrane interfaces, developing from external CP; and C_3 is the concentration at interface between the active layer and support layer.

Regardless of membrane orientation, a following simplified equation, proposed by Loeb (Loeb, 1998), implemented the models which were formulated by Lee et al. (Lee et al., 1981) express the water flux in FO.

$$J_w = \frac{1}{K} \ln \frac{\pi_{Hi}}{\pi_{Low}} \tag{2.2}$$

where K is the resistance to solute diffusion within the membrane porous support layer; and π_{Hi} and π_{Low} are respectively the osmotic pressures of the bulk feed solution (C_1) and draw solution (C_5). Despite effects of external polarization, K is expressed as

$$K = \frac{t\tau}{\varepsilon D_s} \tag{2.3}$$

$$S = \frac{t\tau}{\varepsilon} \tag{2.4}$$

where S, t, τ and ε are respectively the structure parameter, thickness, tortuosity and porosity of membrane; and D_s is the coefficient of solute diffusion. Nevertheless, Eq. (2.2) has been just proved that it is acceptable only in very low water flux condition (Gray et al., 2006). Later, this equation has been further improved to a more general equation of concentrative internal CP as the following equation:

$$K = (\frac{1}{J_w}) \ln(\frac{B + A\pi_{Hi} - J_w}{B + A\pi_{Low}}) \tag{2.5}$$

where B is the coefficient of the solute permeability of the membrane active layer. This value can be determined by the following equation, obtained by using an RO-type experiment:

$$B = \frac{(1-R)A(\Delta P - \Delta \pi)}{R} \tag{2.6}$$

where R is the salt rejection. The severity of internal CP can be described by Eq. (2.5); the greater values of K, the more severity of internal CP.

2.4.4 Modeling dilutive internal concentration polarization

Fig.2.5b depicted the concentration profile for dilutive internal CP across an asymmetric/composite FO membrane. A simplified general equation for dilutive internal CP in such membranes was suggested by Loeb (Loeb, 1976) as the following equation:

$$K = \left(\frac{1}{J_w}\right) \ln\left(\frac{B + A\pi_{Hi}}{B + J_w + A\pi_{Low}}\right) \tag{2.7}$$

In later investigations (Gray et al., 2006; McCutcheon et al., 2006), during experiments on an FO-designed membrane, this simplified equation has forecasted the results of such experiments as effectively as the equation for concentrative internal CP.

2.4.5 Influence of internal concentration polarization on water flux

Both effects of the porous support layer on internal CP and high draw solution concentrations on the total membrane permeability coefficient were investigated by Mehta and Loeb (1978, 1979). As the results of investigation, by switching the performing fluids on the both sides of the membrane, flux, defined by the permeability coefficient, rapidly decreases because of internal CP as shown in Fig. 2.6.
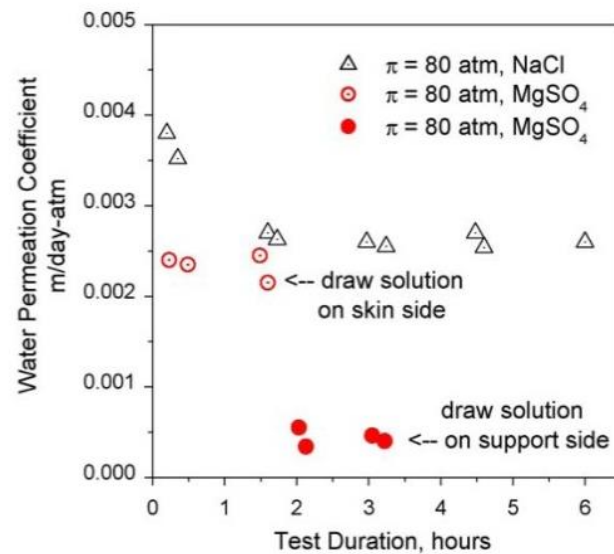
Fig. 2.6 Test of FO by using NaCl and MgSO4 solutions as draw solutions. After 2 hour when the membrane support side is exposed to the draw solution in place of DI water, the effect of internal CP can be noticed. The rate of water permeation rapidly reduces after dilutive internal CP initiates (Mehta & Loeb, 1978)

After the experiments on DuPont B-9 (plane sheet) and B-10 (hollow fiber) Permasep RO membrane, Mehta and Loeb (1979) revealed that the permeability constant of membrane (A, used in Eq. (2.1) is not steady in PRO and FO. Instead, it decreases while osmotic pressure (i.e., concentration) of the draw solution increases. The reduction of the membrane permeability coefficient was caused by the membrane osmotic dehydration or partial drying, which is able to be accompanied by pore concentration, called osmotic deswelling. Thus, such partial draying enlarged resistance to transport of water. Based on recent studies, the internal CP has been verified as the reason of the considerable reduction of flux.

2.4.6 Define solute permeability coefficient (B_s)

Solute permeability coefficient (B_s), a membrane's ability to retain salt, is determined by testing the FO membrane in a cross-flow RO mode with target feed solution. Solute rejection is measured during RO operation with varying of applied

pressure. Solute permeability coefficient (B_s) related to applied pressure is derived from the classical solution-diffusion model as following:
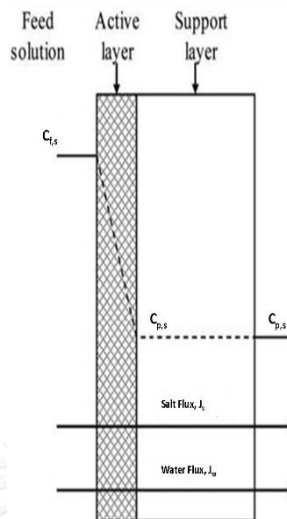
Fig. 2.7 Solute concentration profile in RO operation mode.

Fig.2.7 shows the system, operating in a RO mode. The feed solution is against the active layer and salt flux through FO membrane is expressed as

$$J_s = B_s(C_{f,s} - C_{p,s}) \qquad (2.8)$$

J_s and J_w are related by the salt concentration transported across the selective membrane.

$$C_{p,s} = \frac{J_s}{J_w} \qquad (2.9)$$

Substituting equation 2.9 in equation 2.8 yields an expression for salt flux from feed solution side to permeate side.

$$J_s = \frac{B_s}{1+\frac{B_s}{J_w}} \, C_{f,s} \qquad (2.10)$$

where B_s is the salt permeability coefficient, $C_{f,s}$ is salt concentration in a feed solution and $C_{p,s}$ is leakage salt concentration that transports through a selective membrane. The salt rejection in a cross flow RO mode is define as

$$R_s = 1 - \frac{C_{p,s}}{C_{f,s}}$$ (2.11)

Substituting equation 2.9 in equation 2.11 yields

$$R_s = 1 - \frac{J_s}{J_w C_{f,s}}$$ (2.12)

Combining equation 2.10 and 2.12 express as

$$R_s = 1 - \frac{B_s}{B_s + J_w}$$ (2.13)

Substituting J_w from Equation 2.1 into equation 2.13, thus

$$R_s = \frac{1}{1 + \frac{B_s}{A(\Delta P - \Delta \pi)}}$$ (2.14)

where R_s is solute rejection, ΔP and $\Delta \pi$ are applied pressure and solute osmotic pressure across the membrane, respectively.

2.4.7 Define of structure parameter (S)

The structure parameter is defined by using FO membrane operation in a FO mode. The structure parameter can derive from the classical solution-diffusion theory and the diffusion-convection transport model as following:
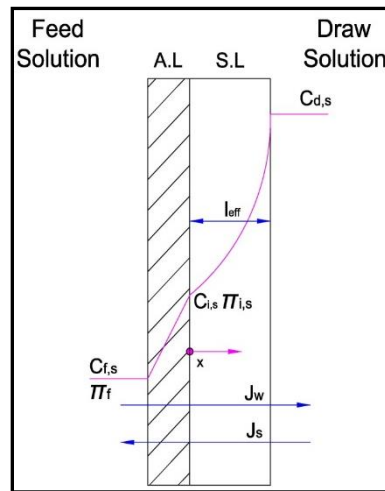
Fig. 2.8 The solute concentration profile, active layer facing feed solution.

Fig. 2.8 depicts the water flux (J_w) direction coming from feed solution side by passing through the FO membrane. The solute flux (J_s) direction comes from draw solution side then the solution-diffusion model are applied to the non-porous rejection layer.

$$J_w = A\left(\pi_{i,s} - \pi_f\right) \tag{2.15}$$

$$J_s = B(C_{i,s} - C_{f,s}) \tag{2.16}$$

where A and B are water permeability coefficient and solute permeability coefficient respectively; π_f and $C_{f,s}$ are osmotic pressure and concentration of feed solution; and $\pi_{i,s}$ and $C_{i,s}$ are osmotic pressure and concentration at the interface of support layer and active layer.

The solute transport in support layer due to the convection $(J_w C)$ and the solute back-transport through the active layer (J_s) can be balanced by using the diffusion convection transport model

$$J_w C + J_s = D_{eff}\frac{dc}{dx} \tag{2.17}$$

where C is the solute concentration in the support layer at a distance X away from the interface between active layer and support layer. D_{eff} is the effective solute

diffusion coefficient in support layer. ϵ is a porosity of support layer and relates to $D_{eff}, D_{eff} = \epsilon D.$ D is the solute diffusion coefficient.

The boundary conditions of equation 2.17 are

$$C = C_{i,s} \qquad \text{at} \qquad X = 0$$

$$C = C_{d,s} \qquad \text{at} \qquad X = l_{eff} = \tau l$$

where l is the real thickness of support layer; l_{eff} is the effective thickness of support layer; and τ is the tortuosity of support layer.

Solving equation 2.15, 2.16 and 2.17 results in:

$$J_w = \frac{S}{D}\ln(\frac{C_{i,s} + B(C_{i,s} - C_{f,s})/A\ (\pi_{i,s} - \pi_f)}{C_{d,s} + B(C_{i,s} - C_{f,s})/A\ (\pi_{i,s} - \pi_f)}) \qquad (2.18)$$

where S is structure parameter, given by $S = \tau l/\epsilon$. The osmotic pressure of the solution is equivalent to its molar concentration. Equation 2.18 can be simplified to:

$$J_w = \frac{S}{D}\ln(\frac{A\pi_d + B}{A\pi_f + J_W + B}) \qquad (2.19)$$

To define the structure parameter in FO testing by using de-ionized water as feed solution $(\pi_f = 0)$, the final equation is

$$S = \frac{D}{J_w}\ln\frac{(B_s + A\pi_d)}{(B_s + J_w)} \qquad (2.20)$$

Where D is solute diffusion coefficient, water flux (J_w) is measured constantly in a FO testing. π_d is a osmotic pressure of draw solution. A is a water permeability coefficient and B_s is a solute permeability coefficient.

2.5 Membrane for Forward Osmosis

To use as a FO membrane, any dense, impermeable or preferably permeable material can be selected. Early, many membranes, as stated, (Kravath and Davis, 1975; Anderson, 1977; Mehta and Loeb, 1978, 1979; Loeb et al., 1997) have been evaluated in flat and capillary configurations for several FO applications. Such period, researchers tested on many types of the membrane materials that were available such as bladders of cattle, fish and pigs, rubber, collodion (nitrocellulose), goldbeaters' skin (Anderson & University of Rhode, 1977) and porcelain. By 1960, the Loeb-Sourirajan process had been developed to create flawless, high flexible, anisotropic RO membranes (Baker, 2004). This breakthrough crucially leaded to switch the membrane applications form laboratorial scale to industrial scale. Loeb et al. (1995, 1997) researched the FO and PRO applications of asymmetric aromatic polyamide membranes.

All related osmosis researches, mostly about PRO, in the 1970s, exploited either flat sheet or tubular RO membranes and demonstrated much lower flux than expected. Votta et al. (1974) and Anderson (1977) utilized various available RO membranes in the market and domestic cellulose acetate membranes for FO applications to treat diluted wastewater by selecting simulated seawater as the draw solution. Kravath and Davis (1975) utilized Eastman's flat sheet RO membranes and Dow's cellulose acetate hollow fiber membranes for FO applications to desalinate seawater by selecting glucose as the draw solution. Goosens and Van-Haute (1978) utilized cellulose acetate membranes, which were strengthen by mineral fillers, to investigate the predictability of membrane performance under RO process through FO experiment. Mehta and Loeb (1978, 1979), in the 1970s, tested with DuPont B-9 flat sheet and B-10 Permasep hollow fiber RO membranes, consisted of an aromatic polyamide polymer. The latter membrane caused relatively low flux (around 8 l/m^2 at 80 bar driving force).

In the 1990s, Osmotek Inc. Albany, Oregon, presently Hydration Technologies Inc. (HTI), developed a special FO membrane, which has been evaluated to use in extensively various applications by several researchers (Beaudry & Herron, 1997; Cath et al., 2005a; Cath et al., 2006; Cath et al., 2005b; McCutcheon et al., 2006; Wrolstad et al., 1993). This membrane was effectively utilized in marketable applications that were emergency relief, water purification for military and recreational purposes. Fig. 2.7 shows a cross-sectional SEM image of the membrane. This exclusive membrane is assumed to be consisted of cellulose triacetate (CTA). The thickness of this CTA FO membrane is smaller than 50 μm and the structure of such membrane is rather different from general RO membranes which usually comprise an extremely thin active layer, smaller than 1 μm, and a thick porous support layer. Unlike RO membrane, CTA FO membrane doesn't have this thick porous support layer. As the result of many researches, the CTA FO membrane, produced by HTI, is the high quality membrane, utilized in FO configuration. The main supporting reasons are seemed to be the comparable thinness of the membrane and the absence of the constructed support layer.
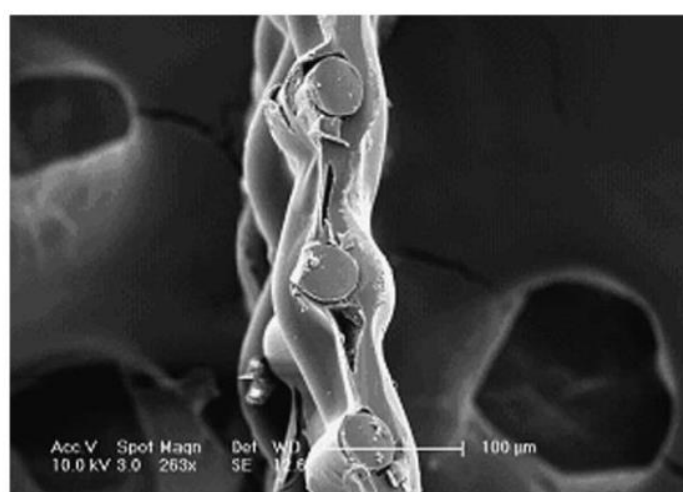
Fig. 2.9 A cross-sectional SEM image of HTI's FO membrane. A cross-sectional SEM image of HTI's FO membrane. Within the polymer material, a polyester mesh is

embedded for mechanical support. The thickness of the membrane is less than 50 μm. (McCutcheon et al., 2005)

Conclusively, the favorable characteristics of FO membranes would be the high density active layer in order to gain the high solute rejection and minimum porosity support layer in order to obtain low internal CP. Those characteristics lead to less membrane fouling, higher water flux, hydrophobicity for improved flux and greater mechanical force to maintain hydraulic pressure while operate in PRO mode. With the crucial advance of semi-permeable FO membranes, the membranes can obtain high flux, salt rejection, minimal internal CP and the mechanical strength to maintain high hydraulic pressures. This critical improvement will direct to more effective performance of existing applications and further development of innovative FO applications.

2.6 Application of Forward Osmosis Processes

2.6.1 Concentration of landfill leachate

As landfill leachate, a highly variable and complex solution, is selected as feed solution, the treatment becomes challenging complicated, especially when a high quality of effluent is required. It comprises organic compound, dissolved heavy metals, organic and nonorganic nitrogen and total dissolved solids (TDS). All constituents are four typical types of pollutants as shown in Table 2.1. Normally landfill leachate is treated in the wastewater treatment facility. However, in such a process, there is frequently no treatment for TDS and sometimes ever increases its concentration (York et al., 1999).

Table 2.1 Water quality characteristics of raw leachate and final effluent (York et al., 1999)

Contaminant (metals or salts)	Untreated leachate (µg/l)	Recovered water from leachate (µg/l)	Rejection (%)	NPDES TMDL(µg/l)
Aluminum	1320	ND	100	
Arsenic	39	ND	100	
Barium	305	1	99.67	
Cadmium	4	0.036	99.1	1.8
Calcium	91600	150	99.84	
Chromium	146	ND	100	16
Copper	18	ND	100	18
Iron	8670	24	99.72	
Lead	12	ND	100	5.3
Magnesium	73700	33	99.96	
Manganese	1480	ND	100	
Nickel	81	ND	100	
Phosphorus	5740	ND	100	
Potassium	56000	543	99.03	
Selenium	ND	ND	100	
Silicon	35500	ND	100	
Silver	7	ND	100	
Sodium	1620000	3990	99.75	
Strontium	2370	ND	100	
Titanium	468	ND	100	
Vanadium	132	ND	100	
Zinc	531	ND	100	120

Table 2.2 (continue) Water quality characteristics of raw leachate and final effluent (York et al., 1999)

Contaminant	Untreated leachate (µg/l)	Recovered water from leachate (µg/l)	Rejection (%)	NPDES TMDL (µg/l)
Alkalinity	5000	4	99.92	
BOD-5	472	2	99.58	45
Chloride-water	1580	23	98.54	378
COD	3190	ND	100	
Fluoride-free	1	ND	100	
N-Ammonia	1110	1.6	99.86	2.8
TKN	780	ND	100	
pH	8	5.6	30	
TDS	2380	48	97.98	
TSS	100	ND	100	
Specific conductance	9940 µS/cm	25 µS/cm	99.75	

ND = not detect

There are two commercial available treatment practices to remove TDS from wastewater, mechanical evaporation and membrane process. Based on thorough evaluation on both practices, FO, one of membrane process, is successfully effective process to treat landfill leachate (York et al., 1999). In 1998, at the Coffin Butte Landfill in Corvallis, where obtains more than 1,400 mm per year of rainfall and produces annually nearly 20,000 – 40,000 m^3 of leachate, Osmotek build a pilot-scale FO system to experiment on the concentration landfill leachate. As the requirement of the National Pollutant Discharge Elimination System (NPDES), the total maximum daily load (TMDL), TDS level of the leachate must be less than 100 mg/l before land application. For 3 months of experiment, Osmotek used CTA membrane (Section 2.5) and NaCl as the draw solution. As the result of the experiment, 94-96% water recoveries were yielded with excessive contaminant rejection and there is no flux reduction during raw leachate process, but 30-50% flux reduction was notices during concentrated leachate process. However, after cleaning, almost overall flux restoration was resulted.
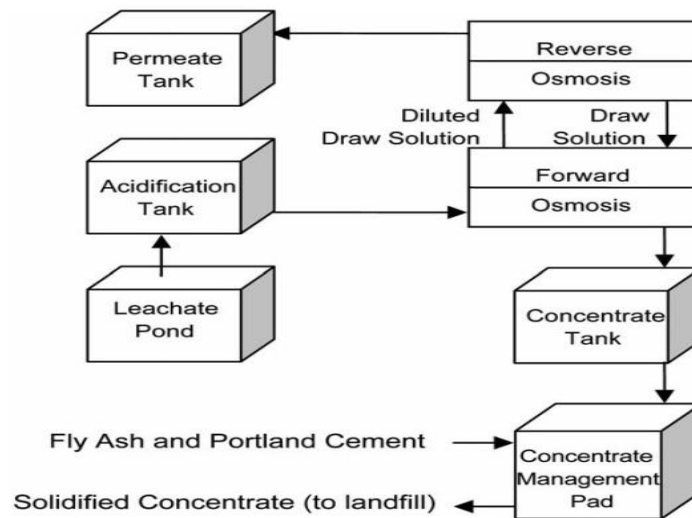
Fig. 2.10 The flow diagram of full-scale FO leachate treatment process. (York et al., 1999)

The successful result of this pilot scale experiment directed to the full scale application, as shown in Fig. 2.8 In the full scale application, prior to six stages of FO cells in water extraction, the raw leachate is accumulated and pretreated. Both a flux of purified water for land application and a reconcentrated flux of draw solution at around 75 g/l NaCl are generated by a three-pass (permeate-staged) RO system and after the solidification, the solidified concentrated leachate is disposed. From June 1998 to March 1999, over 18,500 m^3 of leachate were treated in the treatment plant that accomplished 91.9% average water recovery and 35 µS/cm average RO permeate conductivity (York et al., 1999). Table 2.1 shows the main contaminant concentrations in feed and the full-scale FO system effluent. As shown in such table 2.2, exceeding 99% rejections of most contaminants were obtained and the NPDES TMDL required levels on final effluent concentrations were undoubtedly achieved.

2.6.2 Direct potable reuse for advanced life support systems

In the long-distance space missions, such as mission to Moon or Mars, the self-sufficient and incessant condition of fresh water supply is very critical. This fresh water

supply is depleted for consumptive, hygienic and maintenance purposes. To achieve such a condition, the water treatment system is compulsory in order to retrieve potable water from wastewater, produced on the spacecraft during a mission. Typically, in the long-range space mission, the three major sources of wastewater, which can be treated or reused in water treatment system, are urine, hygienic wastewater and a product of humidity condensation. Due to demanding conditions of space missions, this water treatment system must be dependable, robust, lightweight and highly efficient in team of fresh water recovery rate, compared to input wastewater. Beside those characteristics, the system is supposed to operate independently and require minimal energy consumption, low maintenance and least consumables (Wieland, 1994).

On the basis of the direct osmotic concentration (DOC) system, a pilot scale FO system was built to produce potable water during space missions (Beaudry and Herron, 1997; Beaudry et al., 1999). The U.S. National Aeronautics and Space Administration (NASA) is assessing various technologies and DOC is one of them. The DOC test module of NASA comprises a permeate-staged RO cascade and two subsystems of pretreatments. An FO process is used in the first subsystem (DOC#1) and a distinctive mix of FO and osmotic distillation (OD) is used in the second (DOC#2). The purpose of DO process is to reject miniscule composition, such as urea, which effortlessly propagate across semi-permeable membranes (Beaudry & Herron, 1997; Cath et al., 2005a). Fig. 2.9 depicts a schematic diagram of the original NASA DOC test module.
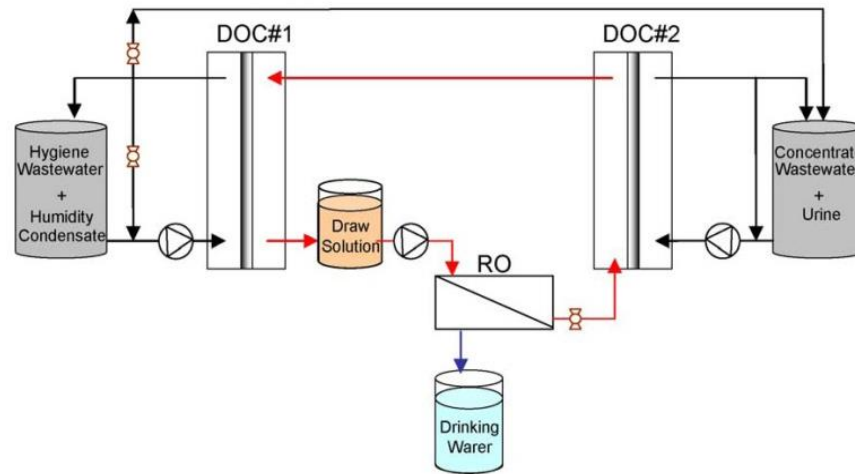
Fig. 2.11 A flow diagram of the original NASA DOC test module. In DOC#1 and DOC#2, three waste streams are pretreated. Then, the draw solution is reconcentrated and, in the RO subsystem, drinking water is generated.

During 1994-1999, the first DOC investigation stage which pertained to design, construction and testing of the fundamental functions of the DOC module was completed by NASA and Osmotek (Beaudry & Herron, 1997) and during 2002-2004, this DOC module was handover to the University of Nevada, Reno to complete the second stage which related to system optimization of functioning environments (Cath et al., 2005a; Cath et al., 2005b). This stage aimed to investigate the long-run performance, including pros and cons of DOC, and to justify a mass metric data to compare the DOC system with other potential advanced alternative recovery processes. The third stage occurred during 2004-2007, where the up-to-date prototype was designed and built for delivery to NASA. In spite of resemblance to the original DOC system, which FO will be utilized as a RO pretreatment, this prototype was distinctive system in company with an isolated urine treatment process (Cath et al., 2005a).

Table 2.3 Water flux across one FO membrane and four RO membranes in FO mode (Cath et al., 2005)

Membrane used	Water flux (l/m^2h)
CTA (Osmotek)	17.4
Osmonics (CE)	1.9
Osmonics (CD)	1.97
Hydranautics (LFC1)	0.54
Hydranautics (LFC3)	0.66

*Remark 100 g/l NaCl solution was used as the draw solution and DI water as the feed.

The second stage of this investigation focused on two issues that were consumption of energy and performance of membranes. Probably because of the unique structure of the CTA membrane, utilized in this investigation, led to the lower internal CP in the membrane, the CTA outclassed available RO membrane in the market, as shown in Table 2.3 The energy consumption was gauged under variable working conditions. The parameters which were discovered to be crucial for this process operation were various. Under these variable working conditions, the particular energy consumption for every 1 m^3 of effluent of recovered water is approximately always less than 30 kWh, illustrated in Fig. 2.10. Besides the process optimization is currently studied.
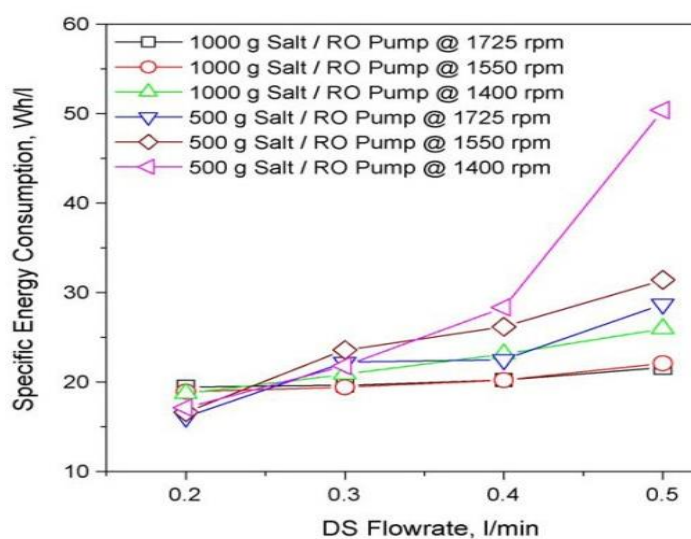
Fig. 2.12 Specific energy consumption corresponds to draw solution flow rate, salt loading, and RO pump capacity for the DOC system. (Cath et al., 2005a)

2.6.3 Forward osmosis for source water purification-hydration bags

Section 2.6.2 concisely presented the FO application for water purification. The purpose of hydration bags is also to be used for recreation, military and emergency life support system where potable water is rare or not accessible (HTI, 2005). Not many FO applications are marketable and hydration bags are one of them. Compared to other water purification implements, the FO hydration-bags are slower, but there are no energy consumption and minimal fouling when even feed by muddy water. By meticulously selecting FO membrane, hydration-bags are confidently workable in most conditions and for most water sources, moreover, and able to produce water without microorganism, most ions and most macromolecules.

The edible draw solution, such as sugar or beverage powder, is contained in a closed semi-permeable FO membrane bag. Once immersed into the source solution, the bag is diffused into by water, because of the difference of osmotic pressures, and then the originally solid draw solution is gradually diluted. The output of the process is the consumable diluted draw solution with minerals and nutrients, containing the bag. The hydration-bags characterize a fundamental treatment process, not a pretreatment process. Fig. 2.11 illustrates the personal hydration-bag, whose process may take 3-4 hour to complete and produces a 12 oz. beverage. The hydration-bags can be immersed directly in the source water or in any container where the source water is held so that it offers better flexibility with self-governing. The hydration-bags were recently world-wildly acquired by the military for emergency and relief uses (Cohen, 2004). However, some experts argue that hydration-bags don't provide water

treatment since it can be only utilized for particular applications and the output of process is a sweet drink, not fresh water.
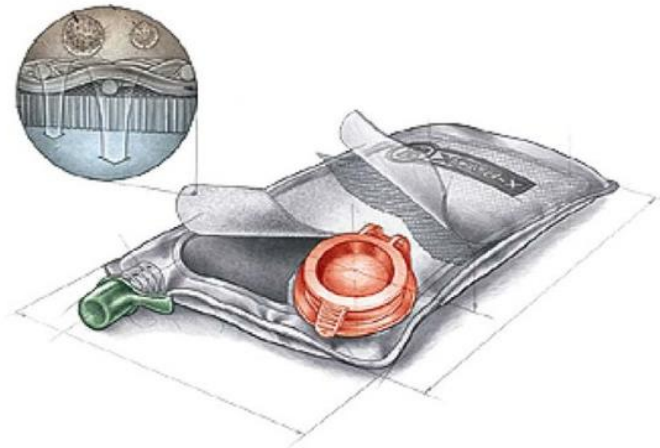
Fig. 2.13 Figure of water purification hydration bag (HTI, 2005)

2.6.4 Osmotic power – Pressure-Retarded Osmosis

In any circumstance in which two streams of different chemical potentials, including saline solutions, interact, the restorable energy can be obtained. For instance, the interaction between the salinity of seawater, producing roughly 2.7 MPa of osmotic pressure, and river water, yielding relatively insignificant osmotic pressure, can generate energy due to a large 2.7 MPa of osmotic pressure difference. To attain such energy, PRO, one of many processes which can be used for this purpose, is selected, as described in Section 2.2.1 and depicted in Fig. 2.1 and 2.2.
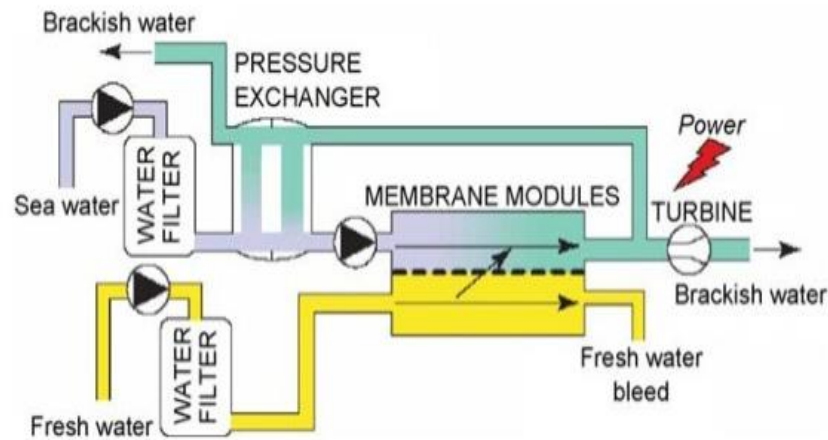
Fig. 2. 14 Simplified process diagram of a PRO power generation plant (Aaberg, 2003)

Fig. 2.12 depicts the schematic diagram of PRO energy generation. In the PRO module, fresh water is pumped into membrane module, consisted of membranes, theoretically, same as the FO semi-permeable membranes as previously described. As the fresh water flows into one side of the membrane modules and disperses across the membrane into another side of the membrane, which is pressurized, the pressurized seawater is diluted and then separated into two streams. One flows to the pressure exchanger, which facilitates the pressurizing process for the incoming water and other flows to a turbine to be depressurized in order to generate the power. Both the pressure exchanger and the membrane are two crucial components in PRO system. Referred to suggestion of Loeb et al (Loeb, 1998; Loeb, 1992) two methods, to pressurize the entering draw solution on the permeable receiving side of the membrane, possibly provide the stable flow system. Those are draw solution pump pressurization, as shown in Fig. 2.12, and draw solution permeate pressurization which utilizes no less than two containers, which are able to resist the pressure on the permeable receiving side.

Due to the characteristics of the energy, produced by PRO, which are large unutilized resource, restorability, least environmental impact and relatively high

density (i.e., power capacity per physical size), compared to other potential oceanic sources (Loeb, 1998). From estimation, the world-wide salinity gradient power production potential is approximately 2,000 TWh per year (Aaberg, 2003).

Loeb and colleagues (Loeb, 1998; Loeb, 2001; Loeb, 2002) has research on the PRO process for power generation for the last many decades. In addition, the power was generated by PRO pilot testing, experimented by Jellinek and Masuda (Jellinek & Masuda, 1981) and the comparison of power generation by PRO and reverse-electrodialysis form salinity gradient sources was conducted by Wick (Wick, 1978). At this time, the comprehensive research on PRO to generate energy is being subsidized by the European Union and being carried on by collaboration of several parties, which are Statkraft SF of Norway, ICTPOL of Portugal, SINTEF of Norway, GKSS Forschungszentrum of Germany and Helsinki University of Technology of Finland. The core objective of this research is to develop PRO membranes for energy generation that is capable to produce the energy, no less than 4 W/m^2 or equivalent (Aaberg, 2003). Both PRO for energy generation and FO for water treatment and desalination rely on the similar semi-permeable membranes, currently not for a commercial purpose, so the further development and accomplishment of PRO for energy generation will favorably affect the development and success of FO for water treatment and desalination as well.

2.7 Literature Reviews

McCutcheon et al., (2006) studied about osmosis through asymmetric membrane in desalination e by means of forward osmosis and power generation known as pressure retarded osmosis (PRO). The primary obstacle in osmosis processes by using asymmetric membrane is severe internal concentration polarization which reduces the osmotic driving force. They explored the effect of both concentrative and dilutive internal concentration polarization on transmembrane flux by using a

commercially available FO membrane. The coupling effect of internal and external concentration polarization is also studied. A developed water flux model to describe the presence of both internal and external concentration polarization involving feed and draw solution for two membrane orientations is present. The mathematic model is validated by experiment data obtained from well controlled laboratory-scale experiments in both membrane orientations. Additionally, the model is used to describe water flux improving performance after changing in membrane structure or system condition.

(Holloway et al., 2007) In wastewater treatment plant, the influent, a raw wastewater, normally mingle with the high nutrient liquid stream created by the dewatering process in digested biomass such as centrate. It results in the rise of operating cost due to increasing of nitrogen and phosphorous load on biological processes. Sometimes it causes the effluent has built-up nutrient concentration. Bench scale testing on forward osmosis (FO) was conducted to verify the practicability to concentrate centrate in batch and continuous operating environment. In continuous bench scale testing, forward osmosis was utilized as pretreatment for reverse osmosis (RO). The testing outcome confirmed that the high-water flux and high nutrient rejection were attainable. With the blended processes between forward osmosis and reverse osmosis, flux could sustain over extensive period of time. To find the specific energy, power and required area of membrane for a greater scale of this process, the mathematic model was acquired. From model, the system should be run at roughly 70 % water recovery in order to optimize power and required area of membrane.

(Cornelissen et al., 2008) Under the development of osmotic membrane bioreactor (OMBR), applying forward osmosis technology, grounded on the osmotic pressure difference, the FO membrane must perform at satisfactory high level, whereas the membrane fouling and draw solution leakage must be at satisfactory low level as

well with the aim of achieving both technical and economical purposes. To optimize FO membrane, applying de-ionized water, investigation of the effect of temperature, membrane type, membrane orientation, type and concentration of draw solution was executed. By utilizing an activated sludge solution, generated from a membrane bioreactor (MBR), investigation of both membrane fouling and draw solution leakage was carried out at laboratory scale experiment.

As the results of the investigation, a FO-type membrane (J_w = 6.2 l/m^{2}h at 20±2 ℃ with 0.5 M NaCl, whereas π = 24 bar) yielded the best FO membrane performance with an activated sludge solution. As the flux of the FO membrane were non-linearly corresponded to the concentration of the draw solution, monovalent ions (NaCl and NaNO$_3$) salts created draw solution which had a better quality than bivalent ions (MgSO$_4$, ZnSO$_4$) did. The deterring of internal concentration polarization on the FO performance resulted in the thickness and structure of the porous substructure of the sample membrane. No fouling was explored during investigation for both reversible and irreversible membrane. In addition, draw solution leakage for FO membrane were insignificant at several draw solution concentrations for various draw solutions.

(Garcia-Castello et al., 2009) In many food productions, concentrating process is generally used for both dewatering of the valuable products and concentrating of effluent wastewater effluent. Due to immense energy consumption, thermal and pressure-driven dewatering method is avoidably expensive. The alternative, like osmotic membrane method, such as forward osmosis, may be a feasible and sustainable solution. By applying NaCl as the substitute draw solution, forward osmosis process outclasses the pressure-driven membrane process, such as reverse osmosis, by ability to concentrate sucrose. For instance, forward osmosis can provide a concentration factor at 5.7 with an initial sucrose concentration at 0.29 M, while reverse osmosis can obtain only up to 2.5. This substantial higher concentration factors in

association with internal concentration polarization affect the lower water fluxes. However, a present prevailing problem of forward osmosis is the utilization of anisotropic polymeric membrane technology. Thus, the enhanced membrane technology can lead to the greater water fluxes and concentration factors.

(Tang et al., 2010) Applications of forward osmosis (FO) in water and wastewater treatment and desalination are being interested growingly. However, besides the major disadvantage of FO, internal concentration polarization (ICP), hindering the efficiency of the FO flux, membrane fouling also can undesirably influence FO membrane flux. In current study, investigation on the effects of ICP and fouling on flux was methodically performed. Based on theoretical model and experiment results, the FO flux was significantly non-linear regarding the apparent driving force, the concentration difference between the draw solution and feed water, due to ICP. Because of the exponential dependence of ICP on flux level, ICP extremely prevailed on FO flux behavior at higher draw solution concentration and/or higher membrane fluxes. While the membrane active layer faced the feed water (AL-facing-FW) by means of dilutive ICP in the FO support layer, ICP was more critical by comparing with the active layer facing draw solution (AL-facing-DS) configuration. Noticeably, against both dilution of bulk draw solution and membrane fouling, impressive flux stability was gained by the AL-facing-FW configuration because a reduction of ICP level counteracted the any attempt to decrease membrane flux. The remaining was only a minimal level of flux reduction. Furthermore, this configuration also yielded a minor foulant deposition. For this reason, the AL-facing-FW configuration intrinsically retained stable flux at the cost of severer initial ICP. On the other hand, the AL-facing-DS configuration adversely gained severe flux reduction while porous membrane support exposed to the humic acid containing feed water. Both the internal congestion of the FO support structure and the consequential greater ICP in the support layer were the reasons of the flux

shortfall in this configuration. The greater ICP was resulted in the reductions of porosity and mass transfer coefficient of the support. At the high level of flux, FO flux reduction was possibly mainly caused by the pore congestion which boosted ICP mechanism. As author stated, this investigation was the first study which systematically revealed the couple effects of both ICP and fouling on the FO flux behavior as author stated.

Nayak & Rastogi, (2010) explored the forward osmosis process in the concentration of anthocyanin extract from Garcinia indica Choisy, normally known as kokum, evaluated against thermally concentrated sample. When feed has, the different molecular compound causes the different characteristics of water flow from feed to osmotic agent side through forward osmosis. These characteristics were studied, including the effects of membrane orientation, osmotic agent concentration, flow rate of feed and osmotic agent and temperature on transmembrane flux amidst the concentration of anthocyanin extract. The concentration of anthocyanin extract was varied from 49 mg/l to 2.69 g/l (54-fold increase) in large scale experiment. Compared with the concentration generated by forward osmosis, non-enzymatic browning index for thermally concentration sample was uncovered about two (0.78-0.35) times, also degradation constant about eight (63.0×10^{-3} to 8.0×10^{-3} day^{-1}) times. During experiment, ratios of HCA lactone to HCA of sample produced by thermal concentration and concentrate produced by forward osmosis were founded at 2.84:1 and 1.50, respectively. From the results of the experiment, the advantages of the concentration of anthocyanin extract created by forward osmosis over the thermal concentration were obviously shown as greater stability, lesser browning index and less ratio of HCA lactone to HCA.

Phillip et al., (2010) developed mathematical model for describes the reverse permeation flux of draw solute across an asymmetric membrane from the draw solution into feed solution in a forward osmosis operation. The effective forward

osmosis system requires that solute reverse flux to be minimized. Forward osmosis experiments were set up to validate the mathematical model predictions with a highly soluble draw solution salt (NaCl) and a cellulose acetate FO membrane. Strong agreement between the model predictions and experiment result was observed by using independently determined membrane transport coefficients. Further analysis indicates that the ratio of forward water flux to reverse solute flux, defined as reverse flux selectivity, is a key design parameter in osmotically driven membrane processes. The both model predictions and experiment result show that this parameter is independent of draw solution concentration and structure of membrane support layer. The reverse flux selectivity value is determined only by the selectivity of membrane active layer.

Garcia-Castello and McCutcheon (2011) studied an alternative for dewatering orange peel press liquor by a forward osmosis that have used in this experiment. A Dewatering process, generated by osmosis into concentrated draw solutions (comprised of sodium chloride at 2 M and 4 M of concentration) across a polymetric cellulose acetate membrane, causes the press liquor to be concentrated. Synthetic press liquor used these draw solutions for concentration process. At a 4 M NaCl draw solution and the synthetic press liquor, concentration factors greater than 3.7 gained. The fouling behavior was noticed, during the experiment, whose mechanism was systematically clarified by pinpointing the crucial elements of the press liquor which mostly cause the fouling. Although Calcium typically is the key element of organic fouling and omnipresent to press liquor, it had a slightly influence on fouling activity in this experiment because of perplexity with citric acid. By decreasing 50% of infiltrate flux, Pectin was the prevailing element providing the fouling. Thus, in order to improve dewatering process, Pectin should be eliminated before proceeding.

Zhang et al., (2011) developed a novel osmotic microbial fuel cell (OsMFC) by using a forward osmosis membrane as a separator. The draw solution, NaCl solution and artificial seawater as catholyte, is used to define the performance of OsMFC. Compare to a conventional MFC with a cation exchange membrane was also operated in parallel. The result indicated that the OsMFC produced more electricity than the MFC in both batch operation using NaCl solution and continuous operation using seawater. Because of a better proton transport or water flux pass through the FO membrane from anode into cathode was observed in the OsMFC system but not in the MFC. While the solute concentration of catholyte effect on both electricity generation and transmembrane flux. This result show a proving concept that the OsMFC can accomplish wastewater treatment, water extraction from wastewater and electricity generation. The potential OsMFC applications are proposed (linking to microbial desalination cells for further wastewater treatment and desalination).

Jin et al., (2011) studied the contaminant transportation and reject by forward osmosis processes in permeate from feed solution into draw solution through the semi-permeable membrane. This is a first time, in the investigation, rejection of contaminants was defined in forward osmosis processes. The processes have significant technical implications by the way of separating clean water from diluted draw solution. In this study a model was developed to predicted boron flux in forward osmosis operation. The experiment result and model have strong agreement which indicated that the model developed in this study can accurately predict the boron transport through forward osmosis membrane. Furthermore, they can guide an improvement of forward osmosis membrane fabrication with decreased boron permeability and structure parameter to minimize boron flux. Both theoretical model and experiment results indicated that when membrane active layer was against draw solution, boron

flux was significantly greater compared to the other membrane orientation, causing by more severe internal concentration polarization.

Zhu et al., (2012) investigated which performed to explore the practicability to use forward osmosis to the simultaneous thickening, digestion, and direct dewatering of waste activated sludge. The overall reduction efficiencies of both the simultaneous thickening and digestion, as operation run for 19 days, in aspects of mixed liquor suspended solids (MLSS) and mixed liquor volatile suspended solids (MLVSS) were nearly 63.7% and 80 % in a row as the MLVSS/MLSS ratio constantly diminished from 80.8% to 67.2%. A superior thickening efficiency was verified by the increasing of MLSS concentration from 7 g/L at the beginning to 39 m/L. To exercise forward osmosis for sludge dewatering, initial sludge depth and draw solution (DS) concentration were specified. At 3 mm sludge depth, roughly 35% of dry sludge content can be obtained in around 60 min, suitable for further applications, whereas the workability of applying the seawater reverse osmosis as draw solution was supported by the current study.

CHAPTER III

METHODOLOGY

3.1 Experimental Scope

All experiment scope is shown by Fig. 3.1

Experimental preparations

1. Bench scale FO apparatus preparation

2. Equipment setup

3. Synthetic feed and draw solution

Experimental steps

1. Take the experiment following experimental plan

2. Collecting the experimental data

3.Comparing to developed mathematical model
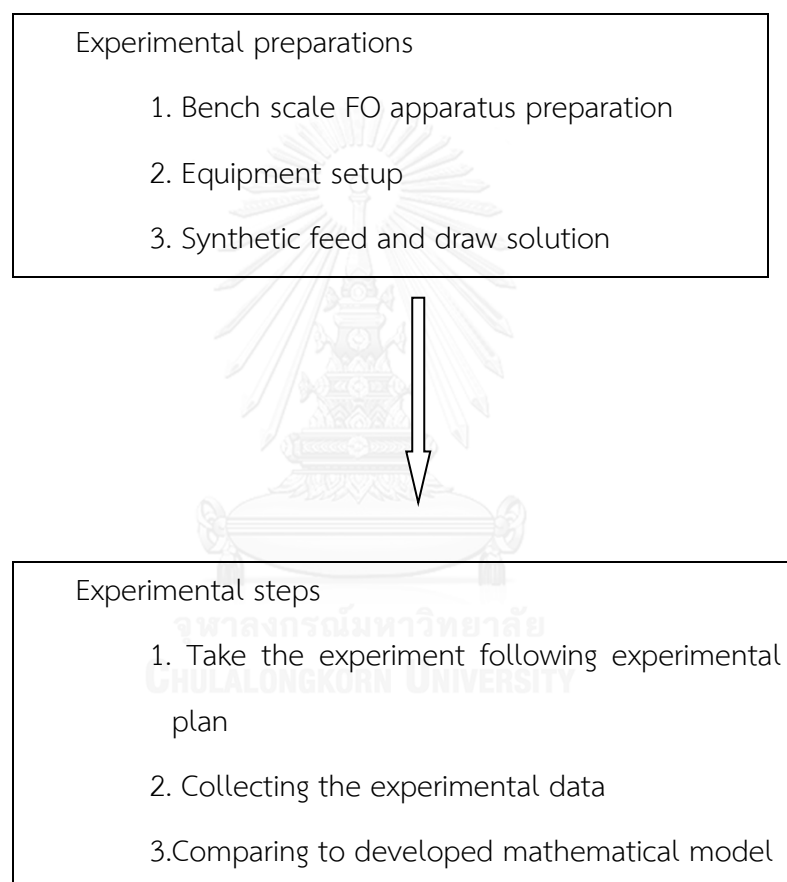
Fig. 3.1 Experimental diagram

3.2 Experimental Chemicals and Instruments

3.2.1 Chemicals

All chemical substances used in this research are reagent grade and supply from Merck Company. All solution is prepared by 0.1 microsiemens/cm de-ionized water from pure water treatment system.

1. Sodium Chloride 99.99 Suprapur® NaCl, Merck Co., Ltd.

2. Ammonium Chloride (99%) NH_4Cl, Merck Co., Ltd.

3. Butyric Acid $C_4H_8O_2$, Merck Co., Ltd.

4. Acetic Acid (30%) $C_2H_4O_2$, Merck Co., Ltd.

5. Valeric Acid $C_5H_{10}O_2$, Merck Co., Ltd.

6. Lactic Acid (90%) $C_3H_6O_3$, Merck Co., Ltd.

3.2.2 Experiment Instruments and Equipment

1. pH meter: HANNA HI 4222

2. Analytical Balances: JADEVER SCALES JIK-8CAB

3. Peristaltic Pump: Shenchen BT600N

4. High Pressure Pump: SKY WATER RO-50

5. Personal Computer: SONY

6. FO Test Cell: STELITECH

7. TFC FO Membrane: Hydration Technologies, Inc.

3.3 Experimental Variables

Experimental Variables are classified as Fixed Variables, Independent Variables, and Dependent Variables. All experimental variables are shown as following;

3.3.1   Fixed Variables

Fixed Variables are the constant parameters in the experiment, there are

- Flow rates of feed and draw solution
- Surface area of a membrane.
- Operating temperature
- Sizing of membrane cell channel

3.3.2 Independent Variables

Independent Variables are parameters that can be changed to know their effects on system performance, there are

- Carboxylic acid types

(acetic acid, butyric acid, valeric acid and lactic acid)

- Draw solution types

(NaCl and NH$_4$Cl)

- Initial carboxylic acid concentration and volume
- Initial draw solution concentration and volume

3.3.3 Dependent Variables

Dependent Variables are parameters that are changed resulted from changing in independent variables, there are

- Physical and chemical properties of carboxylic acid

  feed solution

- Physical and chemical properties of salt draw solution
- Water flux
- Salt flux
- Acid flux

3.4 Forward osmosis process experimental procedures

3.4.1 Feed solution

The FO experiments use each four-single carboxylic acid as feed solution. The chemical properties of four carboxylic acids are shown in Table 3.1.

Table 3.1 pKa and molar mass of carboxylic acids

Carboxylic acid	Formula	Molar mass(g/mol)	pKa
Acetic Acid	$C_2H_4O_2$	60.05	4.75
Lactic Acid	$C_3H_6O_3$	90.08	3.86
Butyric Acid	$C_4H_8O_2$	88.11	4.82
Valeric Acid	$C_5H_{10}O_2$	102.13	4.82

In table 3.1, comparing acetic acid with valeric acid, valeric acid has a significant molar mass greater than acetic acid but they have close pKa value. Therefore, the mixed feed solution of acetic and valeric acid is the selected one that will be used to the simulation of FO experiment. In contrast, lactic acid and butyric acid have significantly different pKa value but they are closely molar mass. Consequently, in this study, the mixed solution of lactic acid and butyric acid will be used to simulate competitive retention of these two carboxylic acids by FO process.

3.4.2 Draw solution

The osmotic agent or draw solution is the source of the osmotic driving force in FO process. As previously stated, two draw solution types, NaCl 1 M and NH_4Cl 1 M, are used to operate in FO process in this study.

3.5 Membrane testing

3.5.1 Determination of water permeability coefficient (A), salt permeability coefficient (B_s) and structure parameter (S)

To characterize the pure water permeability (A), salt permeability coefficient (B_s) and structure parameter (S) of the FO membrane, the testing would be defined by FO methodology. The transport and structure parameter (A, B_s, S) received from RO+FO protocol significantly different from FO protocol (Tiraferri et al., 2013). The fundamental principal in the FO process, the permeate water driving force is created by the natural different osmotic pressure across membrane which differs from RO process that utilizes hydraulic pressure, exerting on the membrane active layer. Thus, the intrinsic transport and structure properties evaluation of a FO membrane characterized by means of FO protocol represent more plausible values. In this research, the transport parameters were evaluated from the following models (Tiraferri et al., 2013):

$$J_w = A \left[\frac{\pi_D e^{\left(-\frac{J_w S}{D}\right)} - \pi_F e^{\left(\frac{J_w}{k}\right)}}{1 + \frac{B}{J_w}\left(e^{\left(\frac{J_w}{k}\right)} - e^{\left(-\frac{J_w S}{D}\right)}\right)} \right] \tag{3.1}$$

$$J_s = B \left[\frac{c_D e^{\left(-\frac{J_w S}{D}\right)} c_F e^{\left(\frac{J_w}{k}\right)}}{1 + \frac{B}{J_w}\left(e^{\left(\frac{J_w}{k}\right)} - e^{\left(-\frac{J_w S}{D}\right)}\right)} \right] \tag{3.2}$$

where k is the feed solute mass transfer coefficient. D is the bulk diffusion coefficient of the draw salt. A is the intrinsic permeability coefficient of water. B is the intrinsic permeability coefficient of salt. Last of all, S is the support layer structure parameter as a function of the thickness of support layer (t_s), tortuosity (τ) and porosity (ε), which is defined as $\frac{t_s \tau}{\varepsilon}$.
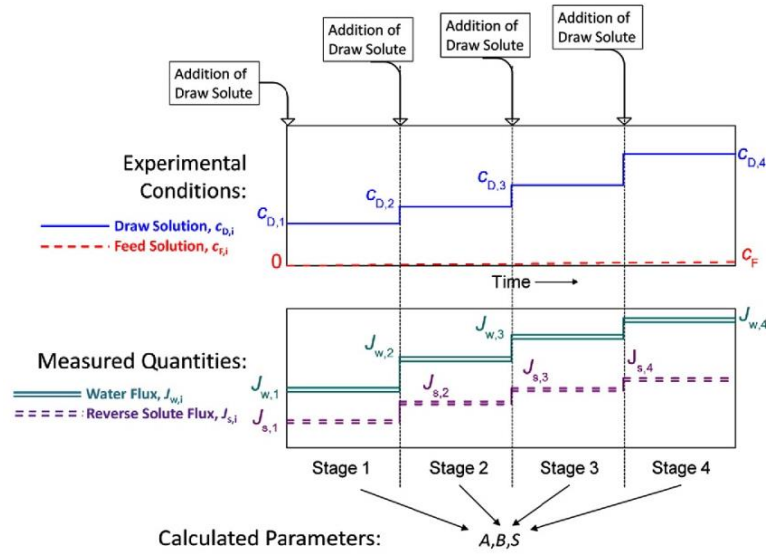
Fig. 3.2 Four stages FO experiment, draw solution concentration, C_D, and feed solution concentration, C_F, are plotted as single lines of the top diagram. Experimental water flux, J_w, and experimental solute flux, J_s, are showed as double line in the bottom diagram

According to Fig.(3.2), in the four stage experiments, $J_{w,i}$ and $J_{s,i}$ will be measured, where $i = 1, 2, 3, 4$. As a result, the eight equations have been acquired with only three unknowns $(A, B$ and $S)$. To fit Eq.(3.1) and Eq.(3.2) with the current experiment, least-squares non-linear regression is applied by defining A, B and S as regression parameters as well as $C_D (\pi_D)$ and $C_F (\pi_F)$ as independent variables, gained by experiment. D and k are known parameters which are the bulk diffusion coefficient of the draw salt and the feed solute mass transfer coefficient respectively. Pertaining to error of fitting, the global error can be determined by the following equation:

$$E_{global} = E_w + E_s = \sum_{i=1}^{n}(\sum_{i=1}^{n}(\frac{J_{w,i}^{Exp}-J_{w,i}^{Cal}}{\bar{J}_w^{Exp,n}}) + \sum_{i=1}^{n}(\frac{J_{s,i}^{Exp}-J_{s,i}^{Cal}}{\bar{J}_s^{Exp,n}})$$

(3.3)

where the superscript Cal and Exp are assigned to the calculated and experimental values respectively. The minimum global error in Eq.(3.3) can be accomplished, where

are $J_{w,i}^{Exp} = J_{w,i}^{Cal}$ and $J_{s,i}^{Exp} = J_{s,i}^{Cal}$ Thus, Eq.(3.4) and Eq.(3.5) can be reformed to these the following equations:

$$J_{w,i}^{Cal} = A \left[\frac{\pi_{D,i} e^{\left(-\frac{J_{w,i}^{Cal}S}{D}\right)} - \pi_{F,i} e^{\left(\frac{J_w^{Cal}}{k}\right)}}{1 + \frac{B}{J_{w,i}^{Cal}} \left(e^{\left(\frac{J_{w,i}^{Cal}}{k}\right)} - e^{\left(-\frac{J_{w,i}^{Cal}S}{D}\right)} \right)} \right] \quad (i = 1, 2, 3, 4)$$

(3.4)

$$J_{s,i}^{Cal} = B \left[\frac{\pi_{D,i} e^{\left(-\frac{J_{w,i}^{Cal}S}{D}\right)} - \pi_{F,i} e^{\left(\frac{J_w^{Cal}}{k}\right)}}{1 + \frac{B}{J_{w,i}^{Cal}} \left(e^{\left(\frac{J_{w,i}^{Cal}}{k}\right)} - e^{\left(-\frac{J_{w,i}^{Cal}S}{D}\right)} \right)} \right] \quad (i = 1, 2, 3, 4)$$

(3.5)

Initially, three estimated sets of regression parameters $(A, B$ and $S)$, applying as lower, higher and the best-estimated value, were mandatory for the calculating process. The solutions were calculated by reiterating process which the acceptable solutions can be obtained where the lowest E_{global} was reached on the subject of the initial estimates of regression parameters.

The coefficients of determination which were represented the goodness of both final solution fits, can be calculated by the following equation:

$$R_W^2 = 1 - \frac{SSE_W}{SST_W} = 1 - \frac{\sum_{i=1}^{n} (J_{w,i}^{Exp} - J_{w,i}^{Cal})^2}{\sum_{i=1}^{n} (J_{w,i}^{Exp} - \bar{J}_w^{Exp})^2}$$

(3.6)

$$R_S^2 = 1 - \frac{SSE_S}{SST_S} = 1 - \frac{\sum_{i=1}^{n} (J_{S,i}^{Exp} - J_{S,i}^{Cal})^2}{\sum_{i=1}^{n} (J_{S,i}^{Exp} - \bar{J}_S^{Exp})^2}$$

(3.7)

A, B_s and S have been solved by downloadable excel spreadsheet at http://dx.doi.org/10.1016/j.memsci.2013.05.023, supplementary materials of the online

version, by inputting these experiment data into the excel algorithm (Tiraferri et al., 2013).

3.5.2 Determination of acid permeability coefficient (B_a)

Pertaining to acid concentration, the experiment operating in FO mode was configured by arranging dense selective active layer of a FO membrane against acid solution (feed solution), and the porous support layer against salt solution (draw solution). Otherwise, the internal clogging of organic acid within support layer is prone to reduce support layer porosity that promotes undesired ICP in the support layer (Tang et al., 2010). PRO experiment therefore was conducted in order to determine the acid permeability coefficient of a FO membrane by setting dense active layer against concentrated acid solution (draw solution) and using de-ionized water as feed solution. The acid permeability coefficient (B_a) was determined from both governing equations of water flux in Eq. (3.8) and acid flux in Eq. (3.9). The derivation of these two equations, adapted from the developed J_w model and J_s model from Eq. (3.8) and Eq. (3.9) respectively, were delineated in Appendix A:

$$J_w = A\left[\frac{\pi_{d,a}e^{\left(-\frac{J_w}{k_d}\right)} - \pi_{f,a}e^{\left(\frac{J_w S}{D_a}\right)}}{1 - \frac{B_a}{J_w}\left(e^{\left(-\frac{J_w}{k_d}\right)} - e^{\left(\frac{J_w S}{D_a}\right)}\right)}\right] \tag{3.8}$$

$$J_a = B_a\left[\frac{C_{d,a}e^{\left(-\frac{J_w}{k_d}\right)} - C_{f,a}e^{\left(\frac{J_w S}{D_a}\right)}}{1 - \frac{B_a}{J_w}\left(e^{\left(-\frac{J_w}{k_d}\right)} - e^{\left(\frac{J_w S}{D_a}\right)}\right)}\right] \tag{3.9}$$

where D_a is apparent weak acid diffusion coefficient, $\pi_{d,a}$ and $\pi_{f,a}$ are osmotic pressure of acid draw solution and acid feed solution respectively, $C_{d,a}$ and $C_{f,a}$ are the acid concentration in draw solution tank and in feed solution tank respectively.

The acid concentration in feed solution was measured by the calibrated conductivity meter (HANNA Instruments, USA). Each experiment used the membrane pieces which were manufactured in the same production batch, so their values of structure parameters (S) and water permeability (A) were identical. Both values were determined from mathematical equations of Eq.(3.1), Eq.(3.2) and the results of experiment which was set deionized water as feed solution and sodium chloride or ammonium chloride as draw solution. Since the number of conditional equations was more than the number of unknown (S, A and B_s), the minimal global error techniques was applied to acquire A, B_s and S values. The yielded values of S and A can directly represented as S and A values in both Eq.(3.8) and Eq.(3.9). Consequently, B_a value was a single unknown in both Eq.(3.8) and Eq.(3.9). By setting experiment with two different initial acid concentrations ($C_{d,a}$) for each membrane sample, two values of B_a were yielded by applying minimal global error principle. The averaged value of B_a was determined to reasonably represent the acid permeability coefficient of the membrane for further calculation.

3.6 Carboxylic acid concentration by FO experiment

The well-controlled experiment of FO laboratory scale was run to validate mathematical model. The FO test cell unit had a rectangular channel sized 9.207 width X 4.572 length X 0.23 depth (cm) on both sides of the installed membrane coupon, the effective membrane area of which was 42 cm^2. Feed draw solutions were pumped into the FO test cell unit which set as the counter-current mode of closed loop system, by using two variable speed peristaltic pumps (Shenchen Precision Pump, China) at flow rate of 1.58 l/min (25 cm/s), gauged by two flow meters. Membrane orientation was in FO mode and experiments were carried out using four carboxylic acid types (acetic acid, lactic acid, butyric acid and valeric acid), a mixture of acetic acid and valeric acid or a mixture of lactic acid and butyric acid as feed solution while draw solution used one molar of sodium chloride or one molar of ammonium chloride. An

initial draw solution volume was 0.5 liter and initial feed solution volume was 1 liter. The temperature of both solutions was maintained at $28 \pm 0.5°C$ by air conditioning. Water permeate volume from feed to draw solution was detected by measuring the online weight change of draw solution with analytical balance (Jadever Scales, UK) and the real time pH of feed solution was measured by on-line pH meter (HANNA Instruments, USA). All observed data were recorded in personal computer.
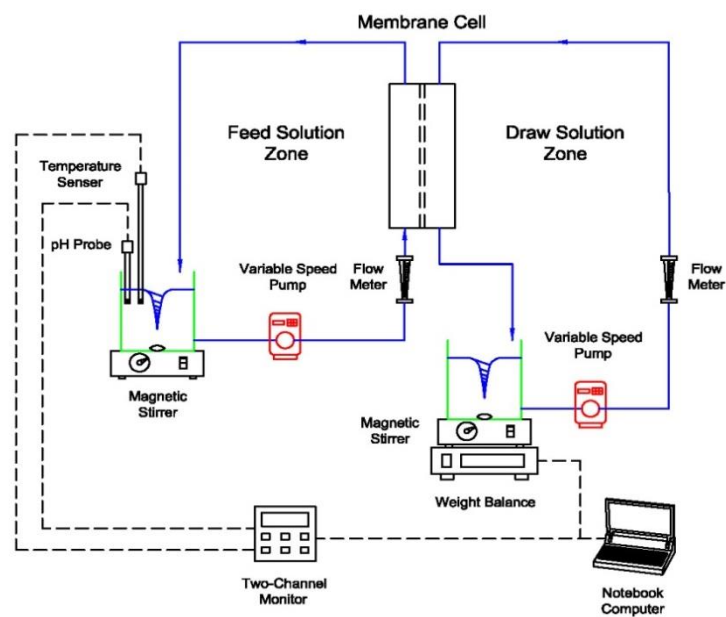
Fig. 3.3 Schematic of FO bench scale process

3.7 Experimental plan

As shown in Fig. 3.4 experimental plan is divided into two parts. The first part is to define the all membrane constants which were inputted into the developed mathematical models. In the second part, the acid concentration experiments of forward osmosis processes, the acquired experimental data are used to validate developed mathematical models.
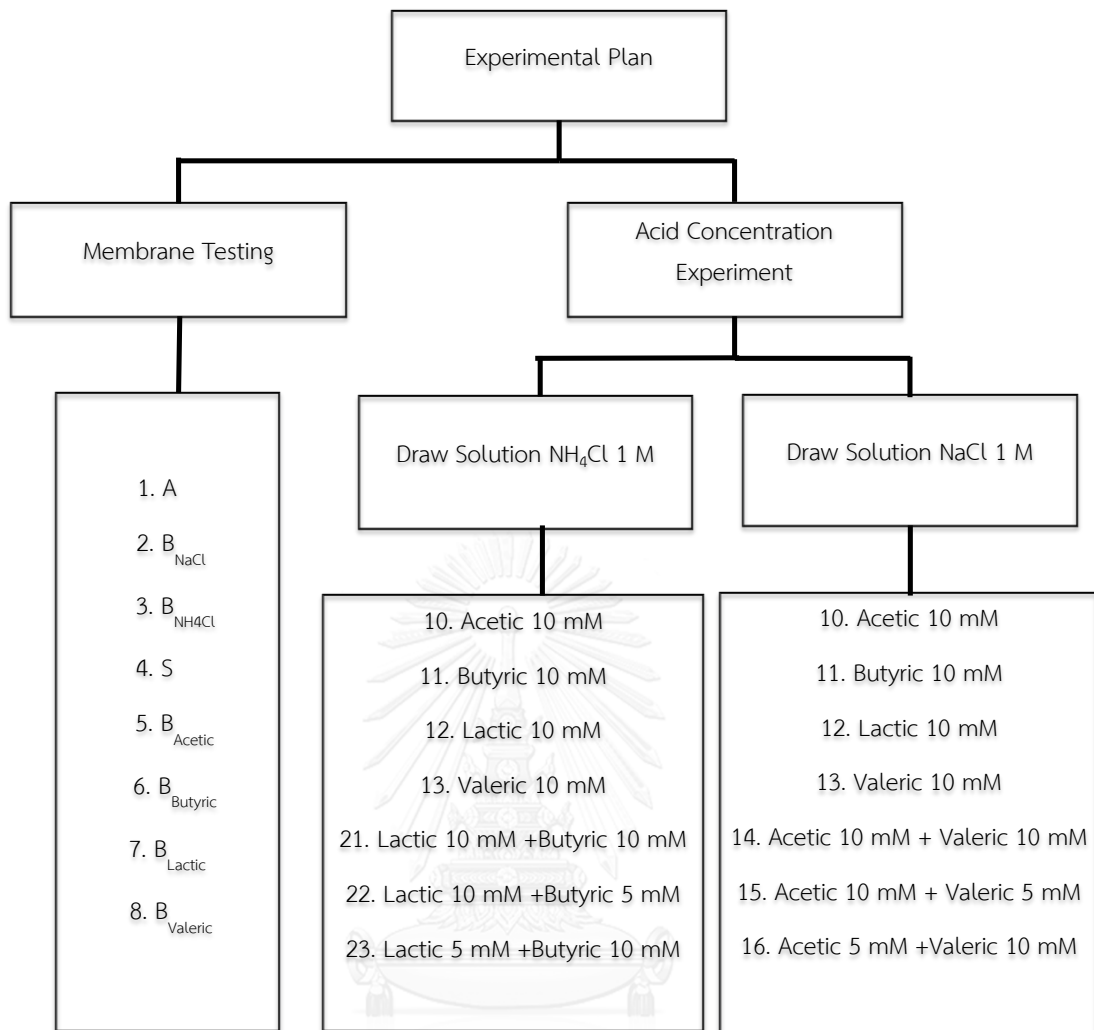
Fig. 3.4 Experimental plan

Table 3.2 Summarized of all experimental runs

No.	Description	FO operation	PRO operation	Feed solution	Draw solution
1.	Define A, B_{NaCl}, S	√	×	De-ionized water	NaCl 1 M
2.	Define A, B_{NH4Cl}, S	√	×	De-ionized water	NH_4Cl 1 M
4.	Define B_{Acetic}	×	√	De-ionized water	Acetic acid
5.	Define B_{Lactic}	×	√	De-ionized water	Lactic acid
6.	Define $B_{Butyric}$	×	√	De-ionized water	Butyric acid
7.	Define $B_{Valeric}$	×	√	De-ionized water	Valeric acid
10.	Acid Concentration	√	×	Acetic acid 10 mM	NaCl 1 M
11.	Acid Concentration	√	×	Lactic acid 10 mM	NaCl 1 M
12.	Acid Concentration	√	×	Butyric acid 10 mM	NaCl 1 M
13.	Acid Concentration	√	×	Valeric acid 10 mM	NaCl 1 M
14.	Acid Concentration	√	×	Acetic 10 mM & Valeric 10mM	NaCl 1 M
15.	Acid Concentration	√	×	Acetic 10 mM & Valeric 5 mM	NaCl 1 M
16.	Acid Concentration	√	×	Acetic 5 mM & Valeric 10 mM	NaCl 1 M
17.	Acid Concentration	√	×	Acetic acid 10 mM	NH_4Cl 1 M
18.	Acid Concentration	√	×	Lactic acid 10 mM	NH_4Cl 1 M
19.	Acid Concentration	√	×	Butyric acid 10 mM	NH_4Cl 1 M
20.	Acid Concentration	√	×	Valeric acid 10 mM	NH_4Cl 1 M
21.	Acid Concentration	√	×	Lactic 10 mM & Butyric 10mM	NH_4Cl 1 M
22.	Acid Concentration	√	×	Lactic 10 mM & Butyric 5 mM	NH_4Cl 1 M
23.	Acid Concentration	√	×	Lactic 5 mM & Butyric 10 mM	NH_4Cl 1 M

3.8 Prediction of FO process performance by developing mathematical models

The acquired performance parameters of membrane testing are used to accomplish developed mathematical models. An integration of involved equations for developing mathematical models in concentrating carboxylic acid by FO process is as following:

1. The water flux model

2. The salt flux model

3. The acid flux model

4. The internal concentration polarization model

5. The external concentration polarization model

6. The osmotic pressure of membrane interface model in a mixture of leakage acid ion and salt ion at a membrane interface of active layer and support layer

7. The osmotic pressure of membrane surface model at feed solution side

8. The mole balance model of acid and salt ion in membrane test cell, both draw and feed solution stock tank

3.9 Model verification

The predicted data, simulated from mathematical model, were evaluated by coefficient of determination (R^2), root mean square error $(RMSE)$ and standard error of prediction (SEP), according to the the following equations:

$$R^2 = 1 - \frac{\sum_{j=1}^{n}(X_{j,pred}-X_{j,exp})^2}{\sum_{j=1}^{n}(X_{j,pred}-\bar{X}_{j,exp})^2} \qquad (3.10)$$

$$RMSE = \sqrt{[\frac{\sum_{j=1}^{n}(X_{j,exp}-X_{j,pred})^2}{n}} \qquad (3.11)$$

$$SEP = \frac{RMSE}{\bar{X}_{j,exp}} \times 100 \qquad (3.12)$$

where $X_{j,pred}$ is predicted data from mathematical simulation model, $X_{j,exp}$ is experimental data, $\bar{X}_{j,exp}$ is average value of experimental data and n is the total number of data. R^2 should be close to 1, $RMSE$ and SEP should be as less as possible.

CHAPTER IV

RESULTS AND DISCUSSIONS

4.1 Single carboxylic acid as feed solution and NH_4Cl as draw solution

4.1.1 FO process modeling of a single carboxylic acid

Fig. 4.1 illustrates carboxylic acid and salt concentration gradients across a semi-permeable membrane and depicts water transport, acid flux and salt flux operating in FO mode at any moment during filtration. Osmosis, as natural physics phenomena, generates osmotic pressure, driving the water molecules through a semi-permeable membrane from diluted feed to highly concentrated draw solution. Consequently, the feed solution becomes more concentrated while the draw solution is diluted over the period of time.
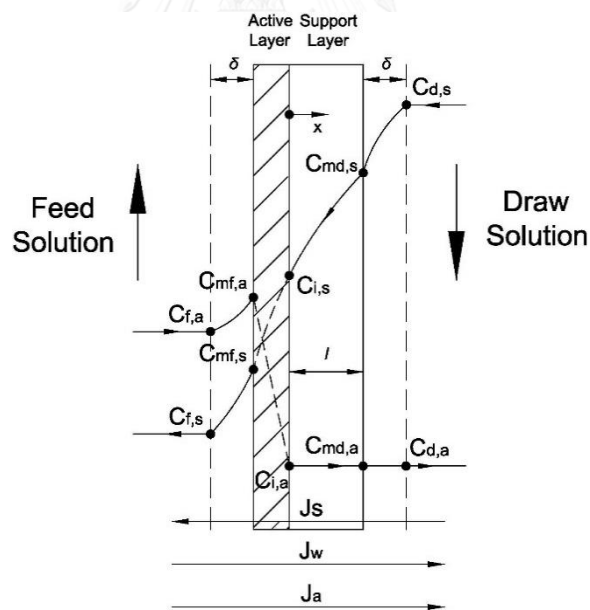
Fig. 4. 1 Schematic of water transport, acid and salt concentration gradients across a semi-permeable membrane for acid concentration by FO filtration process

.

4.1.1.1 Permeate water permeability

Referred to classical solution-diffusion model, the water flux, J_w, across the active layer in a Fig. 4.1 relies on the different osmotic pressure and is expressed as

$$J_w = \sigma A\left(\pi_i - \pi_{mf}\right) \qquad (4.1.1)$$

where A is the water permeability coefficient of the membrane, π_i is the osmotic pressure at interface of support layer-active layer, π_{mf} is the osmotic pressure at the active layer surface and the reflection coefficient, σ, is equal to 1.

4.1.1.2 Reverse salt permeability

The reverse salt flux through active layer, J_s, is proportional to salt concentration gradient across the membrane active layer and expressed as

$$J_s = B_s(C_{i,s} - C_{mf,s}) \qquad (4.1.2)$$

where B_s is the salt permeability coefficient of the membrane, $C_{i,s}$ is the salt concentration at support layer-active layer interface and $C_{mf,s}$ is the salt concentration at active layer surface.

4.1.1.3 Acid permeability

The acid flux, J_a, induced from acid solution side, depends on its concentration gradient across the membrane active layer. It can be expressed as

$$J_a = B_a(C_{mf,a} - C_{i,a}) \qquad (4.1.3)$$

where B_a is the acid permeability coefficient of the membrane, $C_{mf,a}$ is the acid concentration at active layer surface and $C_{i,a}$ is the acid concentration at support layer-active layer interface.

The acid concentration in bulk draw solution, $C_{d,a}$, is given as

$$C_{d,a} = C_{i,a} = C_{md,a} \tag{4.1.4}$$

$C_{d,a}$ equals to $C_{i,a}$ as well as $C_{md,a}$, where $C_{md,a}$ is the acid concentration at support layer surface. Since the acid species diffuse through the active layer and then pass into the support and boundary layers in the same direction of permeate water, acid species thus do not experience the internal and external concentration polarizations likewise the reverse osmosis membrane (Jin et al., 2011). J_a and J_w are related to the acid concentration in bulk draw solution by

$$C_{d,a} = J_a/J_w \tag{4.1.5}$$

      4.1.1.4 Osmotic pressure at the support layer-active layer joint

      π_i, the osmotic pressure at support layer-active layer interface, is the sum of osmotic pressure of salt and weak acid species at the support layer-active layer joint. At equilibrium, weak acids partially dissociate to acid ions and hydrogen ions. π_i is expressed as

$$\pi_i = \emptyset_s n_s C_{i,s} RT + \emptyset_a (C_{i,a} + C_{i,H^+}) RT \tag{4.1.6}$$

where $\emptyset_s$ is the osmotic coefficient of salt solutions, n_s is the number of salt ion species, R is the gas constant, T is the absolute temperature, $\emptyset_a$ is the osmotic coefficient of acid solutions and assumed to be equal to 1. C_{i,H^+}, taking ionic strength into account, is the hydrogen ion concentration at support layer-active layer interface at equilibrium. It can be expressed as

$$C_{i,H^+} = \sqrt{\frac{K_{a,T}(C_{i,a} - C_{i,H^+})}{\gamma_{i,H^+} \gamma_{i,A^-}}} \tag{4.1.7}$$

where $K_{a,T}$ is the ionization constant, γ_{i,H^+} is hydrogen ion activity coefficient at support layer-active layer interface and γ_{i,A^-} is acid ion activity coefficient at support

layer-active layer interface. γ_{i,H^+} and γ_{i,A^-} can be calculated from the Davies equation which is recommended for high ionic strength and proved for strong electrolyte or weak acid/base pair (Merkel and Planer-Friedrich, 2008):

$$-\log\gamma_{i,H^+} = A_I Z^2\left(\frac{I_i^{\frac{1}{2}}}{1+I_i^{\frac{1}{2}}} - 0.3I_i\right) \tag{4.1.8}$$

$$-\log\gamma_{i,A^-} = A_I Z^2\left(\frac{I_i^{\frac{1}{2}}}{1+I_i^{\frac{1}{2}}} - 0.3I_i\right) \tag{4.1.9}$$

where A_I is the temperature-dependent constant, calculated by (Merkel and Planer-Friedrich, 2008)

$$A_I = \frac{1.82483\times10^6\sqrt{d}}{(\epsilon T)^{1.5}} \tag{4.1.9-a}$$

$$d = 1 - \frac{\left((T-273)-3.9863\right)^2\left((T-273)+288.9414\right)}{508929.2\times\left((T-273)+68.12963\right)} + 0.011445e^{\frac{-374.3}{(T-273)}}$$

$$\tag{4.1.9-b}$$

$$\epsilon = 2727.586 + 0.6224107\times T - 466.9151\ln T - \frac{52000.87}{T}$$

$$\tag{4.1.9-c}$$

where d is density, ϵ is dielectric constant, Z is the charge of ion and I_i is ionic strength at support layer-active layer interface and calculated by

$$I_i = 0.5\left(2C_{i,s} + 2C_{i,H^+}\right) \tag{4.1.10}$$

4.1.1.5 Osmotic pressure at the active layer surface

The osmotic pressure at the active layer surface, π_{mf}, is the sum of osmotic pressure of salt and weak acid species at active layer surface. It can be expressed as

$$\pi_{mf} = \emptyset_s n_s C_{mf,s} RT + \emptyset_a \left(C_{mf,a} + C_{mf,H^+} \right) RT \qquad (4.1.11)$$

where $C_{mf,s}$ is the salt concentration at active layer surface, $C_{mf,a}$ and C_{mf,H^+} are the acid and hydrogen ion concentration at active layer surface respectively. C_{mf,H^+} can be determined by

$$C_{mf,H^+} = \sqrt{\frac{K_{a,T}(C_{mf,a} - C_{mf,H^+})}{\gamma_{mf,H^+} \gamma_{mf,A^-}}} \qquad (4.1.12)$$

where γ_{mf,H^+} and γ_{mf,A^-} are hydrogen ion activity coefficient and acid ion activity coefficient at active layer surface and calculated by

$$-\log \gamma_{mf,H^+} = A_I Z^2 \left(\frac{I_{mf}^{\frac{1}{2}}}{1 + I_{mf}^{\frac{1}{2}}} - 0.3 I_{mf} \right) \qquad (4.1.13)$$

$$-\log \gamma_{mf,A^-} = A_I Z^2 \left(\frac{I_{mf}^{\frac{1}{2}}}{1 + I_{mf}^{\frac{1}{2}}} - 0.3 I_{mf} \right) \qquad (4.1.14)$$

Where I_{mf} is ionic strength at active layer surface, determined by

$$I_{mf} = 0.5 \left(2 C_{mf,s} + 2 C_{mf,H^+} \right) \qquad (4.1.15)$$

4.1.1.6 Dilutive internal concentration polarization

The internal concentration polarization (ICP) of salt species inside the membrane support layer can be expressed as

$$-J_s = J_w C_s - \frac{dD_{eff}C_s}{dx} \tag{4.1.16}$$

where C_s is the salt concentration in membrane support layer. Related to the salt diffusion coefficient, D_s, by $D_{eff} = \varepsilon D_s$, D_{eff} is the effective salt diffusion coefficient in porous support layer where ε is membrane support layer porosity.

Integrating Eq. (4.1.16) across the membrane support layer thickness

at $x = 0, C_s = C_{i,s}$ and $x = l_{eff} = \tau l, C_s = C_{md,s}$

where $C_{i,s}$ is the salt concentration at support layer-active layer interface, $C_{md,s}$ is the salt concentration at support layer surface, l_{eff} is the effective thickness of support layer, τ is the tortuosity of support layer and l is the actual thickness of support layer yields

$$S = \int_{C_{i,s}}^{C_{md,s}} \frac{1}{(J_w C_s + J_s)} \, d(D_{d,s}C_s) \tag{4.1.16-a}$$

where S, defined by $S = \frac{\tau l}{\varepsilon}$, is the structure parameter of membrane support layer and $D_{d,s}$, the average NH$_4$Cl diffusion coefficient in draw solution, is calculated by (Cussler, 1997)

$$D_{d,s} = \frac{2}{(\frac{1}{D_{NH_4^+,T}} + \frac{1}{D_{Cl^-,T}})} \tag{4.1.16-b}$$

where $D_{NH_4^+,T}$ and $D_{Cl^-,T}$ are the ammonium ion and chloride ion diffusion coefficient respectively.

4.1.1.7 Dilutive external concentration polarization and the salt mass transfer coefficient at draw solution side

At support layer surface, the external concentration polarization (ECP), which is affected by dilutive permeate water from feed solution side, reduces the salt concentration away from membrane support layer surface at the draw solution side and is expressed as

$$-J_s = J_w C - \frac{d D_{d,s} C}{dx}$$ (4.1.17)

where C is the salt concentration in the boundary layer.

Integrating Eq. (4.1.17) across the boundary layer thickness (δ)

at $x = 0, C = C_{md,s}$ and $x = \delta, C = C_{d,s}$

where $C_{d,s}$ is the salt concentration in bulk draw solution yields

$$\delta = \frac{D_{d,s}}{k_d} = \int_{C_{md,s}}^{C_{d,s}} \frac{1}{(J_w C + J_s)} \, d(D_{d,s} C)$$ (4.1.17-a)

where k_d is the salt mass transfer coefficient at draw solution side.

For flowing in rectangular channel for the appropriate flow regimes in open channel at draw solution side, Sherwood number of salt, Sh_d, is represented by (McCutcheon & Elimelech, 2006; McCutcheon et al., 2006)

Laminar flow ($Re_d \leq 2{,}100$):

$$Sh_d = 1.85 (Re_d Sc_d \frac{d_h}{L})^{1/3}$$ (4.1.18-a)

Turbulent flow ($Re_d > 2{,}100$):

$$Sh_d = 0.04 Re_d^{3/4} Sc_d^{1/3}$$ (4.1.18-b)

where d_h is the hydraulic diameter and Re_d, the Reynolds number in draw solution channel, is defined by

$$Re_d = \frac{L v_d \rho_d}{\mu_d}$$ (4.1.19)

where L is the channel length of FO test cell, v_d is the average flow velocity in draw solution channel and ρ_d is the density of NH_4Cl solution as a function of its molar concentration and absolute temperature (Novotny & Sohnel, 1988). It can be calculated by

$$\rho_d = \rho_w + 0.2061 \times 10^2 C_{d,s} - 0.1577 C_{d,s}(T - 273) + 1.553 \times 10^{-3} C_{d,s}(T - 273)^2 - 2.556 C_{d,s}^{1.5} + 5.67 \times 10^{-2} C_{d,s}^{1.5}(T - 273) - 5.082 \times 10^{-4} C_{d,s}^{1.5}(T - 273)^2$$

(4.1.20)

where ρ_w is the water density, according to the empirical equation Eq. (4.20-a) (Novotny & Sohnel, 1988):

$$\rho_w = 999.65 + 2.0438 \times 10^{-1}(T - 273) - 6.174 \times 10^{-2}(T - 273)^{3/2}$$

(4.1.20-a)

where μ_d is the dynamic viscosity of NH_4Cl solution as a function of its molar concentration, density and absolute temperature. It can be calculated by (Marc, 2007)

$$\mu_d = 6e^{\left(\frac{12.396(53.491 C_{d,s}/\rho_d)^{1.5039} - 1.7756}{(0.23471(T-273)+1)\left(-2.7591(53.491 C_{d,s}/\rho_d)^{2.8408}+1\right)}\right)}$$

(4.1.21)

where Sc_d, Schmidt number of salt at draw solution side, is defined by

$$Sc_d = \frac{\mu_d}{\rho_d D_{d,s}}$$

(4.1.22)

The salt mass transfer coefficient at draw solution side, k_d, is related to Sh_d by

$$k_d = \frac{Sh_d D_{d,s}}{d_h}$$

(4.1.23)

    4.1.1.8 Concentrative concentration polarization and acid mass transfer coefficient at feed solution side

The active layer of the semi-permeable membrane retains selective solutes as the water molecules permeate across the membrane. Consequently, the solute concentration is built up in the boundary layer on the active layer surface. The concentrative concentration polarization can be expressed by

$$J_a = J_w C_a + \frac{d D_{f,a} C_a}{dx} \qquad (4.1.24)$$

where C_a is the acid concentration in the boundary layer.

Integrating Eq. (4.1.24) across the boundary layer thickness (δ)

at $x = 0, C_a = C_{mf,a}$ and $x = -\delta, C_a = C_{f,a}$

where $C_{mf,a}$ is the acid concentration at active layer surface and $C_{f,a}$ is the acid concentration in bulk feed solution, yields

$$\delta = \frac{D_{f,a}}{k_{f,a}} = \int_{C_{mf,a}}^{C_{f,a}} \frac{1}{(J_w C_a - J_a)} \, d(D_{f,a} C_a) \qquad (4.1.24\text{-}a)$$

where $k_{f,a}$, the acid mass transfer coefficient at feed solution side, is defined by

$$k_{f,a} = \frac{Sh_{f,a} D_{f,a}}{d_h} \qquad (4.1.25)$$

where $D_{f,a}$, the apparent weak acid diffusion coefficient in feed solution, is calculated by **(Cussler, 1997)**

$$D_{f,a} = \left(\frac{D_{ion,T}}{2KC_{f,a}} \left(-1 + \sqrt{1 + 4KC_{f,a}} \right) + \frac{D_{HA,T}}{4KC_{f,a}} \left(-1 + \sqrt{1 + 4KC_{f,a}} \right)^2 \right) \qquad (4.1.26)$$

where $D_{ion,T}$, the average acid ion diffusion coefficient, is defined by (Cussler, 1997)

$$D_{ion,T} = \frac{2}{\left(\frac{1}{D_{H^+,T}} + \frac{1}{D_{A^-,T}} \right)} \qquad (4.1.26\text{-}a)$$

where $D_{H^+,T}$ and $D_{A^-,T}$ are hydrogen and acid ion diffusion coefficient respectively, K equal to $1/K_{a,T}$, $K_{a,T}$ is the ionization constant, $C_{f,a}$ is the acid concentration in feed solution and $D_{HA,T}$ is the acid molecule diffusion coefficient.

$Sh_{f,a}$ is Sherwood number of acid in open channel at feed solution side. There are two different equations for different flow regimes, as following:

Laminar flow ($Re_f \leq 2{,}100$):

$$Sh_{f,a} = 1.85(Re_f Sc_{f,a} \frac{d_h}{L})^{1/3} \qquad \text{(4.1.27-a)}$$

Turbulent flow $(Re_f > 2{,}100)$:

$$Sh_{f,a} = 0.04 Re_f^{3/4} Sc_{f,a}^{1/3} \qquad \text{(4.1.27-b)}$$

where Re_f, Reynolds number in feed solution channel, is defined by

$$Re_f = \frac{L v_f \rho_f}{\mu_f} \qquad \text{(4.1.28)}$$

where v_f is the average flow velocity in feed solution channel and ρ_f is the density of feed solution. The dynamic viscosity of feed solution, μ_f, is assumed to be equal to water viscosity (ϑ_T), according to Eq. (4.1.28-a) **(Marc, 2007)**:

$$\vartheta_T = \frac{(T-273)+246}{(0.05594 \times (T-273)+5.2842) \times (T-273)+137.37} \qquad \text{(4.1.28-a)}$$

where $Sc_{f,a}$, Schmidt number of acid at feed solution side, is defined by

$$Sc_{f,a} = \frac{\mu_f}{\rho_f D_{f,a}} \qquad \text{(4.1.29)}$$

     4.1.1.9 Dilutive external concentration polarization and the salt mass transfer coefficient at feed solution side

That the leakage salts diffuse from draw solution side and build up within the boundary layer at the membrane active layer surface. As a result, the dilutive external concentration polarization can be expressed by

$$-J_s = J_w C - \frac{dD_{f,s}C}{dx} \tag{4.1.30}$$

Integrating Eq. (4.30) across the boundary layer thickness (δ)

at $x = 0, C_{mf,s}$ and $x = -\delta, C = C_{f,s}$

where $C_{mf,s}$ is salt concentration at active layer surface and $C_{f,s}$ is salt concentration in bulk solution, yields

$$-\delta = -\frac{D_{f,s}}{k_{f,s}} = \int_{C_{mf,s}}^{C_{f,s}} \frac{1}{(J_w C + J_s)} \, d(D_{f,s}C) \tag{4.1.30-a}$$

where the average salt diffusion coefficient in feed solution, $D_{f,s}$, is estimated to the salt diffusion coefficient in draw solution, $D_{d,s}$. $k_{f,s}$, the salt mass transfer coefficient at the feed solution side, is defined by

$$k_{f,s} = \frac{Sh_{f,s}D_{f,s}}{d_h} \tag{4.1.31}$$

where $Sh_{f,s}$ is Sherwood number of salt in open channel at feed solution side. There are two different equations for different flow regimes, as following:

Laminar flow ($Re_f \leq 2,100$):

$$Sh_{f,s} = 1.85(Re_f Sc_{f,s} \frac{d_h}{L})^{1/3} \tag{4.1.32-a}$$

Turbulent flow $(Re_f > 2,100)$:

$$Sh_{f,s} = 0.04 Re_f^{3/4} Sc_{f,s}^{1/3} \tag{4.1.32-b}$$

where $Sc_{f,s}$, Schmidt number of salt at feed solution side, is defined by

$$Sc_{f,s} = \frac{\mu_f}{\rho_f D_{f,s}} \qquad (4.1.33)$$
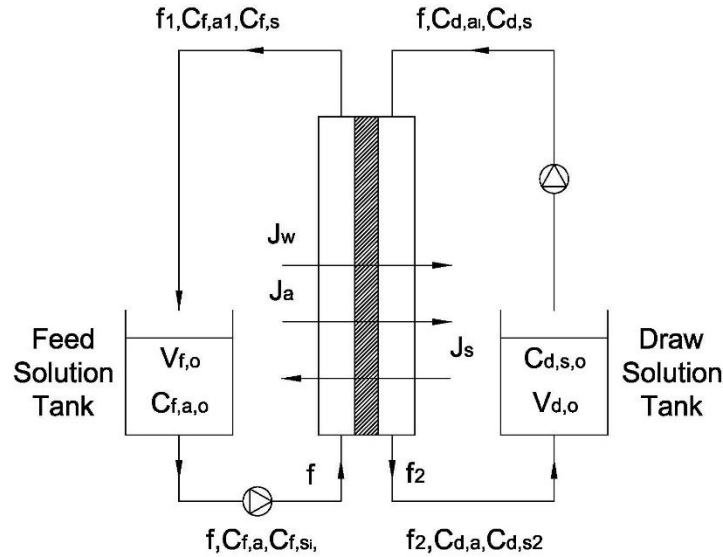
Fig. 4.2 Schematic diagram of forward osmosis system for carboxylic acid concentration.

4.1.1.10 Mole balance on operation units

In this experiment, the recirculating pump delivered the feed and draw solutions into the FO test cell unit, internally installed with a FO membrane piece, and then recycled them back to their storage tanks (Fig. 4.2) where the feed and draw solution would be concentrated and diluted respectively corresponded with elapsed time. Subsequently, the differential osmotic pressure across a membrane between feed and draw solution would decrease and the water flux of the system would also decline in the period of experiment time.

In a Fig. 4.2, making mole balance equations on acid and salt in the feed solution tank can be written as

$$V_{f,0}C_{f,a,0} + C_{f,a1}f_1 t - C_{f,a}ft = C_{f,a}\left(V_{f,0} + f_1 t - ft\right) \qquad (4.1.34)$$

$$C_{f,s}f_1 t - C_{f,si}ft = C_{f,si}\left(V_{f,0} + f_1 t - ft\right) \qquad (4.1.35)$$

where $V_{f,0}$ is the initial volume of feed solution, $C_{f,a,0}$ is the initial acid concentration in feed solution tank, $C_{f,a1}$ is the acid concentration from the outlet of feed solution channel, f is the flow rate of recirculating pump, t is elapsed time, and $C_{f,si}$ is the salt concentration into the inlet of feed solution channel. f_1, the flow rate from the outlet of feed solution channel, is determined by

$$f_1 = f - J_w A_M \tag{4.1.36}$$

where A_M is the effective membrane area.

Making mole balance equations on acid and salt in the feed solution channel can be written as

$$C_{f,a} f t - J_a A_M t = C_{f,a1} f_1 t \tag{4.1.37}$$

$$C_{f,si} f t + J_s A_M t = C_{f,s} f_1 t \tag{4.1.38}$$

Making mole balance equations on acid and salt in the draw solution tank can be written as

$$C_{d,a} f_2 t - C_{d,ai} f t = C_{d,ai}\left(V_{d,0} + f_2 t - f t\right) \tag{4.1.39}$$

$$C_{d,s,0} V_{d,0} + C_{d,s2} f_2 t - C_{d,s} f t = C_{d,s}\left(V_{d,0} + f_2 t - f t\right) \tag{4.1.40}$$

where $V_{d,0}$ is the initial volume of draw solution, $C_{d,s,0}$ is the initial salt concentration in draw solution tank, $C_{d,s2}$ is the salt concentration from the outlet of draw solution channel and $C_{d,ai}$ is the acid concentration into the inlet of draw solution channel. f_2, the flow rate from the outlet of draw solution channel, is determined by

$$f_2 = f + J_w A_M \tag{4.1.41}$$

Making mole balance equations on acid and salt in the draw solution channel can be written as

$$C_{d,ai}ft + J_a A_M t = C_{d,a}f_2 t \tag{4.1.42}$$

$$C_{d,s}ft - J_s A_M t = C_{d,s2}f_2 t \tag{4.1.43}$$

v_f, the average flow velocity in the feed solution channel, is calculated by

$$v_f = \frac{f_1 + f}{WD} \tag{4.1.44}$$

where W and D are the channel width and depth of FO test cell, respectively.

v_d, the average flow velocity in the draw solution channel, is calculated by

$$v_d = \frac{f_2 + f}{WD} \tag{4.1.45}$$

The weight change of draw solution, W_d, can be determined by

$$W_d = J_w A_M t \, \rho_d \tag{4.1.46}$$

where ρ_d is the density of NH_4Cl solution.

4.1.1.11 pH of feed solution

As feed solution was exposed to the atmosphere, open system, the carbonate species in the feed solution was in equilibrium with the CO_2 gas in the atmosphere over the solution. At ground level, the atmosphere contains $10^{-3.408}$atm of CO_2 gas (p_{CO_2}) and then equilibrates with feed solution. This can be written as

$$C_{H_2CO_3^*} \approx C_{f,CO_2} = K_H p_{CO_2} = 0.0387 \times 10^{-3.408} = 1.51 \times 10^{-5} M$$

where $K_H = 0.0387$ molar/atm, the temperature dependence Henry's constant at 301 K, is calculated from Eq. (4.1.69) and $C_{H_2CO_3^*}$ or C_{f,CO_2} is the carbonic acid concentration in feed solution. The dissociation equilibrium of diprotic carbonic acid in feed solution can be written as

$$H_2CO_3 \leftrightarrow H^+ + HCO_3^-$$

$$HCO_3^- \leftrightarrow H^+ + CO_3^{2-}$$
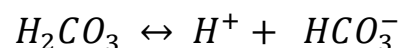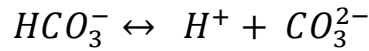
The equilibrium reactions are also established to determine C_{f,HCO_3^-} and $C_{f,CO_3^{2-}}$ by the following equations:

$$C_{f,HCO_3^-} = \frac{K_{CO_2,T}(C_{f,CO_2} - C_{f,HCO_3^-})}{C_{f,H^+}\gamma_{f,H^+}\gamma_{f,HCO3^-}} \tag{4.1.47}$$

$$C_{f,CO_3^{2-}} = \frac{K_{HCO_3^-,T}(C_{f,HCO_3^-} - C_{f,CO_3^{2-}})}{C_{f,H^+}\gamma_{f,H^+}\gamma_{f,CO_3^{2-}}} \tag{4.1.48}$$

where $K_{CO_2,T}$ and $K_{HCO_3^-,T}$ are the first ionization constant of carbonic acid, K_{a1}, and the second ionization constant of carbonic acid, K_{a2}, respectively, C_{f,HCO_3^-} is the bicarbonate ion concentration and $C_{f,CO_3^{2-}}$ is the carbonate ion concentration in feed solution. $\gamma_{f,H^+}, \gamma_{f,HCO3^-}, \gamma_{f,A^-}, \gamma_{f,OH^-}, \gamma_{f,CO_3^{2-}}$ and γ_{f,NH_4^+}, hydrogen, bicarbonate, acid, hydroxide, carbonate and ammonium ion activity coefficients in feed solution, are determined from Davis equation:

$$-\log\gamma_{f,H^+} = A_I Z^2 \left(\frac{I_f^{\frac{1}{2}}}{1+I_f^{\frac{1}{2}}} - 0.3I_f\right) \tag{4.1.49}$$

$$-\log\gamma_{f,HCO3^-} = A_I Z^2 \left(\frac{I_f^{\frac{1}{2}}}{1+I_f^{\frac{1}{2}}} - 0.3I_f\right) \tag{4.1.50}$$

$$-\log\gamma_{f,A^-} = A_I Z^2 \left(\frac{I_f^{\frac{1}{2}}}{1+I_f^{\frac{1}{2}}} - 0.3I_f\right) \tag{4.1.51}$$

$$-\log\gamma_{f,OH^-} = A_I Z^2 \left(\frac{I_f^{\frac{1}{2}}}{1+I_f^{\frac{1}{2}}} - 0.3I_f\right) \tag{4.1.52}$$

$$-\log\gamma_{f,CO_3^{2-}} = A_I Z^2 \left(\frac{I_f^{\frac{1}{2}}}{1+I_f^{\frac{1}{2}}} - 0.3I_f\right) \tag{4.1.53}$$

$$-\log\gamma_{f,NH_4^+} = A_I Z^2 \left(\frac{I_f^{\frac{1}{2}}}{1+I_f^{\frac{1}{2}}} - 0.3I_f\right) \tag{4.1.54}$$

where H^+, A^-, HCO_3^-, OH^- and NH_4^+ ion have one valence electron $(Z = 1)$, CO_3^{2-} has two valence electrons $(Z = 2)$ and I_f, the ionic strength of feed solution, is determined by

$$I_f = 0.5\left(C_{f,Cl^-} + C_{f,NH_4^+} + C_{f,H^+} + C_{f,A^-} + C_{f,OH^-} + C_{f,HCO_3^-} + 4C_{f,CO_3^{2-}}\right) \tag{4.1.55}$$

where the chloride ion concentration in feed solution, C_{f,Cl^-}, is equal to $C_{f,si}$, C_{f,NH_4^+} is the ammonium ion concentration, C_{f,H^+} is the hydrogen ion concentration, C_{f,OH^-} is the hydroxide ion concentration and C_{f,A^-} is the acid ion concentration in feed solution.

Soluble ammonium chloride salt releases ammonium ions into the solution. It functions as weak acid and ionizes a small amount of ammonia and hydrogen ion in equilibrium, written as
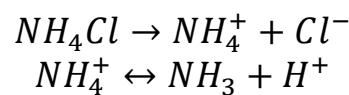
$$NH_4Cl \rightarrow NH_4^+ + Cl^-$$
$$NH_4^+ \leftrightarrow NH_3 + H^+$$

C_{f,NH_4^+} can be calculated from ionization constant which accounts for ionic strength and expressed by

$$C_{f,NH_4^+} = \frac{\gamma_{f,H^+} C_{f,H^+} C_{f,NH_3}}{\gamma_{f,NH_4^+} K_{NH_4^+,T}} \tag{4.1.56}$$

where $K_{NH_4^+,T}$ is the NH₄Cl ionization constant and C_{f,NH_3} is the ammonia concentration in feed solution.

In equilibrium, weak acid donates its proton into the water and can be written in the short notation as

$$HA \leftrightarrow A^- + H^+$$
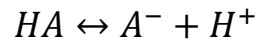
The above equilibrium reaction of weak acid can determine C_{f,A^-} by Eq. (4.1.57):

$$C_{f,A^-} = \frac{K_{a,T} C_{f,HA}}{C_{f,H^+} \gamma_{f,H^+} \gamma_{f,A^-}} \qquad (4.1.57)$$

where $C_{f,HA}$ is the acid molecule concentration in feed solution.

The self-ionization of water acts as either an acid or base as following reaction:

$$H_2O \leftrightarrow H^+ + OH^-$$

C_{f,OH^-} can be determined from the equilibrium constant and written as

$$C_{f,OH^-} = \frac{K_{W,T}}{C_{f,H^+} \gamma_{f,H^+} \gamma_{f,OH^-}} \qquad (4.1.58)$$

where $K_{W,T}$ is the water ionization constant.

The mass balance for the conjugate acid-base pair in feed solution can be expressed by

$$C_{f,si} = C_{f,NH_4^+} + C_{f,NH_3} \qquad (4.1.59)$$

$$C_{f,a} = C_{f,HA} + C_{f,A^-} \qquad (4.1.60)$$

The charge balance equation in feed solution can be described by

$$C_{f,H^+} + C_{f,NH_4^+} = C_{f,A^-} + C_{f,OH^-} + C_{f,si} + C_{f,HCO_3^-} + 2C_{f,CO_3^{2-}}$$

$$(4.1.61)$$

The pH of feed solution, pH_f, can be calculated by

$$pH_f = -\log{(\gamma_{f,H^+} C_{f,H^+})} \qquad\qquad (4.1.62)$$

4.1.1.12 List of all variables and unit in mathematical model

Table 4.1 Constant variables

Variable	Value	Unit	Ref.
$D_{Cl^-,298}$	2.032×10^{-5}	cm^2/sec	(Haynes, 2014-2015)
$D_{H^+,298}$	9.311×10^{-5}	cm^2/sec	(Haynes, 2014-2015)
$D_{NH_4^+,298}$	1.957×10^{-5}	cm^2/sec	(Haynes, 2014-2015)
n_s	2	−	-
R	0.08314	$L\ bar\ K^{-1}mol^{-1}$	-
σ	1		-
$\emptyset_s$	0.897	−	(Robinson and Stokes, 1959)

Table 4.2 Diffusion coefficients of carboxylic acid in water at 298 K

Acid Type	$D_{A^-,298}$ (cm^2/sec)	Ref.	$D_{a,298}$ (cm^2/sec)	Ref.	$D_{HA,298}$[a] (cm^2/sec)
Acetic acid	1.089×10^{-5}	(Bidstrup and Geankoplis, 1963)	1.27×10^{-5}	(Haynes, 2014-2015)	1.26×10^{-5}
Butyric acid	0.868×10^{-5}	(Bidstrup and Geankoplis, 1963)	0.918×10^{-5}	(Haynes, 2014-2015)	0.905×10^{-5}
Lactic acid	1.033×10^{-5}	(Bidstrup and Geankoplis, 1963)	0.993×10^{-5}	[Ribeiro et al., 2005)	0.764×10^{-5}
Valeric acid	0.871×10^{-5}	(Bidstrup and Geankoplis, 1963)	0.817×10^{-5}	(Haynes, 2014-2015)	0.80×10^{-5}

[a] $D_{HA,298}$ was calculated from Eq. (4.1.26)

Table 4.3 Fixed variables

Variable	Value	Unit	Variable	Value	Unit
A_M	0.42	dm^2	f	1.58	L/min
D	0.023	dm	T	301	K
d_h	0.0438	dm	W	0.4572	dm

Table 4.4 Initial independent variables

Variable	Value	Unit	Variable	Value	Unit
$C_{d,s,0}$	0.5	$mole/L$	$V_{f,0}$	1	L
$C_{f,a,0}$	0.01	$mole/L$	t	$0\ to\ 30$	hr
$V_{d,0}$	0.5	L			

Table 4.5 Unknown dependent variables

Variable	Unit	Variable	Unit
1. $C_{d,a}$	$mole/L$	32. J_w	$L/dm^2/min$
2. $C_{d,ai}$	$mole/L$	33. k_d	dm/min
3. C_{d,H^+}	$mole/L$	34. $k_{f,a}$	dm/min
4. $C_{d,s}$	$mole/L$	35. $k_{f,s}$	dm^2/min
5. $C_{d,s2}$	$mole/L$	36. pH_f	-
6. $C_{f,a}$	$mole/L$	37. Re_d	-
7. C_{f,A^-}	$mole/L$	38. Re_f	-
8. $C_{f,ai}$	$mole/L$	39. Sc_d	-
9. $C_{f,CO_3^{2-}}$	$mole/L$	40. $Sc_{f,a}$	-
10. C_{f,H^+}	$mole/L$	41. $Sc_{f,s}$	-
11. $C_{f,HA}$	$mole/L$	42. Sh_d	-
12. C_{f,HCO_3^-}	$mole/L$	43. $Sh_{f,a}$	-
13. C_{f,NH_3}	$mole/L$	44. $Sh_{f,s}$	-
14. C_{f,NH_4^+}	$mole/L$	45. v_d	dm/min
15. C_{f,OH^-}	$mole/L$	46. v_f	dm/min
16. $C_{f,s}$	$mole/L$	47. W_d	g
17. $C_{f,si}$	$mole/L$	48. γ_{f,A^-}	-
18. $C_{i,a}$	$mole/L$	49. $\gamma_{f,CO_3^{2-}}$	-
19. $C_{i,s}$	$mole/L$	50. γ_{f,H^+}	-
20. $C_{md,s}$	$mole/L$	51. $\gamma_{f,HCO3^-}$	-
21. $C_{mf,a}$	$mole/L$	52. γ_{f,NH_4^+}	-
22. C_{mf,H^+}	$mole/L$	53. γ_{f,OH^-}	-
23. $C_{mf,s}$	$mole/L$	54. γ_{i,A^-}	-
24. $D_{f,a}$	dm^2/min	55. γ_{i,H^+}	-
25. f_1	L/min	56. γ_{mf,A^-}	-
26. f_2	L/min	57. γ_{mf,H^+}	-
27. I_f	$mole/L$	58. μ_d	$g/dm/min$
28. I_i	$mole/L$	59. π_i	bar
29. I_{mf}	$mole/L$	60. π_{mf}	bar
30. J_a	$mole/dm^2/min$	61. ρ_d	g/L
31. J_s	$mole/dm^2/min$	62. ρ_f	g/L

4.1.1.13 Temperature dependence of constant variables

The solution temperature is one of the major factors which directly influence the performance of membrane process. It has the positive correlation with permeate water flux. Thereby, the temperature dependence of constant variables (the equilibrium constants, the solute diffusion coefficients and the Henry's law constant correction) are provided as functions of temperature. These constant variables must be adjusted to desired operation temperature, 301K, prior to use in the model.

4.1.1.14 Equilibrium constants

The equilibrium constant is a function of temperature. As the temperature increases, the equilibrium reaction produces more ionization products. The ionization constants at the given temperature, $K_{a,T}$, can be determined by the equation in Table 4.6.

Table 4.6 Ionization constants as functions of temperature

Acid type	Equation	Reference
Acetic acid	$-\log K_{a,T} = \dfrac{1170.48}{T} - 3.1649 + 0.013399 \times T$	(Robinson and Stokes, 1959)
Butyric acid	$-\log K_{a,T} = \dfrac{1033.39}{T} - 2.6215 + 0.01334 \times T$	(Robinson and Stokes, 1959)
Lactic acid	$-\log K_{a,T} = \dfrac{1286.49}{T} - 4.8607 + 0.014776 \times T$	(Robinson and Stokes, 1959)
Valeric acid	$-\log K_{a,T} = \dfrac{921.38}{T} - 1.8574 + 0.012105 \times T$	(Robinson and Stokes, 1959)
Carbonic acid	$-\log K_{a_1,T} = \dfrac{3404.71}{T} - 14.8435 + 0.032786 \times T$	(Robinson and Stokes, 1959)
	$-\log K_{a_2,T} = \dfrac{2902.39}{T} - 6.4980 + 0.02379 \times T$	(Robinson and Stokes, 1959)
Ammonium chloride	$-\log K_{a,T} = \dfrac{2835.76}{T} - 0.6322 + 0.001225 \times T$	(Robinson and Stokes, 1959)
Water	$-\log K_{w,T} = \dfrac{4470.99}{T} - 6.0875 - 0.017060 \times T$	(Harned and Owen, 1958)

4.1.1.15 Solute diffusion coefficients

Solute diffusion coefficient and viscosity of solvent are functions of temperature. According to the well-known Stokes-Einstein equation (Einstein, 1926), the temperature correction of the diffusion coefficient for any solute can be calculated by the following equations:

$$D_{H^+,T} = D_{H^+,298} \times \frac{T}{298} \times \frac{\vartheta_{298}}{\vartheta_T} \qquad (4.1.63)$$

$$D_{A^-,T} = D_{A^-,298} \times \frac{T}{298} \times \frac{\vartheta_{298}}{\vartheta_T} \qquad (4.1.64)$$

$$D_{HA,T} = D_{HA,298} \times \frac{T}{298} \times \frac{\vartheta_{298}}{\vartheta_T} \qquad (4.1.65)$$

$$D_{NH_4^+,T} = D_{NH_4^+,298} \times \frac{T}{298} \times \frac{\vartheta_{298}}{\vartheta_T} \qquad (4.1.66)$$

$$D_{Cl^-,T} = D_{Cl^-,298} \times \frac{T}{298} \times \frac{\vartheta_{298}}{\vartheta_T} \qquad (4.1.67)$$

$$\vartheta_T = \frac{(T-273)+246}{(0.05594 \times (T-273)+5.2842) \times (T-273)+137.37} \qquad (4.1.68)$$

where T is absolute temperature, ϑ_T and ϑ_{298} are water viscosity at the given temperature and at 298 kelvins, respectively.

4.1.1.16 Henry's law constant

Henry's law constant as a function of temperature is defined by (Compilation of Henry's Law Constant)

$$K_H = K_{H^\theta} \exp(\frac{-\Delta_{soln}H}{R}(\frac{1}{T} - \frac{1}{T^\theta}))$$ (4.1.69)

where K_{H^θ} is the Henry's law constant at standard condition (T^θ=298.15 kelvins), 3.4×10^{-2} molar/atm, R is the gas constant and $\frac{-\Delta_{soln}H}{R}$ is the temperature dependence, **2400** Kelvins.

4.1.2 Model solutions

A set of sixty-two simultaneous equations, given with the sixty-two jointly dependent variables, describes the logical phenomena during simulating the filtration of carboxylic acid in FO process. This system of equation model was solved by Levenberg-Marquardt (LM) algorithm running in MATLAB R2012a software in order to simultaneously determine the sixty-two jointly dependent variables at each point in simulating time. The procedures of model solutions are described in Fig. 4.3. The calculation was initiated by inputting constant variables (e.g. acid ionization constant, diffusion coefficient, gas constant), fixed variables (e.g. membrane area, test cell channel sizing, liquid temperature, flow rate of pump), initial conditions (e.g. initial draw solution volume and concentration, initial feed solution volume and concentration) and the assumed initial dependent variables at $t = 1$. Based on Levenberg-Marquardt algorithm, the sixty-five dependent variables at $t = 1$ could be simultaneously determined and were applied as the new initial dependent variables at $t = 2$. The sets of dependent variables at each point in time ($t = 1, 2, 3,..., n$) could be yielded at the end of each simulation cycle.
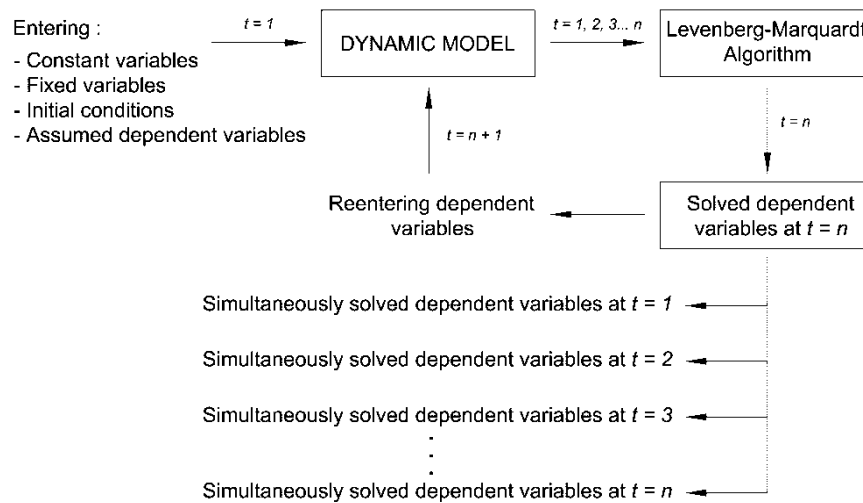
Fig. 4. 3 The flow chart of model solution procedures 1 The flow chart of model solution proceduresThe flow chart of model solution procedures.

4.1.3 FO membrane parameter characterization

Table 4.7 Membrane parameters (A, B_s, S), with the correlated coefficient of determinations of water flux $(R^2 - J_w)$, salt flux $(R^2 - J_s)$ and the coefficient of variation (CV), have been calculated by excel error minimization algorithms from Ref. (Tiraferri et al., 2013).

Sample	A $(L/m^2/h/bar)$	B_s $(L/m^2/h)$	S (μm)	$R^2 - J_w$	$R^2 - J_s$	$CV(\%)$
1	0.419	0.463	494	0.994	0.986	3.83
2	0.382	0.433	456	1	0.993	2.86
3	0.439	0.484	524	0.982	0.998	4.71
Mean Value	0.413	0.460	491			

Three samples of TFC membranes were used to determine water permeability coefficient, NH$_4$Cl permeability coefficient and structure parameters (A, B_s, S) of the membrane in the FO experiment. In Table 4.7, the simulation results are reported along with the related coefficient of determination of water flux $(R^2 - J_w)$, salt flux $(R^2 - J_s)$ and the coefficient of variation (CV). The mean values of the A, B_s and S

obtained from excel algorithms were 0.413 L/m^2/h/bar, 0.460 L/m^2/h and 491 μm respectively.

Table 4.8 Experiment results and calculation report of four acid permeability coefficients (B_a) by excel algorithms

Acid Type	Sample	Stage	$C_{d,a}$ (mM)	$C_{f,a}$ (mM)	J_w $(L/m^2/h)$	J_a $(mmole/m^2/h)$	B_a $(L/m^2/h)$	Mean Value $(L/m^2/h)$
Acetic	1	1	379	8.82	1.6	930.83	1.979	2.10
		2	736	29.8	3.4	1,999.60	2.145	
	2	1	488	19.3	2.2	1,337.28	2.178	
		2	906	42.0	3.9	2,385.42	2.102	
Butyric	1	1	392	2.11	1.8	241.46	0.576	0.64
		2	606	5.89	2.1	343.86	0.539	
	2	1	280	1.09	1.4	239.31	0.767	
		2	521	5.14	2.1	379.28	0.674	
Valeric	1	1	120	3.02	0.5	59.42	0.470	0.49
		2	254	8.97	0.9	123.87	0.468	
	2	1	159	5.19	0.7	85.23	0.510	
		2	292	12.74	1.1	158.02	0.521	
Lactic	1	1	365	0.17	4.3	75.22	0.198	0.15
		2	536	0.55	5.2	94.22	0.170	
	2	1	415	0.51	3.1	54.23	0.128	
		2	677	1.48	4.9	79.96	0.116	

To characterize the acid permeability coefficients of the membrane, two membrane samples were tested for each acid type. The mean value of acid permeability coefficient (B_a) was calculated from two membrane samples and determined by two-stage individual calculation, using excel algorithms. The calculation of J_w and J_a were delineated in Appendix B. The experimental data and simulation results of four different carboxylic acids are reported in Table 4.8. The highest permeability value was acetic acid, 2.10 L/m^2/h. Butyric and valeric acid had relatively close permeability value, 0.64 L/m^2/h and 0.49 L/m^2/h, respectively and the value of

lactic was lowest at 0.15 L/m^2/h. These all performance parameters were inputted to the upcoming mathematical model.

4.1.4 Validation of the developed mathematical model
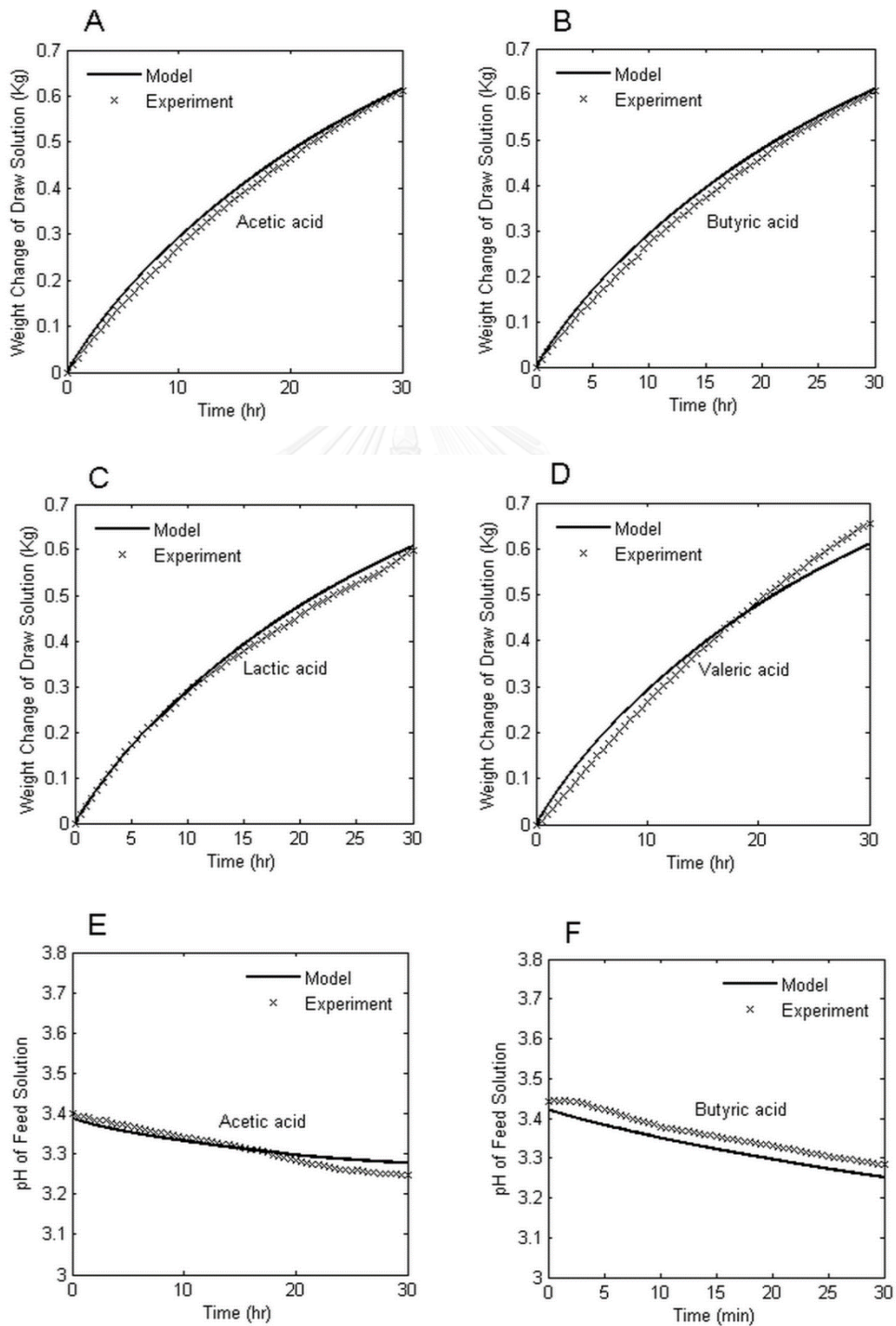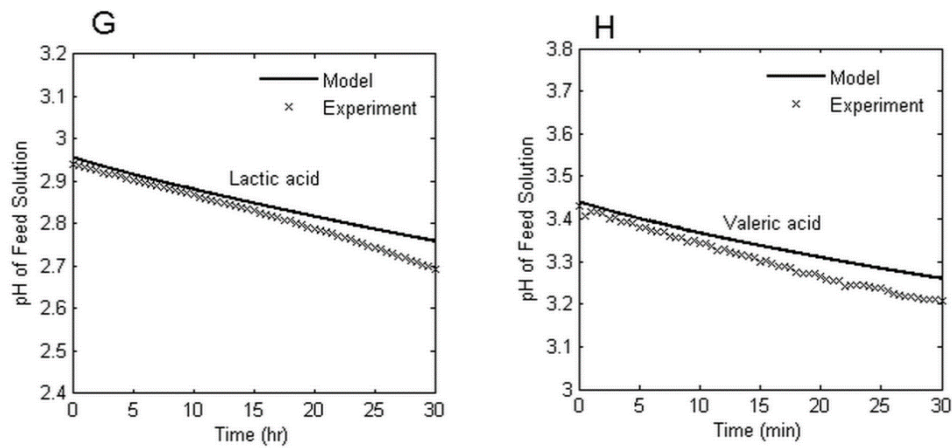
Fig. 4.4 Comparison of the simultaneous results from developed mathematical model (solid line) and the experimental data (cross symbols) from the acid concentration FO experiment of four different carboxylic acids, weight change of draw solution (A, B, C, D) and pH of feed solution (E, F, G, H), are plotted against elapsed time.

In Fig. 4.4, the model has also been compared to dynamic experiments performed with acid concentration by FO processes. Each carboxylic acid model was simulated to predict the weight changes of draw solution (W_d) and pH of feed solution as a function of time. The values of predicted and measured weight change of draw solution (Fig. 4.4A-D) and pH of feed solution (Fig.4.4E-H) were compared in the 30 hour operation. As a result of the different osmotic pressures of feed and draw solutions, the weight change of draw solution (W_d) increases over the period of experimental time by convective permeate water from feed solution accumulated in draw solution tank. Consequently, the volume reduction of acid solution in feed solution tank affects to the pH of feed solution declining over the period of time. The agreement between the simulated results from proposed model and experimental data has been evaluated by statistical factors, as shown in Table 4.9. The coefficient of determination, R^2, is used to gauge the goodness of developed model fit to experimental data. R^2 of W_d of all four carboxylic acids were higher than 0.95. In case of pH of feed solution, due to the extremely high sensitivity of pH probe, any additional

undefined variances from external can easily lead to variation of measuring pH. These cannot be unexpressed in the developed model, but affect directly to R^2 value. Another important aspect of mathematic model validation is the accuracy of the model. This can be measured by root mean square error, $RMSE$. The $RMSE$ of both W_d and pH of all carboxylic acids were fairly close, but represented in different unit of measurement. This value can be interpreted as the standard deviation of unexplained variances. The comparison of prediction accuracy between W_d and pH was performed by using standard error of prediction, SEP, which is unit-less metric. Results in Table 4.9 indicate that the prediction accuracy of W_d is lower than pH, as the unexplained variances on the accuracy of developed model to predict W_d is likely to have more impact than pH, even though the model can comprehend over 98% indicated by R^2 of all W_d phenomena during the period of time in each experiment. Nevertheless, $SEPs$ of all W_d-predictions were still less than 7% where the $SEPs$ of all pH-predictions were less than 1.13%.

Table 4.9 Quantitative comparisons of model predictions to experimental data

Acid	Weight of Draw Solution			pH of Feed Solution		
	R^2	$RMSE$	$SEP\%$	R^2	$RMSE$	$SEP\%$
Acetic	0.99	0.0164	4.65	0.71	0.0171	0.52
Butyric	0.99	0.0170	4.85	0.68	0.0332	0.99
Valeric	0.98	0.0253	6.99	0.64	0.0375	1.13
Lactic	0.99	0.0149	4.19	0.77	0.0304	1.08

4.1.5 Simulations

4.1.5.1 Simulation results of acetic acid feed solution at 30 hour operation of FO process, acid rejection and concentration performance of FO process as functions of time
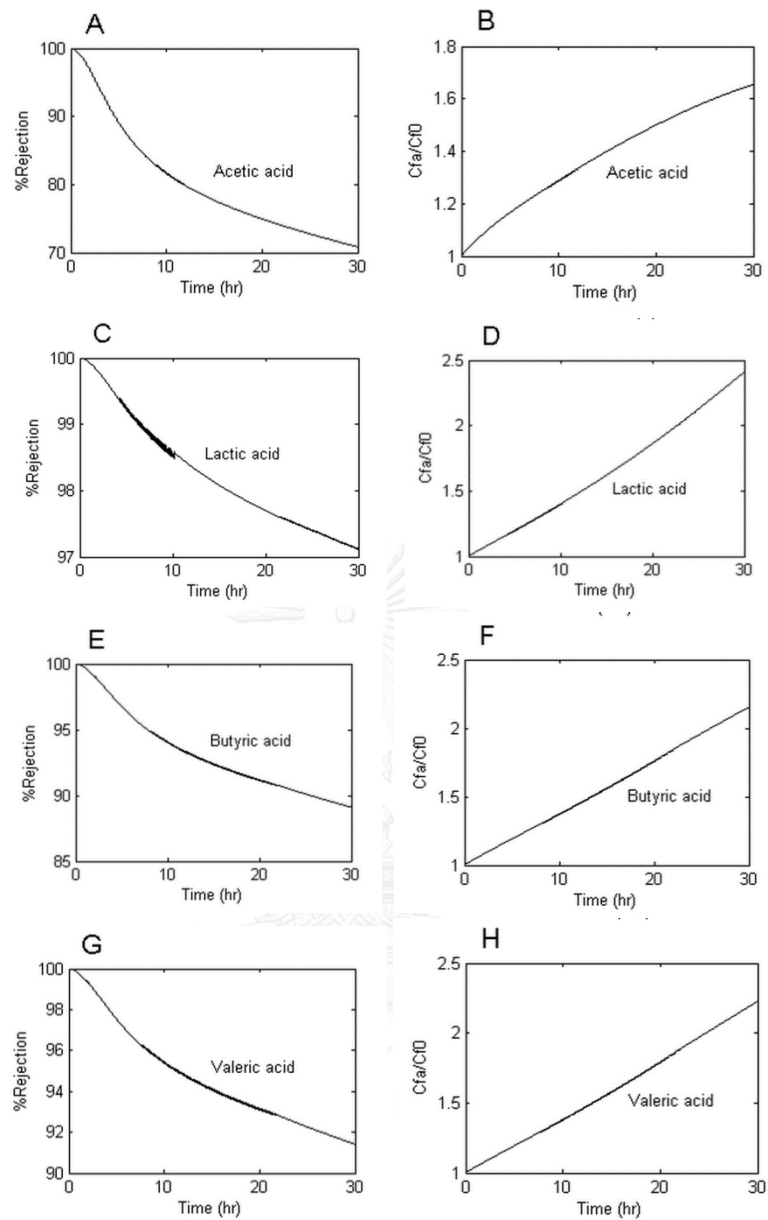
Fig. 4.5 Rejection rate and concentration performance of the FO process from simulation results of four carboxylic acids as functions of time for 30 hour system operation.

In order to quantify the concentration performance of carboxylic acid in the FO process, the developed mathematical model was simulated to predict concentration performance $(C_{f,a}/C_{f,0})$, defined by acid concentration $(C_{f,a})$ which is normalized by its initial concentration $(C_{f,0}=10$ mM) and acid rejection (R_a) as functions of time. The acid rejection of the FO system, R_a, can be expressed by

$$R_a = 1 - \frac{C_{d,ai}}{C_{f,a}} \tag{56}$$

where $C_{d,ai}$ is acid concentration in draw solution tank and $C_{f,a}$ is acid concentration in feed solution tank.

Fig. 4.5 shows rejection and concentration performance of four carboxylic acids as functions of time for 30 hour system operation. According to acetic acid (Fig. 4.5A and B), acid rejection declined rapidly with elapsed time, comparing to other carboxylic acids. At the end of simulation, 30 hours, it showed the acid rejection 71% and concentration up to 1.65 fold increase. Table 4.10 summarizes the acid rejection and concentration performance at 30 hour system operation. Acetic acid performs the lowest acid rejection and concentration performance due to its minimum molar mass, corresponding to the highest acid permeability coefficient among the other carboxylic acids. Butyric and valeric acid have the same pKa value but valeric acid has higher molar mass than butyric acid, causing acid rejection and concentration performance of valeric acid to be higher than those of butyric. In other words, valeric acid is retained easier in the feed solution side than butyric acid. Therefore, the valeric permeability coefficient is rationally lower than butyric acid. Comparing lactic acid to butyric acid, the molar mass of lactic acid is close to butyric acid but lactic acid has significantly lower pKa than butyric acid. Consequently, more ionization products of lactic acid will appear in feed solution. This rising of ionic strength in feed solution affects the increase of membrane surface charge. In this moment, the membrane is certainly able to hold the lactate and hydrogen ion by pushing them away from its surfaces. Lactic acid therefore performed the highest acid rejection of 97.3% and concentration performance of 2.4 fold increase corresponding to the lowest acid permeability coefficient.

Table 4.10 Summary of acid rejections and concentration performances of four carboxylic acids at the end of simulation, thirty hours, along with their chemical properties and acid permeability coefficients

Acid	Rejection	$C_{f,a}/C_{f,0}$	Molar mass (g/mole)	Pk_a	B_a (L/m²/h)
Acetic	71	1.65	60.05	4.75	2.10
Butyric	89	2.2	88.11	4.82	0.64
Valeric	91.5	2.3	102.13	4.82	0.49
Lactic	97.2	2.4	90.08	3.86	0.15

4.1.5.2 Rejection and concentration performance as functions of draw solution concentration and draw solution volume
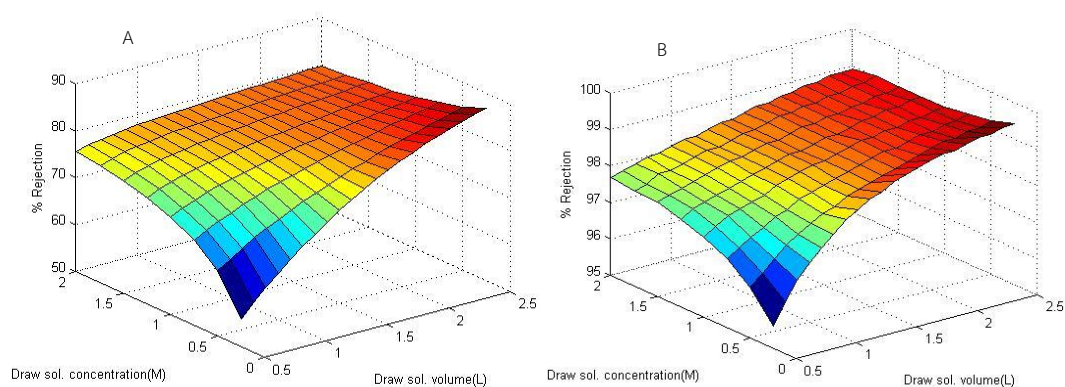
Fig. 4. 6 3-D curved surfaces of acid rejection for acetic acid (A) and lactic acid (B) as functions of the draw solution concentration (M) and of the draw solution volume (L) at 30 hour system operation.

Fig. 4.6 shows 3-D curved surfaces of rejection for acetic acid (Fig. 4.6A) and lactic acid (Fig. 4.6B) as functions of the draw solution concentration (M) and of the draw solution volume (L) at 30 hour system operation. The shape of curved surfaces for these two carboxylic acids is obviously similar. The highest rejection of acetic and lactic acid, 87% and 99.5% respectively, were at 0.25 M of and 2.5 L of draw solution, the minimum concentration and maximum volume of draw solution, whereas the lowest rejection of acetic and lactic acid, 55% and 95.5% respectively, was at 0.25 M and 0.5 L of draw solution, the minimum concentration and volume of draw solution. According to curved surfaces of acetic acid (Fig. 4.6A), in the draw solution area range

from 1.5 L to 2.5 L for any draw solution concentration, the system could perform greater than 79% rejection, remarked by the flat curved surfaces in this area. In similar fashion, lactic acid (Fig. 4.6B) could achieve the rejection greater than 98% in the draw solution area range from 1.25 L to 2.5 L for any draw solution concentration. In the aspect of system operation flexibility and in order to maximize the utilization of draw solution, it evidences that the developed model can suggest the optimal region of initial condition of draw solution to attain the nearby maximum rejection of the system.

4.1.5.3 Concentration performance as functions of draw solution concentration and draw solution volume.
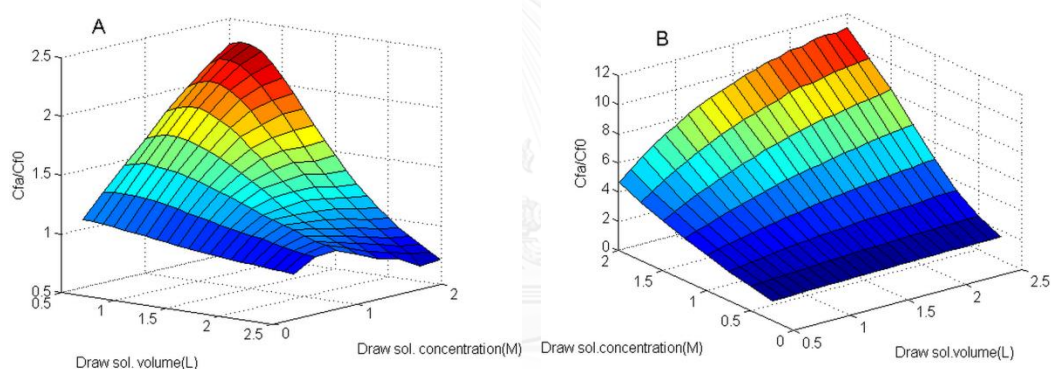
Fig. 4. 7 3-D curved surfaces of concentration performance for acetic acid (A) and lactic acid (B) as functions of the draw solution concentration (M) and of the draw solution volume (L) at 30 hour system operation.

Fig. 4.7 illustrates the concentration performance for acetic acid (Fig. 4.7A) and lactic acid (Fig. 4.7B) as functions of the draw solution concentration (M) and of the draw solution volume (L) at 30 hour system operation. The physical shapes of these two curved surfaces are different. According to acetic acid curved surface (Fig. 4.7A), the maximum concentration performance of acetic acid was concentrated up to 2.3 fold increase on condition that the FO system operated with the initial condition of 2 M and 0.6 L of draw solution, represented by the top of curved surfaces. As regards

the nearby maximum concentration performance, the system performance could accomplish the concentration performance greater than 2 fold increase in the region of draw solution range from 1.5 M to 2 M and 0.5 L to 1.0 L. The curved surfaces then moved down rapidly, either draw solution concentration reduced to 0.25 M or draw solution volume increased to 2.5 L. Once the draw solution volume was increasing more than 0.6 L, the diluted acetic acid concentration in the draw solution will dominantly effect on rising of acetic acid flux into the draw solution, which reduces concurrently both acetic acid concentration in feed solution and concentration performance. Eventually, the concentration performance of acetic acid was down to 0.75 fold decrease at 2 M and 2.5 L of draw solution, located at the base of curved surfaces. According to curved surfaces of lactic acid (Fig. 4.7B), the system performance could achieve the concentration performance greater than 10 fold increase, the nearby maximum concentration performance, in the region of draw solution range from 1.75 M to 2 M and 1.75 L to 2.5 L. The peak of curved surfaces appeared at the concentration up to more than 11 fold increase with the extreme initial condition, 2 M and 2.5 L of draw solution. This reveals that the higher concentration performance of the system can proportionally be obtained from the increasing of initial condition of draw solution. This is caused by the high rejection membrane property or low membrane permeability coefficient of lactic acid. Regarding operational flexibility in FO processes, it can be seen that 3D simulation of the developed model can illustrate the availability of FO process to operate over a range of draw solution conditions while satisfying process performance.

4.1.5.4 Sensitivity analysis

The purpose of sensitivity analysis is to determine the influence of independent variable on dependent variable in terms of percentage change. Changing a value of independent variable affects the dependent variables in varied manners, relying on

the function of each independent variable on dependent variable. The sensitivity analysis of rejection rate and concentration performance, dependent variables, have been conducted as functions of four initial condition variables, independent variables, which were initial draw solution concentration ($C_{d,s,0}$ = 1.0 M), initial acid concentration ($C_{f,a,0}$ = 10 mM), initial draw solution volume ($V_{d,0}$ = 0.5 L) and initial feed solution volume ($V_{f,0}$ = 1.0 L). The sensitivity analysis results of rejection rate and concentration performance for acetic and lactic acids at 30 hour system operation have been determined and presented in sensitivity charts, as shown in Fig. 4.8 A-D. Referred to exhibited charts, the sensitivities of both dependent variables were relatively distinctive in degree of sensitivity and correlation, depending on acid types of feed solution and assigned independent variables.
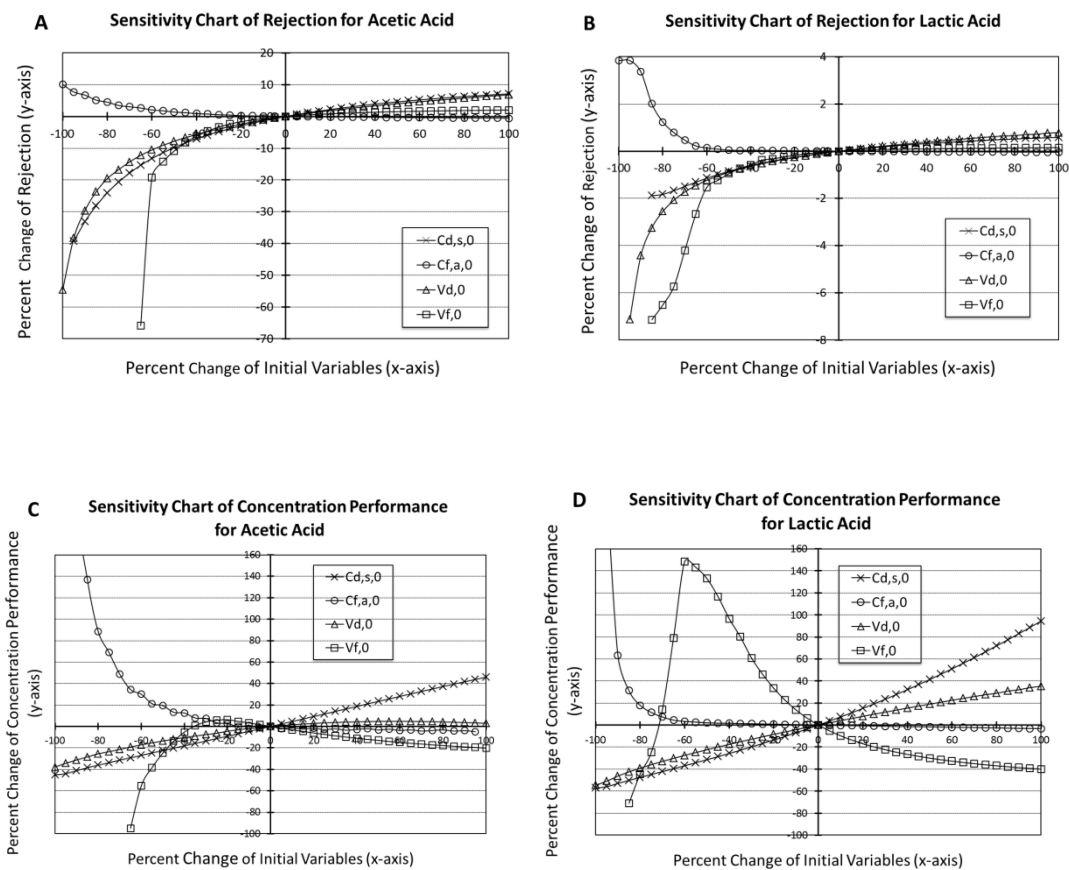
Fig. 4. 8 Sensitivity charts of rejection rate and concentration performance for acetic and lactic acids. Sensitivity charts of rejection rate and concentration performance for acetic and lactic acids.

Table 4.11 Correlation of rejection to $C_{d,s,0}$, $C_{f,a,0}$, $V_{d,0}$ and $V_{f,0}$ for each acid feed solution

Initial variable	Acetic acid	Lactic acid
$C_{d,s,0}$	Positive	Positive
$C_{f,a,0}$	Negative	Negative
$V_{d,0}$	Positive	Positive
$V_{f,0}$	Positive	Positive

Table 4.12 Correlation of $C_{f,a}/C_{f,a,0}$ to $C_{d,s,0}$, $C_{f,a,0}$, $V_{d,0}$ and $V_{f,0}$ for each acid feed solution

Initial variable	Acetic acid	Lactic acid
$C_{d,s,0}$	Positive	Positive
$C_{f,a,0}$	Negative	Negative
$V_{d,0}$	Positive	Positive
$V_{f,0}$	Negative, greater than -30%	Negative, greater than -60%

The shape of sensitivity curve can unveil correlation and degree of sensitivity of designated dependent variables on independent variables. Categorized by feed solution types, the correlation of rejection rate and concentration performance among $C_{d,s,0}$, $C_{f,a,0}$, $V_{d,0}$, and $V_{f,0}$ are demonstrated in Table 4.11 and 4.12. In sensitivity curves, by comparing with initial values, the increase and decrease rates of variable values of are expressed in positive and negative percentages correspondingly. Two types of correlation between independent and dependent variables are positive and negative. With positive correlation, a relationship between two variables is that their values

increase or decrease together. On the other hand, the negative correlation results in a relationship between two variables in which one variable increases as the other decreases, and vice versa. Typically the negative correlation is favorable because the lesser value of independent variable can efficiently yield the greater value of dependent variable.

Referred to sensitivity charts of rejection rate for acetic acid (Fig. 4.8A) and lactic acid (Fig. 4.8B) ,increasing rejection rate was given by reducing $C_{f,a,0}$ or increasing any of $C_{d,s,0}, V_{d,0}$ and $V_{f,0}$. Negatively correlated with $C_{f,a,0}$, the rejection rates began to increase rapidly when percentages of change in $C_{f,a,0}$ were less than -40% and -60% of acetic acid and lactic acid respectively. However, the rejection rate of acetic acid could gain 10% at -100% of change, while the rejection rate of lactic acid could only reach 3.8% at -100% of change. On the contrary, due to the positive correlations, to raise the rejection rate was to enlarge any of $C_{d,s,0}, V_{d,0}$ and $V_{f,0}$ values. According to acetic acid sensitivity chart, with the steepest curve on the positive side, $C_{d,s,0}$ attained the highest degree of sensitivity. It means that at the same percentages of increase the highest rejection rate could be achieved by raising $C_{d,s,0}$, comparing with $V_{d,0}$ and $V_{f,0}$. However, the degree of sensitivity of $V_{d,0}$ was very close to $C_{d,s,0}$ while $V_{f,0}$ was the lowest. In case of lactic acid, increasing $V_{d,0}$ acquired the highest degree of sensitivity while $V_{f,0}$ gained the lowest degree of sensitivity same as acetic. At 100% of change in $V_{d,0}$, the rejection rate could be escalated only to 0.78%, less than 7.1% of rejection rate of acetic acid at 100% of change in $C_{d,s,0}$.

Conversely, a rejection rate also could be decreased by increasing $C_{f,a,0}$ (negative correlation) or reducing any of $C_{d,s,0}, V_{d,0}$ and $V_{f,0}$ (positive correlation). For both acetic and lactic acids, increasing $C_{f,a,0}$ caused a comparatively insignificant effect to reduce a rejection rate while compared with decreasing $C_{d,s,0}$ $V_{d,0}$ and $V_{f,0}$. Decreasing $V_{f,0}$ leaded to the highest percentage of rejection rate deduction only when

decreasing percent was greater than about 45%. In other words, at less than -45% of change, decreasing $V_{f,0}$ of any acid had the highest degree of sensitivity whereas the lowest degree of sensitivity was acquired by decreasing $V_{d,0}$ for acetic acid and $C_{d,s,0}$ for lactic acid. In another condition, whereas percentages of change was greater than -45%, of acetic acid, reducing $C_{d,s,0}$ caused the highest influence to rejection rate, compared to $V_{d,0}$ and $V_{f,0}$. Unlike acetic acid, the highest degrees of sensitivity for rejection rate of lactic acid could be obtained by reducing $V_{d,0}$ and reducing $V_{f,0}$ leaded to the least degrees of sensitivity instead. Nevertheless, by comparing the effect of decreasing positive correlated independent variables on rejection rate, the effect on rejection rate of acetic acid was significantly higher than lactic acid, as shown in the sensitivity charts of rejection rate.

Another designated dependent variable is concentration performance ($C_{f,a}/C_{f,a,0}$) which has been analyzed in terms of sensitivity against the same set of independent variables as rejection rate. The shapes of sensitivity curve of acetic and lactic acids are nearly alike in pattern (Fig. 4.8C and D). Hence, for both acids, changing $C_{d,s,0}$ and $V_{d,0}$ were positively correlated to $C_{f,a}/C_{f,a,0}$ on both positive and negative sides where the highest degrees of sensitivity was obtained by changing $C_{d,s,0}$. With the negative correlation, reducing $C_{f,a,0}$ began to dramatically increase $C_{f,a}/C_{f,a,0}$ whereas the percentage change of $C_{f,a,0}$ was less than -80 % of lactic acid, but, in case of acetic, $C_{f,a}/C_{f,a,0}$ was prone to increase rapidly, once $C_{f,a,0}$ began to decrease. Nevertheless, at -100% of change in $C_{f,a,0}$ of both acids, $C_{f,a}/C_{f,a,0}$ approached to infinity. Because $C_{f,a,0}$ becomes zero, $C_{f,a}/C_{f,a,0}$ is certainly infinity. Last independent variable is $V_{f,0}$ whose the correlation to $C_{f,a}/C_{f,a,0}$ was conversed from negative to positive when percentages of decreasing raised up to around 20% of acetic acid and 60% of lactic acid. This configuration leaded to the concave downward

shape of curve. At inflection points of both curves of $V_{f,0}$, $C_{f,a}/C_{f,a,0}$ could be maximally yielded at 6.2% of acetic acid and 148% of lactic acid.

The sensitivity analysis can be practically applied as a guideline to design and modify a comparable FO system so that the optimal result can be accomplished in terms of performance and cost-effectiveness. The sensitivity analysis depicts not only the effect of changing independent variables on system performance, defined by dependent variables, but also the cost of construction and modification system because some independent variables, which are $C_{d,s,0}$, $V_{d,0}$ and $V_{f,0}$, are associated with cost. During design phase, wastewater influent, rejection rate and concentration performance are basically specified as system requirement. Regarding system performance, the sensitivity analysis on $C_{f,a,0}$, defining as wastewater concentration can help designers to recognize how the system suits to the influent concentration. Designers can increase rejection rate by increasing $C_{d,s,0}$, $V_{d,0}$ and $V_{f,0}$, represented concentration of draw solution, volume of draw solution and wastewater respectively. Inevitably, cost of designated system is also increased in different level, depending on selected draw solution properties and designs of both solution containers. Based on sensitivity analysis for concentration performance ($C_{f,a}/C_{f,a,0}$ ratio), designers can simply raise this value by increasing $C_{d,s,0}$ and $V_{d,0}$. However these subsequently create extra cost, as described previously. Without additional cost, decreasing $V_{f,0}$ similarly can increase $C_{f,a}/C_{f,a,0}$ ratio whereas the optimum value of both acids, which system can attain, are evidently shown in the sensitivity charts at the inflection points of $V_{f,0}$ curves. Besides facilitating design of new system, the sensitivity analysis can also assist designers to modify any pertinent existing system in more systematic and effective manners. Based on sensitivity curves of relevant independent variables, designers can reassign related system parameters, which are volume of containers for wastewater

and draw solution as well as concentration of draw solution, to result in the optimal

system performance with the least cost of modification.

Nomenclature

A	water permeability coefficient of the membrane
A_M	effective membrane area
B_a	acid permeability coefficient of the membrane
B_s	salt permeability coefficient of the membrane
C	salt concentration in the boundary layer
C_a	acid concentration in the boundary layer
$C_{d,a}$	acid concentration in bulk draw solution
$C_{d,b}$	bulk draw solution concentration
$C_{d,ai}$	acid concentration into the inlet of draw solution channel
$C_{d,b}$	bulk draw solution concentration
C_{i,H^+}	hydrogen ion concentration at support layer-active layer interface
$C_{d,s}$	salt concentration in bulk draw solution
$C_{d,s2}$	salt concentration from the outlet of draw solution channel
$C_{d,s,0}$	initial salt concentration in draw solution tank,
$C_{f,a}$	acid concentration in feed solution
$C_{f,b}$	bulk feed solution concentration
C_{f,A^-}	acid ion concentration in feed solution
$C_{f,a1}$	acid concentration from the outlet of feed solution channel
$C_{f,a,0}$	initial acid concentration in feed solution tank
$C_{f,b}$	bulk feed solution concentration
C_{f,Cl^-}	chloride ion concentration in feed solution
C_{f,H^+}	hydrogen ion concentration in feed solution
C_{f,NH_4^+}	ammonium ion concentration in feed solution
C_{f,OH^-}	hydroxide ion concentration in feed solution
$C_{f,s}$	salt concentration in bulk feed solution

$C_{f,si}$	salt concentration into the inlet of feed solution channel
$C_{i,a}$	acid concentration at support layer-active layer interface
$C_{i,s}$	salt concentration at support layer-active layer interface
$C_{md,a}$	acid concentration at support layer surface
$C_{md,s}$	salt concentration at support layer surface
$C_{mf,a}$	acid concentration at active layer surface
C_{mf,H^+}	hydrogen ion concentration at active layer surface
$C_{mf,s}$	salt concentration at active layer surface
C_s	salt concentration in membrane support layer
CV	coefficient of variation
D	channel depth of FO test cell
D_a	apparent weak acid diffusion coefficient
D_{A^-}	acid ion diffusion coefficient at 298 kelvins
D_{Cl^-}	chloride ion diffusion coefficient at 298 kelvins
$D_{d,s}$	average salt diffusion coefficient in draw solution
D_{eff}	effective salt diffusion coefficient in porous support layer
$D_{f,a}$	apparent weak acid diffusion coefficient in feed solution
$D_{f,s}$	average salt diffusion coefficient in feed solution
d_h	hydraulic diameter
D_{H^+}	hydrogen ion diffusion coefficient at 298 kelvins
D_{HA}	acid molecule diffusion coefficient at 298 kelvins
$D_{NH_4^+}$	ammonium ion diffusion coefficient at 298 kelvins
D_s	salt diffusion coefficient
f	flow rate of recirculating pump
f_1	flow rate from the outlet of feed solution channel
f_2	flow rate from the outlet of draw solution channel
J_a	acid flux
J_s	reverse salt flux
$J_{s,l}$	reverse salt flux in stage l
J_w	water flux
$J_{w,l}$	water flux in stage l

k	mass transfer coefficient
K_a	ionization constant
k_d	salt mass transfer coefficient at draw solution side
$k_{f,a}$	acid mass transfer coefficient at feed solution side
$k_{f,s}$	salt mass transfer coefficient at the feed solution side
$K_{NH_4^+}$	NH_4Cl ionization constant
K_w	water ionization constant
l	actual thickness of support layer
L	channel length of FO test cell
l_{eff}	effective thickness of support layer
n	total number of data
n_s	number of salt ion species
pH_f	pH of feed solution
R	gas constant
R^2	coefficient of determination
$R^2 - J_s$	coefficient of determination of salt flux
$R^2 - J_w$	coefficient of determination of water flux
R_a	acid rejection
Re_d	Reynolds number in draw solution channel
Re_f	Reynolds number in feed solution channel
$RMSE$	root mean square error
S	structure parameter of membrane support layer
Sc_d	Schmidt number of salt at draw solution side
$Sc_{f,a}$	Schmidt number of acid at feed solution side
$Sc_{f,s}$	Schmidt number of salt at feed solution side
SEP	standard error of prediction
Sh_d	Sherwood number of salt in open channel at draw solution side
$Sh_{f,a}$	Sherwood number of acid in open channel at feed solution side
$Sh_{f,s}$	Sherwood number of salt in open channel at feed solution side
t	elapsed time
T	absolute temperature

v_d	average flow velocity in the draw solution channel
$V_{d,0}$	initial volume of draw solution
v_f	average flow velocity in feed solution channel
$V_{f,0}$	initial volume of feed solution
W	channel width of FO test cell
W_d	weight change of draw solution

Greek symbols

δ	boundary layer thickness
ε	support layer porosity
μ_d	dynamic viscosity of NH$_4$Cl solution
μ_f	dynamic viscosity of feed solution
$\pi_{d,a}$	osmotic pressure of acid draw solution
$\pi_{d,b}$	osmotic pressure of bulk draw solution
$\pi_{f,a}$	osmotic pressure of acid feed solution
$\pi_{f,b}$	osmotic pressure of bulk feed solution
π_i	osmotic pressure at interface of support layer-active layer
π_{mf}	osmotic pressure at the active layer surface
ρ_d	density of NH$_4$Cl solution
ρ_f	density of feed solution
ρ_w	water density
σ	reflection coefficient
τ	tortuosity of support layer
ϑ_T	water viscosity
$\emptyset_a$	osmotic coefficient of acid solutions
$\emptyset_s$	osmotic coefficient of salt solutions

Subscripts

0	initial condition

1	outlet of feed solution channel
2	outlet of draw solution channel
a	acid species
A^-	acid ion species
$a1$	acid species from the outlet of feed solution channel
ai	acid species into the inlet of draw solution channel
b	bulk solution
Cl^-	chloride ion species
d	draw solution
eff	effective
f	feed solution
h	hydraulic
H^+	hydrogen ion species
i	membrane support layer- active layer interface
ion	ion species
l	number of experiment stages
M	membrane
md	membrane support layer surface
mf	membrane active layer surface
NH_4^+	ammonium ion species
OH^-	hydroxide ion species
s	salt species
si	salt species into the inlet of feed solution channel
$s2$	salt species from the outlet of feed solution channel
T	absolute temperature
w	water species

4.2 Mixture of two carboxylic acids as feed solution and NH$_4$Cl as draw solution

4.2.1 FO process modeling of a mixture of two carboxylic acids

Fig. 4.9 shows salt and two carboxylic acid concentration gradients across a semi-permeable membrane and indicates water transport, acid flux and salt flux operating in FO mode, configured by arranging dense selective active layer of FO membrane against acid solution (feed solution), and the porous support layer against salt solution (draw solution).
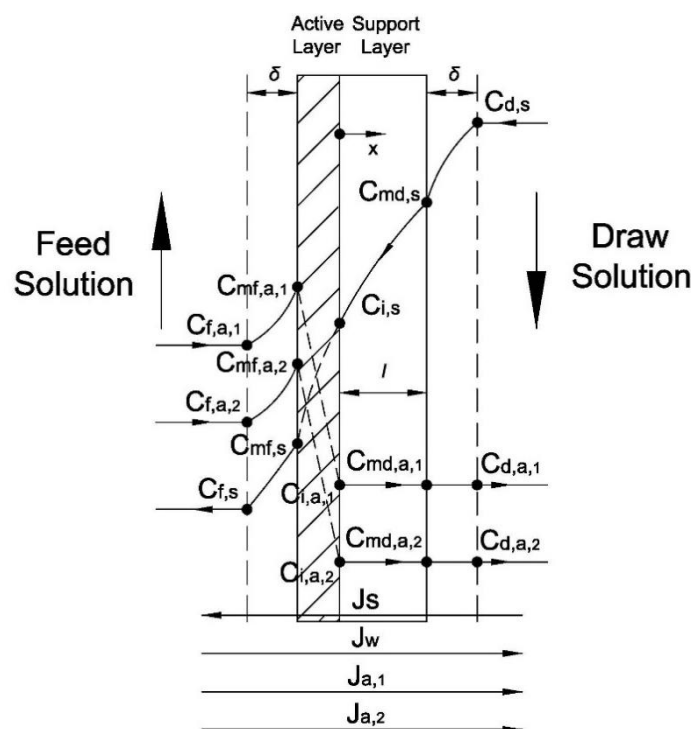

Fig. 4.9 Schematic of semi-permeable membrane cross section for the concentration of two acid species by FO process.

4.2.1.1 Permeate water permeability

The water flux, J_w in a Fig. 4.9 depends on the different osmotic pressure across the membrane active layer and can be expressed as

$$J_w = \sigma A \left(\pi_i - \pi_{mf} \right)$$

(4.2.1)

where A is the water permeability coefficient of the membrane, π_i is the osmotic pressure at interface of support layer-active layer, π_{mf} is the osmotic pressure at the active layer surface and the reflection coefficient, σ, is equal to 4.2.1.

4.2.1.2 Salt permeability

The salt flux, J_s, is related to the salt concentration gradient across the membrane active layer and can be expressed as

$$J_s = B_s(C_{i,s} - C_{mf,s})$$
(4.2.2)

where B_s is the salt permeability coefficient of the membrane, $C_{i,s}$ is the salt concentration at support layer-active layer interface and $C_{mf,s}$ is the salt concentration at active layer surface.

4.2.1.3 Acid permeability

The two acid flux, $J_{a,1}$ and $J_{a,2}$, are simulated from feed solution side, depending on their concentration gradients across the membrane active layer and can be expressed as

$$J_{a,1} = B_{a,1}(C_{mf,a,1} - C_{i,a,1})$$
(4.2.3)

$$J_{a,2} = B_{a,2}(C_{mf,a,2} - C_{i,a,2})$$
(4.2.4)

where $B_{a,1}$ and $B_{a,2}$ are the acid permeability coefficients of the membrane of acid type 1 and type 2 respectively, $C_{mf,a,1}$ and $C_{mf,a,2}$ are the acid concentration at active layer surface of acid type 1 and type 2 respectively and $C_{i,a,1}$ and $C_{i,a,2}$ are the acid concentration at support layer-active layer interface of acid type 1 and type 2 respectively.

The acid concentrations of acid type 1 and type 2 in bulk draw solution, $C_{d,a,1}$ and $C_{d,a,2}$, are given as

$$C_{d,a,1} = C_{i,a,1} = C_{md,a,1} \tag{4.2.5}$$

$$C_{d,a,2} = C_{i,a,2} = C_{md,a,2} \tag{4.2.6}$$

For acid type 1, $C_{d,a,1}$ equals to $C_{i,a,1}$ and $C_{md,a,1}$ in Eq. (4.2.5). A acid type 2 acts like a acid type 1 that $C_{d,a,1}$ equals to $C_{i,a,1}$ and $C_{md,a,1}$ in Eq. (4.2.6) where $C_{md,a,1}$ and $C_{md,a,2}$ are the acid concentrations at support layer surface of acid type 1 and acid type 2 respectively. For the reason that the acid species pass through the active layer and then enter the support and boundary layers in the same direction of permeate water, acid species thus do not experience the internal and external concentration polarizations likewise the reverse osmosis membrane. $J_{a,1}$ and $J_{a,2}$ with J_w are associated to the acid concentration in bulk draw solution by

$$C_{d,a,1} = J_{a,1}/J_w \tag{4.2.7}$$

$$C_{d,a,2} = J_{a,2}/J_w \tag{4.2.8}$$

4.2.1.4 Osmotic pressure at the support layer-active layer interface

The osmotic pressure at support layer-active layer interface, π_i, is the sum of osmotic pressures of salt and both weak acid species at the support layer-active layer interface. At equilibrium, weak acids partially ionize to acid ions and hydrogen ions. π_i can be expressed as

$$\pi_i = \emptyset_s n_s C_{i,s} RT + \emptyset_{a,1}(C_{d,a,1} + C_{d,H^+})RT + \emptyset_{a,2}(C_{d,a,2} + C_{d,H^+})RT \tag{4.2.9}$$

where $\emptyset_s$ is the osmotic coefficient of salt solutions, n_s is the number of salt ion species, R is the gas constant, T is the absolute temperature, $\emptyset_{a,1}$ and $\emptyset_{a,2}$ are the osmotic coefficients of acid solutions of acid type 1 and type 2, assumed to be equal to 1 and C_{d,H^+}, hydrogen ion concentration in bulk draw solution at equilibrium, is defined by charge balance equation as

$$C_{d,H^+} = C_{d,A^-,1} + C_{d,A^-,2} + C_{i,OH^-} + C_{i,s} - C_{i,NH_4^+} \tag{4.2.10}$$

where C_{i,NH_4^+}, ammonium chloride ion concentration at support layer-active layer interface, is defined by

$$C_{i,NH_4^+} = \frac{\gamma_{i,H^+} C_{d,H^+} C_{i,NH_3}}{\gamma_{i,NH_4^+} K_{NH_4^+,T}} \tag{4.2.11}$$

where $K_{NH_4^+,T}$ is the ionization constant of ammonium chloride, C_{i,NH_3}, ammonia concentration at support layer-active layer interface, is defined by the mass balance on nitrogen in feed solution as

$$C_{i,NH_3} = C_{i,s} - C_{i,NH_4^+} \tag{4.2.12}$$

where γ_{i,H^+} is hydrogen ion activity coefficient at support layer-active layer interface and γ_{i,NH_4^+} is ammonium ion activity coefficient at support layer-active layer interface. γ_{i,H^+} and γ_{i,NH_4^+} can be calculated from the Davies equation (Merkel et al., 2008)

$$-\log\gamma_{i,NH_4^+} = A_I Z^2 \left(\frac{I_i^{\frac{1}{2}}}{1+I_i^{\frac{1}{2}}} - 0.3I_i\right) \tag{4.2.13}$$

$$-\log\gamma_{i,H^+} = A_I Z^2 \left(\frac{I_i^{\frac{1}{2}}}{1+I_i^{\frac{1}{2}}} - 0.3I_i\right) \tag{4.2.14}$$

where A_I is the temperature-dependent constant, calculated by

$$A_I = \frac{1.82483 \times 10^6 \sqrt{d}}{(\epsilon T)^{1.5}} \tag{4.2.14-a}$$

$$d = 1 - \frac{((T-273)-3.9863)^2((T-273)+288.9414)}{508929.2 \times ((T-273)+68.12963)} + 0.011445 e^{\frac{-374.3}{(T-273)}} \tag{4.2.14-b}$$

$$\epsilon = 2727.586 + 0.6224107 \times T - 466.9151 \ln T - \frac{52000.87}{T} \tag{4.2.14-c}$$

where d is density, ϵ is dielectric constant, Z is the charge of ion and I_i, ionic strength at support layer-active layer interface, is calculated by

$$I_i = 0.5\left(C_{d,H^+} + C_{i,NH_4^+} + {} + C_{d,OH^-} + C_{i,s} + C_{d,A^-,1} + C_{d,A^-,2}\right) \tag{4.2.15}$$

where, C_{d,OH^-}, hydroxide ion concentration in bulk draw solution and determined from the equilibrium constant, can be written as

$$C_{d,OH^-} = \frac{K_{W,T}}{C_{d,H^+}\gamma_{i,H^+}\gamma_{i,OH^-}} \tag{4.2.16}$$

where $K_{W,T}$ is the water ionization constant. γ_{i,OH^-}, hydroxide ion activity coefficient at support layer-active layer interface, is determined from

$$-\log \gamma_{i,OH^-} = A_I Z^2 \left(\frac{I_i^{\frac{1}{2}}}{1+I_i^{\frac{1}{2}}} - 0.3 I_i\right) \tag{4.2.17}$$

$C_{d,A^-,1}$ and $C_{d,A^-,2}$, acid ion concentration of acid type 1 and type 2, respectively in bulk draw solution, are defined as

$$C_{d,A^-,1} = \frac{K_{a,T,1} C_{d,HA,1}}{C_{d,H^+}\gamma_{i,H^+}\gamma_{i,A^-,1}} \tag{4.2.18}$$

$$C_{d,A^-,2} = \frac{K_{a,T,2}C_{d,HA,1}}{C_{d,H^+}\gamma_{i,H^+}\gamma_{i,A^-,2}} \tag{4.2.19}$$

where $K_{a,T,1}$ and $K_{a,T,2}$ are the ionization constant of acid type 1 and type 2 respectively, $\gamma_{i,A^-,1}$ and $\gamma_{i,A^-,2}$, acid ion activity coefficient of acid type 1 and type 2 at support layer-active layer interface respectively, can be defined as

$$-\log\gamma_{i,A^-,1} = A_I Z^2\left(\frac{I_i^{\frac{1}{2}}}{1+I_i^{\frac{1}{2}}} - 0.3I_i\right) \tag{4.2.20}$$

$$-\log\gamma_{i,A^-,2} = A_I Z^2\left(\frac{I_i^{\frac{1}{2}}}{1+I_i^{\frac{1}{2}}} - 0.3I_i\right) \tag{4.2.21}$$

where $C_{d,HA,1}$ and $C_{d,HA,2}$, the acid molecule concentration of acid type 1 and type 2 in bulk draw solution respectively, are defined by the acid mass balance equation at support layer-active layer interface as

$$C_{d,HA,1} = C_{d,A^-,1} - C_{d,a,1} \tag{4.2.22}$$

$$C_{d,HA,2} = C_{d,A^-,2} - C_{d,a,2} \tag{4.2.23}$$

4.2.1.5 Osmotic pressure at the active layer surface

The osmotic pressure at the active layer surface, π_{mf}, is the sum of osmotic pressures of salt and both weak acid species at active layer surface. π_{mf} can be expressed as

$$\pi_{mf} = \emptyset_s n_s C_{mf,s}RT + \emptyset_{a,1}\left(C_{mf,a,1} + C_{mf,H^+}\right)RT + \emptyset_{a,2}\left(C_{mf,a,2} + C_{mf,H^+}\right)RT \tag{4.2.24}$$

where C_{mf,H^+}, the hydrogen ion concentration at active layer surface, is defined by charge balance equation:

$$C_{mf,H^+} = C_{mf,A,1} + C_{mf,A,2} + C_{mf,OH^-} + C_{mf,s} - C_{mf,NH_4^+} \qquad (4.2.25)$$

where C_{mf,NH_4^+}, ammonium chloride ion concentration at active layer surface, is defined by

$$C_{mf,NH_4^+} = \frac{\gamma_{mf,H^+} C_{mf,H^+} C_{mf,NH_3}}{\gamma_{mf,NH_4^+} K_{NH_4^+,T}} \qquad (4.2.26)$$

where $K_{NH_4^+,T}$ is the ionization constant of ammonium chloride, C_{mf,NH_3}, ammonia concentration at active layer surface, is defined by the nitrogen mass balance equation at active layer surface:

$$C_{mf,NH_3} = C_{mf,s} - C_{mf,NH_4^+} \qquad (4.2.27)$$

where γ_{mf,H^+} is hydrogen ion activity coefficient at active layer surface and γ_{mf,NH_4^+} is ammonium ion activity coefficient at active layer surface. γ_{mf,H^+} and γ_{mf,NH_4^+} can be calculated from

$$-\log \gamma_{mf,NH_4^+} = A_I Z^2 \left(\frac{I_{mf}^{\frac{1}{2}}}{1 + I_{mf}^{\frac{1}{2}}} - 0.3 I_{mf} \right) \qquad (4.2.28)$$

$$-\log \gamma_{mf,H^+} = A_I Z^2 \left(\frac{I_{mf}^{\frac{1}{2}}}{1 + I_{mf}^{\frac{1}{2}}} - 0.3 I_{mf} \right) \qquad (4.2.29)$$

where I_{mf}, ionic strength at active layer surface, is calculated by

$$I_{mf} = 0.5 \left(C_{mf,H^+} + C_{mf,NH_4^+} + + C_{mf,OH^-} + C_{mf,s} + C_{mf,A,1} + C_{mf,A,2} \right)$$

$$(4.114)$$

where, C_{mf,OH^-} is hydroxide ion concentration at active layer surface and determined from the equilibrium constant:

$$C_{mf,OH^-} = \frac{K_{W,T}}{C_{mf,H^+}\gamma_{mf,H^+}\gamma_{mf,OH^-}}$$
(4.2.30)

where γ_{mf,OH^-}, hydroxide ion activity coefficient at active layer surface, is determined from:

$$-\log\gamma_{mf,OH^-} = A_I Z^2 \left(\frac{I_{mf}^{\frac{1}{2}}}{1+I_{mf}^{\frac{1}{2}}} - 0.3I_{mf}\right)$$
(4.2.31)

$C_{mf,A^-,1}$ and $C_{mf,A^-,2}$, acid ion concentration of acid type 1 and type 2 respectively at active layer surface, are defined as

$$C_{mf,A^-,1} = \frac{K_{a,T,1}C_{mf,HA,1}}{C_{mf,H^+}\gamma_{mf,H^+}\gamma_{mf,A^-,1}}$$
(4.2.32)

$$C_{mf,A^-,2} = \frac{K_{a,T,2}C_{mf,HA,1}}{C_{mf,H^+}\gamma_{mf,H^+}\gamma_{mf,A^-,2}}$$
(4.2.33)

where $K_{a,T,1}$ and $K_{a,T,2}$ are the ionization constant of acid type 1 and type 2 respectively, $\gamma_{mf,A^-,1}$ and $\gamma_{mf,A^-,2}$, acid ion activity coefficients of acid type 1 and type 2 at active layer surface respectively, are calculated from

$$-\log\gamma_{mf,A^-,1} = A_I Z^2 \left(\frac{I_{mf}^{\frac{1}{2}}}{1+I_{mf}^{\frac{1}{2}}} - 0.3I_{mf}\right)$$
(4.2.34)

$$-\log\gamma_{mf,A^-,2} = A_I Z^2 \left(\frac{I_{mf}^{\frac{1}{2}}}{1+I_{mf}^{\frac{1}{2}}} - 0.3I_{mf}\right)$$
(4.2.35)

where $C_{mf,HA,1}$ and $C_{mf,HA,2}$, the acid molecule concentration at active layer surface of acid type 1 and type 2 respectively, are defined by the acid mass balance equation at active layer surface as follows:

$$C_{mf,HA,1} = C_{mf,a,1} - C_{mf,A^-,1} \tag{4.2.36}$$

$$C_{mf,HA,2} = C_{mf,a,2} - C_{mf,A^-,2} \tag{4.2.37}$$

4.2.1.6 Dilutive internal concentration polarization

The internal concentration polarization (ICP) of salt species inside the membrane support layer can be expressed as

$$-J_s = J_w C_s - \frac{d D_{eff} C_s}{dx} \tag{4.2.38}$$

where C_s is the salt concentration in membrane support layer and D_{eff}, the effective salt diffusion coefficient in porous support layer, is related to the salt diffusion coefficient, D_s, by $D_{eff} = \varepsilon D_s$ where ε is membrane support layer porosity. Integrating Eq. (4.2.38) across the membrane support layer thickness
at $x = 0, C_s = C_{i,s}$ and $x = l_{eff} = \tau l, C_s = C_{md,s}$
where $C_{i,s}$ is the salt concentration at support layer-active layer interface, $C_{md,s}$ is the salt concentration at support layer surface, l_{eff} is the effective thickness of support layer, τ is the tortuosity of support layer and l is the actual thickness of support layer yields

$$S = \int_{C_{i,s}}^{C_{md,s}} \frac{1}{(J_w C_s + J_s)} \, d(D_{d,s} C_s) \tag{4.2.38-a}$$

where S is the structure parameter of membrane support layer, defined by $S = \frac{\tau l}{\varepsilon}$ and $D_{d,s}$, the average NH$_4$Cl diffusion coefficient in draw solution, is calculated by

$$D_{d,s} = \frac{2}{(\frac{1}{D_{NH_4^+,T}} + \frac{1}{D_{Cl^-,T}})}$$
(4.2.38-b)

where $D_{NH_4^+,T}$ and $D_{Cl^-,T}$ are the ammonium ion and chloride ion diffusion coefficient respectively.

4.2.1.7 Dilutive external concentration polarization and the salt mass transfer coefficient at draw solution side

The result of convective permeate water flux from the feed solution side, reduces the salt concentration away from membrane support layer surface at the draw solution side. The dilutive external concentration polarization (ECP) can be expressed by

$$-J_s = J_w C - \frac{dD_{d,s}C}{dx}$$
(4.2.39)

Integrating Eq. (4.2.39) across the boundary layer thickness (δ)
at $x = 0, C = C_{md,s}$ and $x = \delta, C = C_{d,s}$
where $C_{d,s}$ is the salt concentration in bulk draw solution yields

$$\delta = \frac{D_{d,s}}{k_d} = \int_{C_{md,s}}^{C_{d,s}} \frac{1}{(J_w C + J_s)} d(D_{d,s}C)$$
(4.2.39-a)

where k_d is the salt mass transfer coefficient at draw solution side.

For flowing in rectangular channel for the appropriate flow regimes in open channel at draw solution side, the sherwood number of salt, Sh_d, can be represents by:

Laminar flow $(Re_d \leq 2,100)$:

$$Sh_d = 1.85(Re_d Sc_a \frac{d_h}{L})^{1/3}$$
(4.2.40-a)

Turbulent flow $(Re_d > 2,100)$:

$$Sh_d = 0.04 Re_d^{3/4} Sc_d^{1/3}$$
(4.2.41-b)

where d_h is the hydraulic diameter and Re_d, the Reynolds number in draw solution channel, is defined by

$$Re_d = \frac{Lv_d\rho_d}{\mu_d} \tag{4.2.42}$$

Where L is the channel length of FO test cell, v_d is the average flow velocity in draw solution channel, ρ_d is the density of NH_4Cl solution as a function of its molar concentration and absolute temperature and calculated by

$$\rho_d = \rho_w + 0.2061{\times}10^2 C_{d,s} - 0.1577C_{d,s}(T-273) + 1.553{\times}$$
$$10^{-3}C_{d,s}(T-273)^2 - 2.556C_{d,s}^{1.5} + 5.67{\times}10^{-2}C_{d,s}^{1.5}(T-273) -$$
$$5.082{\times}10^{-4}C_{d,s}^{1.5}(T-273)^2 \tag{4.2.43}$$

where ρ_w is the water density, according to the empirical equation Eq. (20-a):

$$\rho_w = 999.65 + 2.0438{\times}10^{-1}(T-273) - 6.174{\times}10^{-2}(T-273)^{3/2} \tag{4.2.43-a}$$

where μ_d, the dynamic viscosity of NH_4Cl solution as a function of its molar concentration, density and absolute temperature, is calculated by

$$\mu_d = 6e^{\left(\frac{12.396(53.491C_{d,s}/\rho_d)^{1.5039}-1.7756)}{(0.23471(T-273)+1)\left(-2.7591(53.491C_{d,s}/\rho_d)^{2.8408}+1\right)}\right)} \tag{4.2.44}$$

where Sc_d, Schmidt number of salt at draw solution side, is defined by

$$Sc_d = \frac{\mu_d}{\rho_d D_{d,s}} \tag{4.2.45}$$

The salt mass transfer coefficient at draw solution side, k_d, is related to Sh_d by

$$k_d = \frac{Sh_d D_{d,s}}{d_h} \tag{4.2.46}$$

4.2.1.8 Concentrative concentration polarization and acid mass transfer coefficient at feed solution side

The active layer of the semi-permeable membrane holds acid species as the water molecules permeate across the membrane. Therefore, the solute concentration is built up in the boundary layer on the active layer surface. The concentrative concentration polarization can be expressed by

$$J_{a,1} = J_w C_{a,1} + \frac{dD_{f,a,1}C_{a,1}}{dx}$$ (4.2.47)

$$J_{a,2} = J_w C_{a,2} + \frac{dD_{f,a,2}C_{a,2}}{dx}$$ (4.2.48)

where $C_{a,1}$ and $C_{a,1}$ are the acid concentration in the boundary layer of acid type 1 and type 2 respectively.

Integrating Eq. (4.2.47) and Eq. (4.2.48) across the boundary layer thickness (δ)

at $x = 0, C_{a,1} = C_{mf,a,1}$ and $x = -\delta, C_{a,1} = C_{f,a,1}$

at $x = 0, C_{a,2} = C_{mf,a,2}$ and $x = -\delta, C_{a,2} = C_{f,a,2}$

where $C_{mf,a,1}$ and $C_{mf,a,2}$ are the acid concentrations at active layer surface of acid type 1 and type 2 respectively and $C_{f,a,1}$ and $C_{f,a,2}$ are the acid concentrations in bulk feed solution of acid type 1 and type 2 respectively, yields

$$\frac{D_{f,a,1}}{k_{f,a,1}} = \int_{C_{mf,a,1}}^{C_{f,a,1}} \frac{1}{(J_w C_{a,1} - J_{a,1})} \, d\left(D_{f,a,1}C_{a,1}\right)$$ (4.2.47-a)

$$\frac{D_{f,a,2}}{k_{f,a,2}} = \int_{C_{mf,a,2}}^{C_{f,a,2}} \frac{1}{(J_w C_{a,2} - J_{a,2})} \, d\left(D_{f,a,2}C_{a,2}\right)$$ (4.2.48-a)

where $k_{f,a,1}$ and $k_{f,a,2}$, the acid mass transfer coefficient at feed solution side of acid type 1 and type 2 respectively, are defined by

$$k_{f,a,1} = \frac{Sh_{f,a,1}D_{f,a,1}}{d_h}$$ (4.2.49)

$$k_{f,a,2} = \frac{Sh_{f,a,2} D_{f,a,2}}{d_h} \tag{4.2.50}$$

where $D_{f,a,1}$ and $D_{f,a,2}$, the apparent weak acid diffusion coefficient in feed solution of acid type 1 and type 2 respectively, are calculated by

$$D_{f,a,1} = \left(\frac{D_{ion,T,1}}{2K_1 C_{f,a,1}} \left(-1 + \sqrt{1 + 4KC_{f,a,1}} \right) + \frac{D_{HA,T,1}}{4K_1 C_{f,a,1}} \left(-1 + \sqrt{1 + 4KC_{f,a,1}} \right)^2 \right)$$

$$\tag{4.2.51}$$

$$D_{f,a,2} = \left(\frac{D_{ion,T,2}}{2K_1 C_{f,a,2}} \left(-1 + \sqrt{1 + 4KC_{f,a,2}} \right) + \frac{D_{HA,T,2}}{4K_1 C_{f,a,2}} \left(-1 + \sqrt{1 + 4KC_{f,a,2}} \right)^2 \right)$$

$$\tag{4.2.52}$$

where $D_{ion,T,1}$ and $D_{ion,T,2}$, the average acid ion diffusion coefficients of acid type 1 and type 2 respectively, are defined by

$$D_{ion,T,1} = \frac{2}{\left(\frac{1}{D_{H^+,T}} + \frac{1}{D_{A^-,T,1}} \right)} \tag{4.2.51-a}$$

$$D_{ion,T,2} = \frac{2}{\left(\frac{1}{D_{H^+,T}} + \frac{1}{D_{A^-,T,2}} \right)} \tag{4.2.52-a}$$

where $D_{H^+,T}$ is hydrogen ion diffusion coefficient, $D_{A^-,T,1}$ and $D_{A^-,T,2}$ are acid ion diffusion coefficient of acid type 1 and type 2 respectively, K_1 equal to $1/K_{a,T,1}$, K_2 equal to $1/K_{a,T,2}$, $C_{f,a,1}$ and $C_{f,a,2}$ are the acid concentration in feed solution of acid type 1 and type 2 respectively. $D_{HA,T,1}$ and $D_{HA,T,2}$ are the acid molecule diffusion coefficient of acid type 1 and type 2 respectively.

$Sh_{f,a,1}$ and $Sh_{f,a,2}$ are Sherwood numbers of acid in open channel at feed solution side of acid type 1 and type 2 respectively. There are two different equations for different flow regimes, as follows:

Laminar flow ($Re_f \leq 2,100$):

$$Sh_{f,a,1} = 1.85(Re_f Sc_{f,a,1} \frac{d_h}{L})^{1/3} \tag{4.2.53-a}$$

$$Sh_{f,a,2} = 1.85(Re_f Sc_{f,a,2} \frac{d_h}{L})^{1/3} \tag{4.2.54-a}$$

Turbulent flow $\left(Re_f > 2,100\right)$:

$$Sh_{f,a,1} = 0.04 Re_f^{3/4} Sc_{f,a,1}^{1/3} \tag{4.2.53-b}$$

$$Sh_{f,a,2} = 0.04 Re_f^{3/4} Sc_{f,a,2}^{1/3} \tag{4.2.54-b}$$

where Re_f, Reynolds number in feed solution channel, is defined by

$$Re_f = \frac{Lv_f \rho_f}{\mu_f} \tag{4.2.55}$$

where v_f is the average flow velocity in feed solution channel, ρ_f is the density of feed solution and μ_f is the dynamic viscosity of feed solution, assumed to be equal to water viscosity (ϑ_T), according to Eq. (4.2.55-a) :

$$\vartheta_T = \frac{(T-273)+246}{(0.05594 \times (T-273)+5.2842)\times(T-273)+137.37} \tag{4.2.55-a}$$

where $Sc_{f,a,1}$ and $Sc_{f,a,2}$, Schmidt number of acid at feed solution side of acid type 1 and type 2 respectively, are defined by

$$Sc_{f,a,1} = \frac{\mu_f}{\rho_f D_{f,a,1}} \tag{4.2.56}$$

$$Sc_{f,a,2} = \frac{\mu_f}{\rho_f D_{f,a,2}} \tag{4.2.57}$$

4.2.1.9 Dilutive external concentration polarization and the salt mass transfer coefficient at feed solution side

The salt species diffuse from draw solution side and build up within the boundary layer at the membrane active layer surface causing the dilutive external concentration polarization. It can be expressed by

$$-J_s = J_w C - \frac{dD_{d,s}C}{dx}$$ (4.2.58)

Integrating Eq. (4.2.30) across the boundary layer thickness (δ)
at $x = 0, C_{mf,s}$ and $x = -\delta, C = C_{f,s}$
where $C_{mf,s}$ is salt concentration at active layer surface and $C_{f,s}$ is salt concentration in bulk solution, yields

$$-\delta = -\frac{D_{f,s}}{k_{f,s}} = \int_{C_{mf,s}}^{C_{f,s}} \frac{1}{(J_w C + J_s)} \, d(D_{f,s}C)$$ (4.2.58-a)

where $D_{f,s}$ is the average salt diffusion coefficient in feed solution, estimated to the salt diffusion coefficient in draw solution, $D_{d,s}$. $k_{f,s}$, the salt mass transfer coefficient at the feed solution side, is defined by

$$k_{f,s} = \frac{Sh_{f,s}D_{f,s}}{d_h}$$ (4.2.59)

where $Sh_{f,s}$ is Sherwood number of salt in open channel at feed solution side. There are two different equations for different flow regimes, as follows:

Laminar flow ($Re_f \le 2{,}100$):

$$Sh_{f,s} = 1.85(Re_f Sc_{f,s} \frac{d_h}{L})^{1/3}$$ (4.2.60-a)

Turbulent flow $(Re_f > 2{,}100)$:

$$Sh_{f,s} = 0.04 Re_f^{3/4} Sc_{f,s}^{1/3}$$ (4.2.60-b)

where $Sc_{f,s}$ is Schmidt number of salt at feed solution side, defined by

$$Sc_{f,s} = \frac{\mu_f}{\rho_f D_{f,s}}$$ (4.2.61)
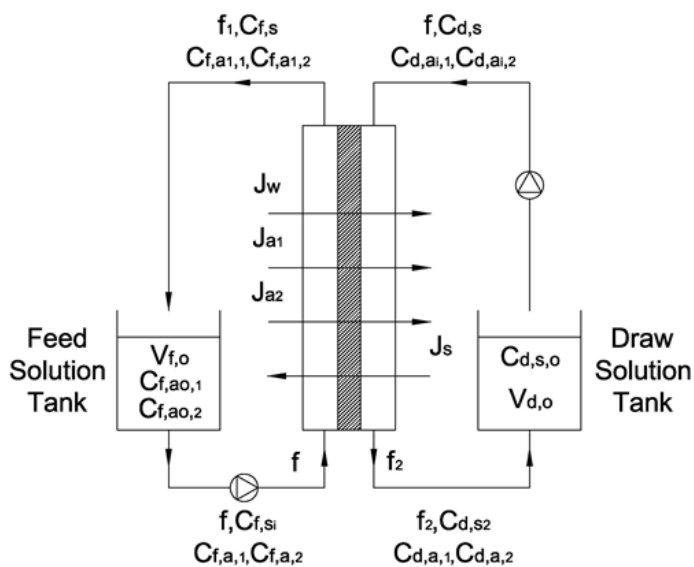
Fig. 4.10 Schematic diagram of forward osmosis system for the concentration of carboxylic acid mixture

4.2.1.10 Mole balance equation

In Fig. 4.10, the recirculating pump delivered the feed and draw solutions into the FO test cell unit installed a FO membrane piece and recycled them back to their storage tanks. The feed and draw solution in these storage tanks will be concentrated and diluted with corresponding elapsed time. Subsequently, the differential osmotic pressure across a membrane between feed and draw solution decreases and thus the water flux of the system declines in the period of experimental time.

In a Fig. 4.10, making mole balance equations on acid and salt in the feed solution tank can be written as

$$V_{f,0}C_{f,a,0,1} + C_{f,a1,1}f_1 t - C_{f,a,1}ft = C_{f,a,1}(V_{f,0}+f_1 t - ft) \qquad (4.2.62)$$

$$V_{f,0}C_{f,a,0,2} + C_{f,a1,2}f_1 t - C_{f,a,2}ft = C_{f,a,2}(V_{f,0}+f_1 t - ft) \qquad (4.2.63)$$

$$C_{f,s}f_1 t - C_{f,si}ft = C_{f,si}(V_{f,0} + f_1 t - ft) \qquad (4.2.64)$$

where $V_{f,0}$ is the initial volume of feed solution, $C_{f,a,0,1}, C_{f,a,0,2}$ are the initial acid concentrations in feed solution tank of acid type 1 and type 2 respectively, $C_{f,a1,1}$ and $C_{f,a1,2}$ are the acid concentrations from the outlet of feed solution channel of acid type 1 and type 2 respectively, f is the flow rate of recirculating pump, t is elapsed time, $C_{f,si}$ is the salt concentration into the inlet of feed solution channel and f_1, the flow rate from the outlet of feed solution channel, is determined by

$$f_1 = f - J_w A_M \qquad (4.2.65)$$

where A_M is the effective membrane area.

Making mole balance equations on acid and salt in the feed solution channel can be written as

$$C_{f,a,1}ft - J_{a,1}A_M t = C_{f,a1,1}f_1 t \qquad (4.2.66)$$

$$C_{f,a,2}ft - J_{a,2}A_M t = C_{f,a1,2}f_1 t \qquad (4.2.67)$$

$$C_{f,si}ft + J_s A_M t = C_{f,s}f_1 t \qquad (4.2.68)$$

Making mole balance equations on acid and salt in the draw solution tank can be written as

$$C_{d,a,1}f_2 t - C_{d,ai,1}ft = C_{d,ai,1}(V_{d,0} + f_2 t - ft) \qquad (4.2.69)$$

$$C_{d,a,2}f_2 t - C_{d,ai,2}ft = C_{d,ai,2}(V_{d,0} + f_2 t - ft) \qquad (4.2.70)$$

$$C_{d,s,0}V_{d,0} + C_{d,s2}f_2 t - C_{d,s}ft = C_{d,s}(V_{d,0} + f_2 t - ft) \qquad (4.2.71)$$

where $V_{d,0}$ is the initial volume of draw solution, $C_{d,s,0}$ is the initial salt concentration in draw solution tank, $C_{d,s2}$ is the salt concentration from the outlet of draw solution channel, $C_{d,ai,1}$ and $C_{d,ai,2}$ are the acid concentrations into the inlet of draw solution channel of acid type 1 and type 2 respectively and f_2, the flow rate from the outlet of draw solution channel, is determined by

$$f_2 = f + J_w A_M \tag{4.2.72}$$

Making mole balance equations on acid and salt in the draw solution channel can be written as

$$C_{d,ai,1} f t + J_{a,1} A_M t = C_{d,a,1} f_2 t \tag{4.2.73}$$

$$C_{d,ai,2} f t + J_{a,2} A_M t = C_{d,a,2} f_2 t \tag{4.2.74}$$

$$C_{d,s} f t - J_s A_M t = C_{d,s2} f_2 t \tag{4.2.75}$$

v_f, the average flow velocity in the feed solution channel, is calculated by

$$v_f = \frac{f_1 + f}{WD} \tag{4.2.76}$$

where W and D are the channel width and depth of FO test cell, respectively.

v_d, the average flow velocity in the draw solution channel, is calculated by

$$v_d = \frac{f_2 + f}{WD} \tag{4.2.77}$$

The weight change of draw solution, W_d, can be determined by

$$W_d = J_w A_M t \, \rho_d \tag{4.2.78}$$

where ρ_d is the density of NH_4Cl solution.

4.2.1.11 pH of feed solution

As feed solution was exposed to the atmosphere, open system, the carbonate species in the feed solution was in equilibrium with the CO_2 gas in the atmosphere over the solution. At ground level, the atmosphere contains $10^{-3.408}$atm of CO_2 gas (p_{CO_2}) and then equilibrates with feed solution by the following equation:

$$C_{H_2CO_3^*} \approx C_{f,CO_2} = K_H p_{CO_2} = 0.0387{\times}10^{-3.408} = 1.51{\times}10^{-5}M$$

where $K_H =$ **0.0387** molar/atm, the temperature dependence Henry's constant at 301 K and $C_{H_2CO_3^*}$ or C_{f,CO_2} is the carbonic acid concentration in feed solution. The dissociation equilibrium of diprotic carbonic acid in feed solution can be written as

$$H_2CO_3 \leftrightarrow H^+ + HCO3^-$$

$$HCO3^- \leftrightarrow H^+ + CO_3^{2-}$$
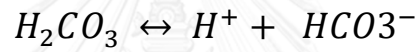
The equilibrium reactions are also established to determine C_{f,HCO_3^-} and $C_{f,CO_3^{2-}}$ by the following equations:

$$C_{f,HCO_3^-} = \frac{K_{CO_2,T}(C_{f,CO_2}-C_{f,HCO_3^-})}{C_{f,H^+}\gamma_{f,H^+}\gamma_{f,HCO3^-}}$$ (4.2.79)

$$C_{f,CO_3^{2-}} = \frac{K_{HCO_3^-,T}(C_{f,HCO_3^-}-C_{f,CO_3^{2-}})}{C_{f,H^+}\gamma_{f,H^+}\gamma_{f,CO_3^{2-}}}$$ (4.2.80)

where $K_{CO_2,T}$ and $K_{HCO_3^-,T}$ are the first ionization constant of carbonic acid, K_{a1}, and the second ionization constant of carbonic acid, K_{a2}, respectively, C_{f,HCO_3^-} is the bicarbonate ion concentration and $C_{f,CO_3^{2-}}$ is the carbonate ion concentration in feed solution. γ_{f,H^+}, $\gamma_{f,HCO3^-}$, $\gamma_{f,A^-,1}$, $\gamma_{f,A^-,2}$, γ_{f,OH^-}, $\gamma_{f,CO_3^{2-}}$ and γ_{f,NH_4^+}, hydrogen, bicarbonate, acid type 1, acid type 2, hydroxide, carbonate and ammonium ion activity coefficients in feed solution, are determined from Davis equation:

$$-\log \gamma_{f,H^+} = A_I Z^2 \left(\frac{I_f^{\frac{1}{2}}}{1+I_f^{\frac{1}{2}}} - 0.3I_f \right) \tag{4.2.81}$$

$$-\log \gamma_{f,HCO3^-} = A_I Z^2 \left(\frac{I_f^{\frac{1}{2}}}{1+I_f^{\frac{1}{2}}} - 0.3I_f \right) \tag{4.2.82}$$

$$-\log \gamma_{f,A^-,1} = A_I Z^2 \left(\frac{I_f^{\frac{1}{2}}}{1+I_f^{\frac{1}{2}}} - 0.3I_f \right) \tag{4.2.83}$$

$$-\log \gamma_{f,A^-,2} = A_I Z^2 \left(\frac{I_f^{\frac{1}{2}}}{1+I_f^{\frac{1}{2}}} - 0.3I_f \right) \tag{4.2.84}$$

$$-\log \gamma_{f,OH^-} = A_I Z^2 \left(\frac{I_f^{\frac{1}{2}}}{1+I_f^{\frac{1}{2}}} - 0.3I_f \right) \tag{4.2.85}$$

$$-\log \gamma_{f,CO_3^{2-}} = A_I Z^2 \left(\frac{I_f^{\frac{1}{2}}}{1+I_f^{\frac{1}{2}}} - 0.3I_f \right) \tag{4.2.86}$$

$$-\log \gamma_{f,NH_4^+} = A_I Z^2 \left(\frac{I_f^{\frac{1}{2}}}{1+I_f^{\frac{1}{2}}} - 0.3I_f \right) \tag{4.2.87}$$

where H^+, A^-, HCO_3^-, OH^- and NH_4^+ ion have one valence electron (Z=1), CO_3^{2-} has two valence electrons (Z=2) and I_f, the ionic strength of feed solution, is determined by

$$I_f = 0.5 \left(C_{f,Cl^-} + C_{f,H^+} + C_{f,NH_4^+} + C_{f,A^-,2} + C_{f,A^-,2} + C_{f,OH^-} + C_{f,HCO_3^-} + 4C_{f,CO_3^{2-}} \right) \tag{4.2.88}$$

where C_{f,Cl^-} is the chloride ion concentration in feed solution which is equal to $C_{f,si}$, C_{f,NH_4^+} is the ammonium ion concentration, C_{f,H^+} is the hydrogen ion concentration, C_{f,OH^-} is the hydroxide ion concentration and $C_{f,A^-,1}$ and $C_{f,A^-,2}$ is the acid ion concentration in feed solution of acid type 1 and type 2 respectively.

C_{f,NH_4^+} can be calculated from ionization constant which accounts for ionic strength and expressed by

$$C_{f,NH_4^+} = \frac{\gamma_{f,H^+} C_{f,H^+} C_{f,NH_3}}{\gamma_{f,NH_4^+} K_{NH_4^+,T}} \tag{4.2.89}$$

where $K_{NH_4^+,T}$ is the NH$_4$Cl ionization constant and C_{f,NH_3} is the ammonia concentration in feed solution.

$C_{f,A^-,1}$ and $C_{f,A^-,2}$ can also be calculated from ionization constant which accounts for ionic strength and expressed by:

$$C_{f,A^-,1} = \frac{K_{a,T,1} C_{f,HA,1}}{C_{f,H^+} \gamma_{f,H^+} \gamma_{f,A^-,1}} \tag{4.2.90}$$

$$C_{f,A^-,2} = \frac{K_{a,T,2} C_{f,HA,2}}{C_{f,H^+} \gamma_{f,H^+} \gamma_{f,A^-,2}} \tag{4.2.91}$$

where $C_{f,HA,1}$ and $C_{f,HA,2}$ is the acid molecule concentration in feed solution of acid type 1 and type 2 respectively.

C_{f,OH^-} can be determined from the water ionization constant and written as

$$C_{f,OH^-} = \frac{K_{W,T}}{C_{f,H^+} \gamma_{f,H^+} \gamma_{f,OH^-}} \tag{4.2.92}$$

where $K_{W,T}$ is the water ionization constant.

The mass balance for the conjugate acid-base pair in feed solution can be expressed by

$$C_{f,si} = C_{f,NH_4^+} + C_{f,NH_3} \qquad\qquad (4.2.93)$$

$$C_{f,a,1} = C_{f,HA,1} + C_{f,A^-,1} \qquad\qquad (4.2.94)$$

$$C_{f,a,2} = C_{f,HA,2} + C_{f,A^-,2} \qquad\qquad (4.2.95)$$

The charge balance equation in feed solution can be described by

$$C_{f,H^+} + C_{f,NH_4^+} = C_{f,A^-,1} + C_{f,A^-,2} + C_{f,OH^-} + C_{f,si} + C_{f,HCO_3^-} + 2C_{f,CO_3^{2-}}$$

$$(4.2.96)$$

The pH of feed solution, pH_f, can be calculated by

$$pH_f = -\log\left(\gamma_{f,H^+} C_{f,H^+}\right) \qquad\qquad (4.2.97)$$

4.2.1.12 List of all variables and unit in developed model

Table 4.13 Constant variables

Variable	Value	Unit	Ref.
$D_{Cl^-,298}$	2.032×10^{-5}	cm^2/sec	(Haynes, 2014-2015)
$D_{H^+,298}$	9.311×10^{-5}	cm^2/sec	(Haynes, 2014-2015)
$D_{NH_4^+,298}$	1.957×10^{-5}	cm^2/sec	(Haynes, 2014-2015)
n_s	2	$-$	-
R	0.08314	$L\ bar\ K^{-1}mol^{-1}$	-
σ	1		-
$\emptyset_s$	0.897	$-$	(Robinson and Stokes, 1959)

Table 4.14 Diffusion coefficients of carboxylic acid in water at 298 K

Acid Type	$D_{A^-,298}$ (cm^2/sec)	Ref.	$D_{a,298}$ (cm^2/sec)	Ref.	$D_{HA,298}$[a] (cm^2/sec)
Acetic acid	1.089×10^{-5}	(Bidstrup and Geankoplis, 1963)	1.27×10^{-5}	(Haynes, 2014-2015)	1.26×10^{-5}
Butyric acid	0.868×10^{-5}	(Bidstrup and Geankoplis, 1963)	0.918×10^{-5}	(Haynes, 2014-2015)	0.905×10^{-5}
Lactic acid	1.033×10^{-5}	(Bidstrup and Geankoplis, 1963)	0.993×10^{-5}	[Ribeiro et al., 2005]	0.764×10^{-5}
Valeric acid	0.871×10^{-5}	(Bidstrup and Geankoplis, 1963)	0.817×10^{-5}	(Haynes, 2014-2015)	0.80×10^{-5}

[a] $D_{HA,298}$ was calculated from Eq. (4.1.26).

Table 4.15 Fixed variables

Variable	Value	Unit	Variable	Value	Unit
A_M	0.42	dm^2	f	1.58	L/min
D	0.023	dm	T	301	K
d_h	0.0438	dm	W	0.4572	dm

Table 4.16 Initial independent variables

Variable	Value	Unit	Variable	Value	Unit
$C_{d,s,0}$	0.5	$mole/L$	$V_{d,0}$	0.5	L
$C_{f,a,0,1}$	0.01,0.005	$mole/L$	$V_{f,0}$	1	L
$C_{f,a,0,2}$	0.01,0.005	$mole/L$	t	0 to 30	hr

Table 4.17 Unknown dependent variables

Variable	Unit	Variable	Unit
1. $C_{d,a,1}$	$mole/L$	50. I_f	$mole/L$
2. $C_{d,a,2}$	$mole/L$	51. I_i	$mole/L$
3. $C_{d,A^-,1}$	$mole/L$	52. I_{mf}	$mole/L$
4. $C_{d,A^-,2}$	$mole/L$	53. $J_{a,1}$	$mole/dm^2/min$
5. $C_{d,ai,1}$	$mole/L$	54. $J_{a,2}$	$mole/dm^2/min$
6. $C_{d,ai,2}$	$mole/L$	55. J_s	$mole/dm^2/min$
7. C_{d,H^+}	$mole/L$	56. J_w	$L/dm^2/min$
8. $C_{d,HA,1}$	$mole/L$	57. k_d	dm/min

9.	$C_{d,HA,2}$	mole/L
10.	C_{d,OH^-}	mole/L
11.	$C_{d,s}$	mole/L
12.	$C_{d,s2}$	mole/L
13.	$C_{f,a,1}$	mole/L
14.	$C_{f,a,2}$	mole/L
15.	$C_{f,A^-,1}$	mole/L
16.	$C_{f,A^-,2}$	mole/L
17.	$C_{f,ai,1}$	mole/L
18.	$C_{f,ai,2}$	mole/L
19.	$C_{f,CO_3^{2-}}$	mole/L
20.	C_{f,H^+}	mole/L
21.	$C_{f,HA,1}$	mole/L
22.	$C_{f,HA,2}$	mole/L
23.	C_{f,HCO_3^-}	mole/L
24.	C_{f,NH_3}	mole/L
25.	C_{f,NH_4^+}	mole/L
26.	C_{f,OH^-}	mole/L
27.	$C_{f,s}$	mole/L
28.	$C_{f,si}$	mole/L
29.	$C_{i,a,1}$	mole/L
30.	$C_{i,a,2}$	mole/L
31.	C_{i,NH_3}	mole/L
32.	C_{i,NH_4^+}	mole/L
33.	$C_{i,s}$	mole/L
34.	$C_{md,s}$	mole/L
35.	$C_{mf,a,1}$	mole/L
36.	$C_{mf,a,2}$	mole/L
37.	$C_{mf,A^-,1}$	mole/L
38.	$C_{mf,A^-,2}$	mole/L
39.	C_{mf,H^+}	mole/L
40.	$C_{mf,HA,1}$	mole/L
41.	$C_{mf,HA,2}$	mole/L
42.	C_{mf,NH_3}	mole/L
43.	C_{mf,NH_4^+}	mole/L
44.	C_{mf,OH^-}	mole/L
45.	$C_{mf,s}$	mole/L
46.	$D_{f,a,1}$	dm^2/min
47.	$D_{f,a,2}$	dm^2/min
48.	f_1	L/min
49.	f_2	L/min

58.	$k_{f,a,1}$	dm/min
59.	$k_{f,a,2}$	dm/min
60.	$k_{f,s}$	dm^2/min
61.	pH_f	-
62.	Re_d	-
63.	Re_f	-
64.	Sc_d	-
65.	$Sc_{f,a,1}$	-
66.	$Sc_{f,a,2}$	-
67.	$Sc_{f,s}$	-
68.	Sh_d	-
69.	$Sh_{f,a,1}$	-
70.	$Sh_{f,a,2}$	-
71.	$Sh_{f,s}$	-
72.	v_d	dm/min
73.	v_f	dm/min
74.	W_d	g
75.	γ_{d,OH^-}	-
76.	$\gamma_{f,A^-,1}$	-
77.	$\gamma_{f,A^-,2}$	-
78.	$\gamma_{f,CO_3^{2-}}$	-
79.	γ_{f,H^+}	-
80.	$\gamma_{f,HCO3^-}$	-
81.	γ_{f,NH_4^+}	-
82.	γ_{f,OH^-}	-
83.	$\gamma_{i,A^-,1}$	-
84.	$\gamma_{i,A^-,2}$	-
85.	γ_{i,H^+}	-
86.	γ_{i,NH_4^+}	-
87.	$\gamma_{mf,A^-,1}$	
88.	$\gamma_{mf,A^-,2}$	-
89.	γ_{mf,H^+}	-
90.	γ_{mf,NH_4^+}	
91.	γ_{mf,OH^-}	-
92.	μ_d	g/dm/min
93.	μ_f	g/dm/min
94.	π_i	bar
95.	π_{mf}	bar
96.	ρ_d	g/L
97.	ρ_f	g/L

4.2.1.13 Temperature dependence of constant variables

The solution temperature is one of the major factors which directly influence the performance of membrane process. It has the positive correlation with permeate water flux. Thereby, the temperature dependences of constant variables, which are the equilibrium constants and the solute diffusion coefficients are provided as functions of temperature. These constant variables must be adjusted to desired operation temperature, 301K, prior to use in the model.

4.2.1.14 Equilibrium constants

The equilibrium constant is a function of temperature. As the temperature increases, the equilibrium reaction produces more ionization products. The ionization constants at the given temperature, $K_{a,T}$, can be determined by the equation in Table 4.18.

Table 4.18 Ionization constants as functions of temperature

Acid type	Equation	Reference
Acetic acid	$-\log K_{a,T} = \dfrac{1170.48}{T} - 3.1649 + 0.013399 \times T$	(Robinson and Stokes, 1959)
Butyric acid	$-\log K_{a,T} = \dfrac{1033.39}{T} - 2.6215 + 0.01334 \times T$	(Robinson and Stokes, 1959)
Lactic acid	$-\log K_{a,T} = \dfrac{1286.49}{T} - 4.8607 + 0.014776 \times T$	(Robinson and Stokes, 1959)
Valeric acid	$-\log K_{a,T} = \dfrac{921.38}{T} - 1.8574 + 0.012105 \times T$	(Robinson and Stokes, 1959)
Carbonic acid	$-\log K_{a_1,T} = \dfrac{3404.71}{T} - 14.8435 + 0.032786 \times T$	(Robinson and Stokes, 1959)
	$-\log K_{a_2,T} = \dfrac{2902.39}{T} - 6.4980 + 0.02379 \times T$	(Robinson and Stokes, 1959)
Ammonium chloride	$-\log K_{a,T} = \dfrac{2835.76}{T} - 0.6322 + 0.001225 \times T$	(Robinson and Stokes, 1959)
Water	$-\log K_{w,T} = \dfrac{4470.99}{T} - 6.0875 - 0.017060 \times T$	(Harned and Owen, 1958)

4.2.1.15 Solute diffusion coefficients

Solute diffusion coefficient and viscosity of solvent are functions of temperature. According to the well- known Stokes-Einstein equation, the temperature correction of the diffusion coefficient for any solute can be calculated by the following equations:

$$D_{H^+,T} = D_{H^+,298} \times \frac{T}{298} \times \frac{\vartheta_{298}}{\vartheta_T} \tag{4.2.98}$$

$$D_{A^-,T,1} = D_{A^-,298,1} \times \frac{T}{298} \times \frac{\vartheta_{298}}{\vartheta_T} \tag{4.2.99}$$

$$D_{A^-,T,2} = D_{A^-,298,2} \times \frac{T}{298} \times \frac{\vartheta_{298}}{\vartheta_T} \tag{4.2.100}$$

$$D_{HA,T,1} = D_{HA,298,1} \times \frac{T}{298} \times \frac{\vartheta_{298}}{\vartheta_T} \tag{4.2.101}$$

$$D_{HA,T,2} = D_{HA,298,2} \times \frac{T}{298} \times \frac{\vartheta_{298}}{\vartheta_T} \tag{4.2.102}$$

$$D_{NH_4^+,T} = D_{NH_4^+,298} \times \frac{T}{298} \times \frac{\vartheta_{298}}{\vartheta_T} \tag{4.2.103}$$

$$D_{Cl^-,T} = D_{Cl^-,298} \times \frac{T}{298} \times \frac{\vartheta_{298}}{\vartheta_T} \tag{4.2.104}$$

$$\vartheta_T = \frac{(T-273)+246}{(0.05594 \times (T-273)+5.2842) \times (T-273)+137.37} \tag{4.2.105}$$

where T is absolute temperature, ϑ_T and ϑ_{298} are water viscosity at the given temperature and at 298 kelvins, respectively.

4.2.2 Validation of the developed mathematical model
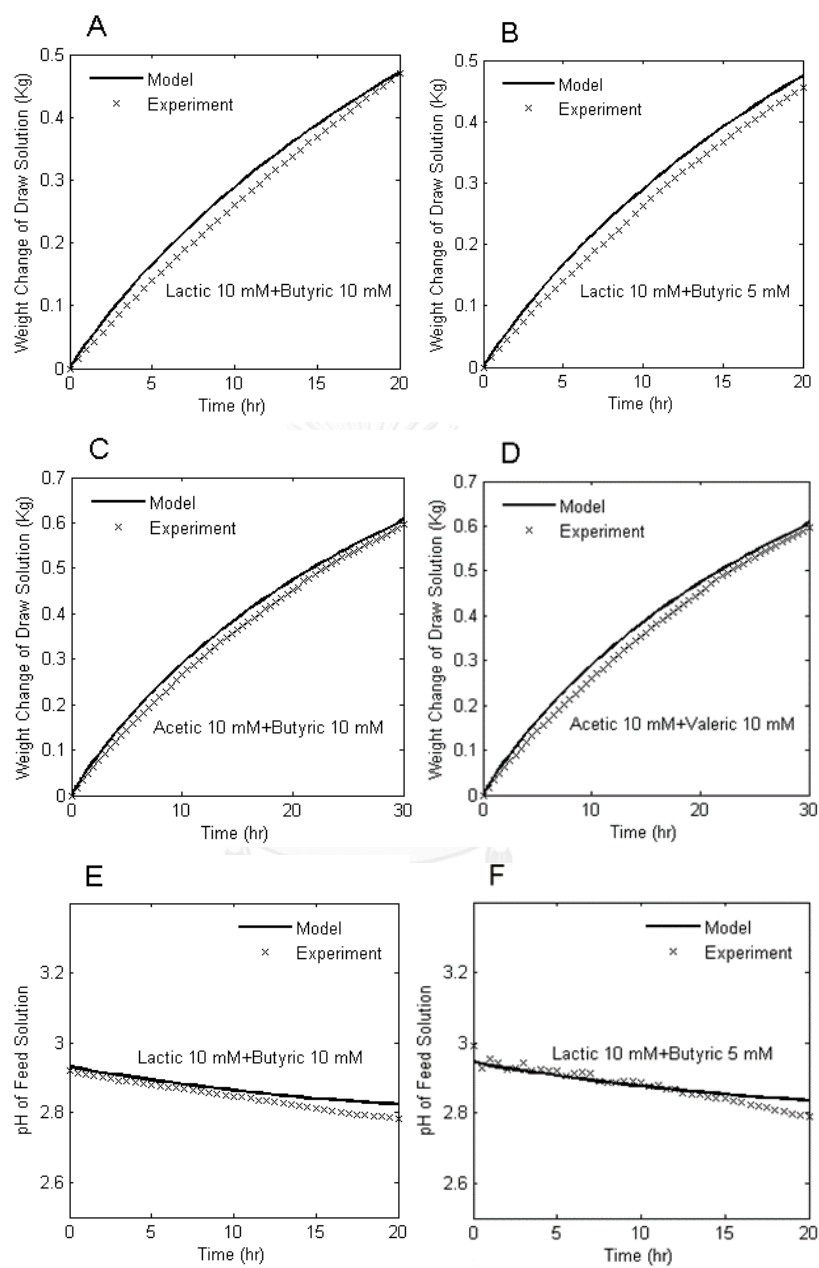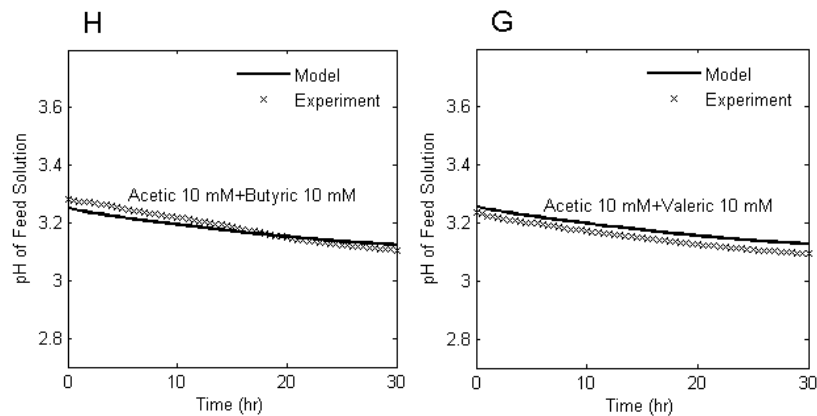
Fig. 4.11 Comparison of the simultaneous results from model predictions (solid line) and the experimental data (cross symbols) from the acid concentration FO experiment of carboxylic acid mixture, weight change of draw solution (A, B, C, D) and pH of feed solution (E, F, G, H), are plotted against elapsed time.

Each mixture of carboxylic acid model was simulated to predict the weight changes of draw solution (W_d) and pH of feed solution, as functions of time. The values of predicted and measured weight change of draw solution (Fig. 4.11A-D) and pH of feed solution (Fig.4.11E-H) were compared in the period of experimental time. In case of a mixture of lactic acid and butyric acid, to obtain the precise simulation results, the algorithm requires a great number of reiterations, leading the very lengthy simulating time for this acid mixture models. Based on the actual simulation, the simulation result at 30 hour of the model of a mixture of 10 mM lactic acid and 10 mM butyric acid can be determined by 2 months. To avoid a lengthy simulating time, the adjustment of simulation time from 30 hour to 20 hour have been introduced on the simulation of a mixture of 10 mM lactic acid and 10 mM butyric acid and 10 mM lactic acid and 5 mM butyric acid. Moreover, in case of a mixture of 5 mM lactic acid and 10 mM butyric acid as feed solution, the consumption of simulation time was much more than 3 months. However, regarding the accuracy of model prediction, the additional experiments of a mixture of 10 mM acetic acid and 10 mM butyric acid and a mixture of 10 mM acetic acid and 10 mM valeric acid as feed solution were presented to validate this acid mixture model. In Fig.4.11, it can be seen that the strong agreements between the model predictions and experiment data for each acid mixture

model were observed in an allowable error of measurement. The agreements between the simulated results from proposed model and experimental data have been evaluated by statistical factors, as given in Table 4.19.

Table 4.19 Quantitative comparisons of model predictions to experimental data

Acid Type	Weight of Draw Solution			pH of Feed Solution		
	R^2	$RMSE$	$SEP\%$	R^2	$RMSE$	$SEP\%$
Lactic 10 mM+Butyric 10 mM	0.97	0.0215	8.55	0.62	0.0236	0.83
Lactic 10 mM+Butyric 5 mM	0.97	0.0235	9.34	0.61	0.0199	0.69
Acetic 10 mM+Butyric 10 mM	0.99	0.0202	5.88	0.68	0.0215	0.68
Acetic 10 mM+Valeric 10 mM	0.98	0.0217	6.33	0.64	0.0278	0.88

4.2.3 Sensitivity analysis

The purpose of sensitivity analysis is to define the effect of independent variables on dependent variables in term of percentage of change. Varying a value of independent variable alters the dependent variable in unique manners, leaning on the relationship of each independent variable to dependent variable. The sensitivity analysis of both rejection rate and concentration performance, as dependent variables, have been directed and initiated at five initial conditions of independent variables, which are initial draw solution concentration ($C_{d,0}$ = 1.0 M), initial acetic acid concentrations (C_{f0}=10 mM), initial butyric acid concentrations (C_{f01}=10 mM), initial draw solution volume ($V_{d,0}$= 0.5 liter) and initial feed solution volume ($V_{f,0}$= 1.0 liter). After 30 hour of simulation, the sensitivity analysis results of rejection rate and concentration performance for acetic and lactic acids have been defined in sensitivity charts, as shown in Fig. 4.12 The sensitivities of both dependent variables are comparatively idiosyncratic in degree of sensitivity and correlation, hinging on acid types of feed solution and designated independent variables.
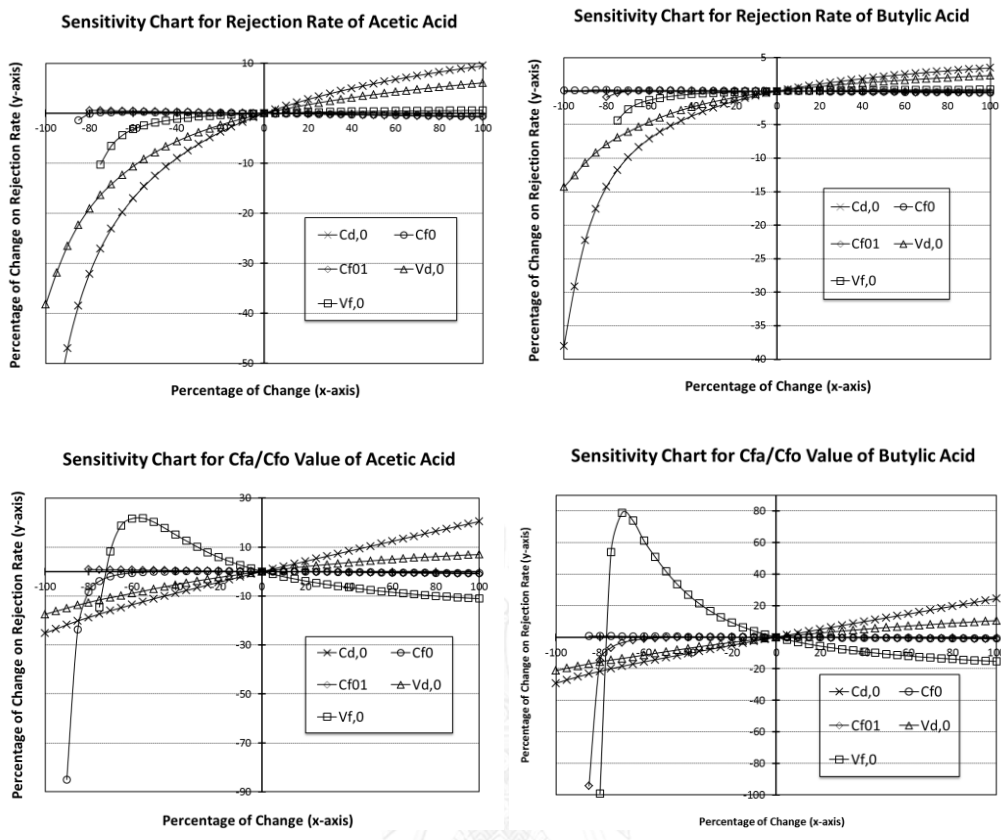
Fig. 4.12 Sensitivity charts of rejection rate and concentration performance for a mixture of 10 mM acetic acid and 10 mM butyric acid at 30 hour system operation

The sensitivity curve shows the correlation and degree of sensitivity between designated dependent and independent variables. Characterized by feed solution types, the correlation of rejection rate and concentration performance ($C_{f,a}/C_{f0}$) or ($C_{f,a}/C_{f01}$) among $C_{d,0}, C_{f0}, C_{f01}, V_{d,0}$ and $V_{f,0}$ are demonstrated in Table 4.20 and Table 4.21

Table 4.20 Correlation of rejection rate to $C_{d,0}$, C_{f0}, C_{f01}, $V_{d,0}$, and $V_{f,0}$ for each acid feed solution

Initial parameter	Acetic Acid	Butyric Acid
$C_{d,0}$	positive	positive
$C_{f,0}$	negative	negative
$C_{f,01}$	negative	negative
$V_{d,0}$	positive	positive
$V_{f,0}$	positive	positive

Table 4.21 Correlation of concentration performance to $C_{d,0}$, C_{f0}, C_{f01}, $V_{d,0}$ and $V_{f,0}$ for each acid feed solution

Initial parameter	Acetic Acid	Butyric Acid
$C_{d,0}$	positive	positive
$C_{f,0}$	negative	negative
$C_{f,01}$	negative	negative
$V_{d,0}$	positive	positive
$V_{f,0}$	negative, greater than -55%	negative, greater than -70%

In sensitivity curves, the increase and decrease rates of values of independent variables are expressed in positive and negative percentage on the x-axis while the dependent value is on the y-axis. The correlation between independent and dependent variables is defined by the slope of the curves. The positive correlation between two variables is that their values increase or decrease together. Oppositely, a negative correlation causes that one variable increases as the other decreases, and vice versa. Usually the negative correlation is advantageous because the lesser value of independent variable can competently yield the greater value of dependent variable.

The sensitivity charts of both acids are fairly alike in pattern. The sensitivity charts of rejection rate expresses that increasing rejection rate can be obtained by reducing any of C_{f0} and C_{f01} or increasing any of $C_{d,0}$, $V_{d,0}$ and $V_{f,0}$. Comparatively, the degree of sensitivity of acetic acid is greater than butyric acid due to the higher absolute slope values for each of independent variables. Considering negative correlation, $C_{d,0}$ obviously possess the highest degree of sensitivity, while $V_{f,0}$ is the lowest. At 100% of change for $C_{d,0}$ the rejection rate can be increased up to 9.52% and 3.47% for acetic acid and butyric acid respectively, but for $V_{f,0}$ the rejection rate can be rise only to 0.61% and 0.25%. Another way to increase the rejection rate is to reducing any of C_{f0} and C_{f01}. However, it causes a slim effect to the rejection rate, because at -50% of change in C_{f0} and C_{f01}, it can be increased only to 0.25% and 0.37% for acetic acid and 0.11% and 0.1% for butyric acid.

Similar to the rejection rate, the sensitivity charts for concentration performance of both acids are comparable in shape, but in term of degree of sensitivity, they are different. The degree of sensitivity of butyric acid is obviously higher than acetic acid for any of independent variables. Increasing $C_{d,0}$ for 100% can highly rise concentration performance up to 20.44% and 24.47% for acetic acid and butyric acid respectively, while, at the same percentage of change, concentration performance can be increased only to 6.94% and 10.15% by increasing $V_{d,0}$. As shown in both charts, C_{f0} and C_{f01} have a negative correlation and also a little impact on concentration performance for both acids. With concaved downward shape of curve, reducing $V_{f,0}$ can optimally raise concentration performance up to 21.81% at -55% of change for acetic acid and to 78.82% at -70% of change for butyric acid. Those values can be identified at the inflection points of the curves.

The sensitivity analysis can be basically useful as a guideline to design and modify an analogous FO system so that the optimal outcome can be obtained in terms

of performance and cost-effectiveness. The sensitivity analysis not only describes the effect of changing independent variables on system performance, expressed by dependent variables, but also on the cost of construction and modification system also since some independent variables, which are $C_{d,0}$, $V_{d,0}$ and $V_{f,0}$, are cost associated.

Throughout design phase, basically, wastewater influent, rejection rate and concentration performance must be specified as system requirement by designer. Concerning system performance, the sensitivity analysis on C_{f0} and C_{f01}, defining as wastewater concentrations, can aid designers to judge whether the designated wastewater influence is appropriate to a system or not. Designers can increase rejection rate by increasing $C_{d,0}$, $V_{d,0}$ and $V_{f,0}$, represented draw solution concentration, volume of draw solution and wastewater respectively. Certainly, cost of designated system is also increased in different level, depending on the particular draw solution properties and the designs of both solution containers. Grounded on sensitivity analysis for concentration performance, designers can simply rise this value by increasing $C_{d,0}$ and $V_{d,0}$. However these subsequently lead to extra cost, as described previously. Without additional cost, decreasing $V_{f,0}$ can also increase concentration performance whereas the optimum value is graphically shown in the sensitivity charts at the inflection points of $V_{f,0}$ curves. Besides helping design of new system, the sensitivity analysis can also assist designers to modify any related existing system in more systematic and effective way. Referred to sensitivity curves of relevant independent variables, designers can reassign some of system parameters, which are volume of containers for wastewater and draw solution as well as concentration draw solution, to optimize the system performance with the suitable cost of modification.

Nomenclature

A	water permeability coefficient of the membrane
A_M	effective membrane area
$B_{a,1}$	acid type 1 permeability coefficient of the membrane
$B_{a,2}$	acid type 2 permeability coefficient of the membrane
B_s	salt permeability coefficient of the membrane
C	salt concentration in the boundary layer
$C_{a,1}$	acid type 1 concentration in the boundary layer
$C_{a,2}$	acid type 2 concentration in the boundary layer
$C_{d,a,1}$	acid type 1 concentration in bulk draw solution
$C_{d,a,2}$	acid type 2 concentration in bulk draw solution
$C_{d,b}$	bulk draw solution concentration
$C_{d,ai,1}$	acid type 1 concentration into the inlet of draw solution channel
$C_{d,ai,2}$	acid type 2 concentration into the inlet of draw solution channel
$C_{d,b}$	bulk draw solution concentration
C_{i,H^+}	hydrogen ion concentration at support layer-active layer interface
$C_{d,s}$	salt concentration in bulk draw solution
$C_{d,s2}$	salt concentration from the outlet of draw solution channel
$C_{d,s,0}$	initial salt concentration in draw solution tank,
$C_{f,a,1}$	acid type 1 concentration in feed solution
$C_{f,a,2}$	acid type 2 concentration in feed solution
$C_{f,b}$	bulk feed solution concentration
$C_{f,A^-,1}$	acid ion type 1 concentration in feed solution
$C_{f,A^-,2}$	acid ion type 2 concentration in feed solution
$C_{f,a1,1}$	acid type 1 concentration from the outlet of feed solution channel
$C_{f,a1,2}$	acid type 2 concentration from the outlet of feed solution channel
$C_{f,a,0,1}$	initial acid type 1 concentration in feed solution tank
$C_{f,a,0,2}$	initial acid type 2 concentration in feed solution tank
$C_{f,b}$	bulk feed solution concentration
C_{f,Cl^-}	chloride ion concentration in feed solution

C_{f,H^+}	hydrogen ion concentration in feed solution
C_{f,NH_4^+}	ammonium ion concentration in feed solution
C_{f,OH^-}	hydroxide ion concentration in feed solution
$C_{f,s}$	salt concentration in bulk feed solution
$C_{f,si}$	salt concentration into the inlet of feed solution channel
$C_{i,a}$	acid concentration at support layer-active layer interface
$C_{i,s}$	salt concentration at support layer-active layer interface
$C_{md,a}$	acid concentration at support layer surface
$C_{md,s}$	salt concentration at support layer surface
$C_{mf,a,1}$	acid concentration type 1 at active layer surface
$C_{mf,a,2}$	acid concentration type 2 at active layer surface
C_{mf,H^+}	hydrogen ion concentration at active layer surface
$C_{mf,s}$	salt concentration at active layer surface
C_s	salt concentration in membrane support layer
CV	coefficient of variation
D	channel depth of FO test cell
$D_{a,1}$	apparent weak acid type 1 diffusion coefficient
$D_{a,2}$	apparent weak acid type 2 diffusion coefficient
$D_{A^-,1}$	acid ion type 1 diffusion coefficient at 298 kelvins
$D_{A^-,2}$	acid ion type 2 diffusion coefficient at 298 kelvins
D_{Cl^-}	chloride ion diffusion coefficient at 298 kelvins
$D_{d,s}$	average salt diffusion coefficient in draw solution
D_{eff}	effective salt diffusion coefficient in porous support layer
$D_{f,a,1}$	apparent weak acid type 1 diffusion coefficient in feed solution
$D_{f,a,2}$	apparent weak acid type 2 diffusion coefficient in feed solution
$D_{f,s}$	average salt diffusion coefficient in feed solution
d_h	hydraulic diameter
D_{H^+}	hydrogen ion diffusion coefficient at 298 kelvins
$D_{HA,1}$	acid molecule type 1 diffusion coefficient at 298 kelvins
$D_{HA,2}$	acid molecule type 2 diffusion coefficient at 298 kelvins
$D_{NH_4^+}$	ammonium ion diffusion coefficient at 298 kelvins

D_s	salt diffusion coefficient
f	flow rate of recirculating pump
f_1	flow rate from the outlet of feed solution channel
f_2	flow rate from the outlet of draw solution channel
$J_{a,1}$	acid flux of type 1
$J_{a,2}$	acid flux of type 2
J_s	reverse salt flux
$J_{s,l}$	reverse salt flux in stage l
J_w	water flux
$J_{w,l}$	water flux in stage l
k	mass transfer coefficient
$K_{a,1}$	ionization constant of acid type 1
$K_{a,2}$	ionization constant of acid type 2
k_d	salt mass transfer coefficient at draw solution side
$k_{f,a,1}$	acid type 1 mass transfer coefficient at feed solution side
$k_{f,a,2}$	acid type 2 mass transfer coefficient at feed solution side
$k_{f,s}$	salt mass transfer coefficient at the feed solution side
$K_{NH_4^+}$	NH_4Cl ionization constant
K_w	water ionization constant
l	actual thickness of support layer
L	channel length of FO test cell
l_{eff}	effective thickness of support layer
n	total number of data
n_s	number of salt ion species
pH_f	pH of feed solution
R	gas constant
R^2	coefficient of determination
$R^2 - J_s$	coefficient of determination of salt flux
$R^2 - J_w$	coefficient of determination of water flux
Re_d	Reynolds number in draw solution channel
Re_f	Reynolds number in feed solution channel

$RMSE$	root mean square error
S	structure parameter of membrane support layer
Sc_d	Schmidt number of salt at draw solution side
$Sc_{f,a,1}$	Schmidt number of acid type 1 at feed solution side
$Sc_{f,a,2}$	Schmidt number of acid type 2 at feed solution side
$Sc_{f,s}$	Schmidt number of salt at feed solution side
SEP	standard error of prediction
Sh_d	Sherwood number of salt in open channel at draw solution side
$Sh_{f,a,1}$	Sherwood number of acid type 1 in open channel at feed solution side
$Sh_{f,a,2}$	Sherwood number of acid type 2 in open channel at feed solution side
$Sh_{f,s}$	Sherwood number of salt in open channel at feed solution side
t	elapsed time
T	absolute temperature
v_d	average flow velocity in the draw solution channel
$V_{d,0}$	initial volume of draw solution
v_f	average flow velocity in feed solution channel
$V_{f,0}$	initial volume of feed solution
W	channel width of FO test cell
W_d	weight change of draw solution

Greek symbols

δ	boundary layer thickness
ε	support layer porosity
μ_d	dynamic viscosity of NH_4Cl solution
μ_f	dynamic viscosity of feed solution
$\pi_{d,a}$	osmotic pressure of acid draw solution
$\pi_{d,b}$	osmotic pressure of bulk draw solution
$\pi_{f,a}$	osmotic pressure of acid feed solution

$\pi_{f,b}$	osmotic pressure of bulk feed solution
π_i	osmotic pressure at interface of support layer-active layer
π_{mf}	osmotic pressure at the active layer surface
ρ_d	density of NH_4Cl solution
ρ_f	density of feed solution
ρ_w	water density
σ	reflection coefficient
τ	tortuosity of support layer
ϑ_T	water viscosity
$\emptyset_{a,1}$	osmotic coefficient of acid type 1
$\emptyset_{a,2}$	osmotic coefficient of acid type 2
$\emptyset_s$	osmotic coefficient of salt solutions

Subscripts

0	initial condition
1	outlet of feed solution channel or acid type 1
2	outlet of draw solution channel or acid type 2
a	acid species
A^-	acid ion species
$a1$	acid species from the outlet of feed solution channel
ai	acid species into the inlet of draw solution channel
b	bulk solution
Cl^-	chloride ion species
d	draw solution
eff	effective
f	feed solution
h	hydraulic
H^+	hydrogen ion species
i	membrane support layer- active layer interface
ion	ion species

l	number of experiment stages
M	membrane
md	membrane support layer surface
mf	membrane active layer surface
NH_4^+	ammonium ion species
OH^-	hydroxide ion species
s	salt species
si	salt species into the inlet of feed solution channel
$s2$	salt species from the outlet of feed solution channel
T	absolute temperature
w	water species

4.3 Single carboxylic acid and a mixture of two carboxylic acids as feed solution using NaCl as draw solution

4.3.1 FO process modeling of a single carboxylic acid

Fig. 4.13 illustrates the cross section of semi-permeable FO membrane along with carboxylic acid concentration gradient, salt concentration gradient, CO_2 concentration gradient, water flux, acid flux, salt flux and CO_2 flux. The different osmotic pressure between feed and draw solution drives the water molecules passing through a semi-permeable membrane from diluted acid, as feed solution, to highly concentrated NaCl solution, as draw solution.
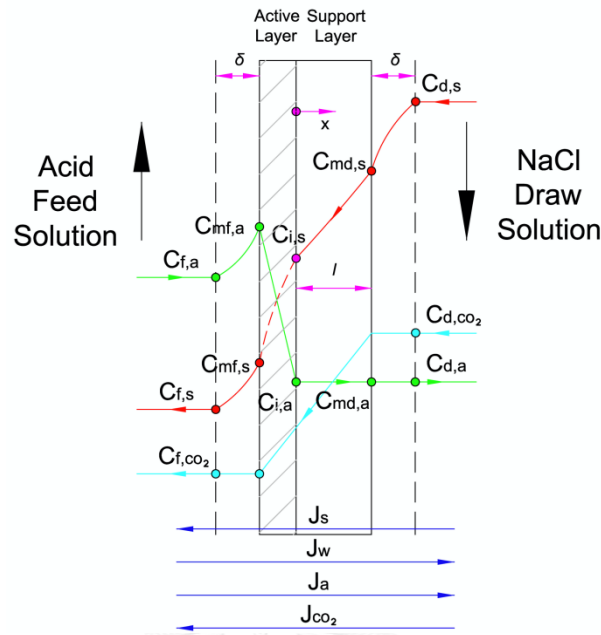
Fig. 4.13 Schematic of a cross section of a semi-permeable FO membrane indicated by water flux, acid flux, salt flux, CO2 flux and concentration gradients for acid filtration in forward osmosis.

4.3.1.1 Permeate water permeability

The water flux across the active layer in a Fig. 4.13, J_w, depends on the different osmotic pressure across the membrane active layer and can be expressed as

$$J_w = \sigma A\left(\pi_i - \pi_{mf}\right) \tag{4.3.1}$$

where A is the water permeability coefficient of the membrane, π_i is the osmotic pressure at interface of support layer-active layer, π_{mf} is the osmotic pressure at the active layer surface and the reflection coefficient, σ, is equal to 1.

4.3.1.2 Reverse salt permeability

The reverse salt flux through active layer, J_s, is proportional to salt concentration gradient across the membrane active layer and can be expressed as

$$J_s = B_s(C_{i,s} - C_{mf,s}) \qquad (4.3.2)$$

where B_s is the salt permeability coefficient of the membrane, $C_{i,s}$ is the salt concentration at support layer-active layer interface and $C_{mf,s}$ is the salt concentration at active layer surface.

    4.3.1.3 Acid permeability

The acid flux, J_a, depends on its concentration gradient across the membrane active layer and can be expressed as

$$J_a = B_a(C_{mf,a} - C_{i,a}) \qquad (4.3.3)$$

where B_a is the acid permeability coefficient of the membrane, $C_{mf,a}$ is the acid concentration at active layer surface and $C_{i,a}$ is the acid concentration at support layer-active layer interface.

The acid concentration in bulk draw solution, $C_{d,a}$, is given as

$$C_{d,a} = C_{i,a} = C_{md,a} \qquad (4.3.4)$$

$C_{d,a}$ equals to $C_{i,a}$ and $C_{md,a}$, where $C_{md,a}$ is the acid concentration at support layer surface. Since the acid species diffuse through the active layer and then pass into the support and boundary layers in the same direction of permeate water, acid species do not thus experience the internal and external concentration polarizations like the reverse osmosis membrane (Jin et al., 2011). J_a and J_w are related to the acid concentration in bulk draw solution by

$$C_{d,a} = J_a/J_w \qquad (4.3.5)$$

4.3.1.4 Osmotic pressure at the support layer-active layer interface

The osmotic pressure at support layer-active layer interface, π_i, is the sum of osmotic pressure of salt and weak acid species and can be expressed as

$$\pi_i = \emptyset_s n_s C_{i,s} RT + \emptyset_a \big(C_{i,a} + C_{i,H^+}\big) RT \tag{4.3.6}$$

where $\emptyset_s$ is the osmotic coefficient of salt solutions, n_s is the number of salt ion species, R is the gas constant, T is the absolute temperature, $\emptyset_a$, the osmotic coefficient of acid solutions, is assumed to be equal to 1 and C_{i,H^+} is the hydrogen ion concentration at support layer-active layer interface and takes into account ionic strength. C_{i,H^+} is defined by

$$C_{i,H^+} = \sqrt{\frac{K_{a,T}(C_{i,a} - C_{i,H^+})}{\gamma_{i,z1}^2}} \tag{4.3.7}$$

where $K_{a,T}$ is the acid ionization constant, $\gamma_{i,z1}$ is single charge activity coefficient of hydrogen and acid ion at support layer-active layer interface and calculated from the Davies equation. This equation is recommended for high ionic strength and proved for strong electrolyte or weak acid/base pair:

$$-\log \gamma_{i,z1} = A_I Z^2 \left(\frac{I_i^{\frac{1}{2}}}{1 + I_i^{\frac{1}{2}}} - 0.3 I_i\right) \tag{4.3.8}$$

where A_I is the temperature-dependent constant which is estimated to 0.51 , Z is the charge of ion and I_i, ionic strength at support layer-active layer interface, is calculated by

$$I_i = 0.5\big(2C_{i,s} + 2C_{i,H^+}\big) \tag{4.3.9}$$

4.3.1.5 Osmotic pressure at the active layer surface

The osmotic pressure at the active layer surface, π_{mf}, is the sum of osmotic pressure of salt and weak acid species at active layer surface and can be expressed as

$$\pi_{mf} = \emptyset_s n_s C_{mf,s} RT + \emptyset_a \left(C_{mf,a} + C_{mf,H^+} \right) RT \qquad (4.3.10)$$

where $C_{mf,s}$ is the salt concentration at active layer surface, $C_{mf,a}$ and C_{mf,H^+} are the acid and hydrogen ion concentrations at active layer surface respectively. C_{mf,H^+} is determined by

$$C_{mf,H^+} = \sqrt{\frac{K_{a,T}(C_{mf,a} - C_{mf,H^+})}{\gamma_{mf,z1}^2}} \qquad (4.3.11)$$

where $\gamma_{mf,z1}$, single charge activity coefficient of hydrogen and acid ion at active layer surface, is calculated by

$$-\log \gamma_{mf,z1} = A_I Z^2 \left(\frac{I_{mf}^{\frac{1}{2}}}{1 + I_{mf}^{\frac{1}{2}}} - 0.3 I_{mf} \right) \qquad (4.3.12)$$

where I_{mf}, ionic strength at active layer surface, is determined by

$$I_{mf} = 0.5 \left(2C_{mf,s} + 2C_{mf,H^+} \right) \qquad (4.3.13)$$

4.3.1.6 Dilutive internal concentration polarization

The internal concentration polarization (ICP) of salt species inside the membrane support layer can be expressed as

$$-J_s = J_w C_s - \frac{dD_{eff}C_s}{dx} \qquad (4.3.14)$$

where C_s is the salt concentration in membrane support layer. D_{eff} is the effective salt diffusion coefficient in porous support layer and related to the salt diffusion coefficient (D_s), by $D_{eff} = \varepsilon D_s$ where ε is membrane support layer porosity.

Integrating Eq. (4.3.14) across the membrane support layer thickness

at $x = 0, C_s = C_{i,s}$ and $x = l_{eff} = \tau l, C_s = C_{md,s}$

where $C_{i,s}$ is the salt concentration at support layer-active layer interface, $C_{md,s}$ is the salt concentration at support layer surface, l_{eff} is the effective thickness, τ is the tortuosity and l is the actual thickness of support layer, yields

$$S = \int_{C_{i,s}}^{C_{md,s}} \frac{1}{(J_w C_s + J_s)} \, d(D_{d,s} C_s) \tag{4.3.14-a}$$

where S, the structure parameter of membrane support layer, is defined by $S = \frac{\tau l}{\varepsilon}$ and $D_{d,s}$, the average NaCl diffusion coefficient as a function of its molar concentration at 28 °C, is obtained from OLI software (Appendix C) and determined by

$$D_{d,s} = 5.05 \times 10^{-11} C_s{}^4 - 2.59 \times 10^{-9} C_s{}^3 + 4.65 \times 10^{-8} C_s{}^2 +$$
$$4.03 \times 10^{-7} C_s + 1.06 \times 10^{-5} \tag{4.3.14-b}$$

4.3.1.7 Dilutive external concentration polarization and salt mass transfer coefficient at draw solution side

At support layer surface, the external concentration polarization (ECP), which is affected by dilutive permeate water from feed solution side, reduces the salt concentration away from membrane support layer surface at the draw solution side and can be expressed as

$$-J_s = J_w C - \frac{d D_{d,s} C}{dx} \tag{4.3.15}$$

where C is the salt concentration in the boundary layer.

Integrating Eq. (4.3.15) across the boundary layer thickness (δ)

at $x = 0, C = C_{md,s}$ and $x = \delta, C = C_{d,s}$

where $C_{d,s}$ is the salt concentration in bulk draw solution, yields

$$\delta = \frac{D_{d,s}}{k_d} = \int_{C_{md,s}}^{C_{d,s}} \frac{1}{(J_w C + J_s)} \, d\left(D_{d,s} C\right) \tag{4.3.15-a}$$

where k_d is the salt mass transfer coefficient at draw solution side and $D_{d,s}$ is determined by

$$D_{d,s} = 5.05 \times 10^{-11} C^4 - 2.59 \times 10^{-9} C^3 + 4.65 \times 10^{-8} C^2 + 4.03 \times 10^{-7} C + 1.06 \times 10^{-5} \tag{4.3.15-b}$$

For flowing in rectangular channel for the turbulent flow regimes $(Re_d > 2,100)$ in open channel at draw solution side, Sherwood number of salt (Sh_d) is represented by:

$$Sh_d = 0.04 Re_d^{3/4} Sc_d^{1/3} \tag{4.3.16}$$

where Re_d, Reynolds number in draw solution channel, is defined by

$$Re_d = \frac{L v_d \rho_d}{\mu_d} \tag{4.3.17}$$

where L is the channel length of FO test cell, v_d is the average flow velocity in draw solution channel, ρ_d, the density of NaCl solution as a function of its molar concentration at 28 °C, is obtained from OLI software (Appendix C) and determined by

$$\rho_d = 37.0166 C_{d,s} + 997.2911 \tag{4.3.18}$$

where μ_d, the dynamic viscosity of NaCl solution as a function of its molar concentration at 28 °C, is obtained from OLI software (Appendix E) and determined by

$$\mu_d = 0.0538C_{d,s}{}^2 + 0.4159C_{d,s} + 5.0095 \qquad (4.3.19)$$

where Sc_d, Schmidt number of salt at draw solution side, is defined by

$$Sc_d = \frac{\mu_d}{\rho_d D_{d,s}} \qquad (4.3.20)$$

where $D_{d,s}$ is determined by

$$D_{d,s} = 5.05 \times 10^{-11} C_{d,s}^4 - 2.59 \times 10^{-9} C_{d,s}^3 + 4.65 \times 10^{-8} C_{d,s}^2 +$$
$$4.03 \times 10^{-7} C_{d,s} + 1.06 \times 10^{-5} \qquad (4.3.21)$$

The salt mass transfer coefficient at draw solution side, k_d, is related to Sh_d by

$$k_d = \frac{Sh_d D_{d,s}}{d_h} \qquad (4.3.22)$$

where d_h is the hydraulic diameter

4.3.1.8 Concentrative concentration polarization and acid mass transfer coefficient at feed solution side

The selective solutes are retained by active layer as the water molecules permeate across the membrane. Thus, the solute concentration is built-up in the boundary layer on the active layer surface. The concentrative concentration polarization can be expressed as

$$J_a = J_w C_a + \frac{dD_{f,a}C_a}{dx} \qquad (4.3.23)$$

where C_a is the acid concentration in the boundary layer. Integrating Eq. (4.3.23) across the boundary layer thickness (δ)

at $x = 0, C_a = C_{mf,a}$ and $x = -\delta, C_a = C_{f,a}$

where $C_{mf,a}$ is the acid concentration at active layer surface and $C_{f,a}$ is the acid concentration in bulk feed solution, yields

$$\delta = \frac{D_{f,a}}{k_{f,a}} = \int_{C_{mf,a}}^{C_{f,a}} \frac{1}{(J_w C_a - J_a)} \, d(D_{f,a} C_a) \tag{4.3.23-a}$$

where $D_{f,a}$, the apparent weak acid diffusion coefficient in feed solution, is determined by

$$D_{f,a} = \left(\frac{D_{ion,T}}{2KC_a} \left(-1 + \sqrt{1 + 4KC_a} \right) + \frac{D_{HA,T}}{4KC_a} \left(-1 + \sqrt{1 + 4KC_a} \right)^2 \right) \tag{4.3.23-b}$$

where $D_{ion,T}$, the average acid ion diffusion coefficient, is defined by

$$D_{ion,T} = \frac{2}{\left(\frac{1}{D_{H^+,T}} + \frac{1}{D_{A^-,T}} \right)} \tag{4.3.23-c}$$

where $D_{H^+,T}$ and $D_{A^-,T}$ are hydrogen and acid ion diffusion coefficients respectively, K equals to $1/K_{a,T}$, $D_{HA,T}$ is the acid molecule diffusion coefficient and $k_{f,a}$, the acid mass transfer coefficient at feed solution side, is defined by

$$k_{f,a} = \frac{Sh_{f,a} D_{f,a}}{d_h} \tag{4.3.24}$$

where $D_{f,a}$ is determined by

$$D_{f,a} = \left(\frac{D_{ion,T}}{2KC_{f,a}} \left(-1 + \sqrt{1 + 4KC_{f,a}} \right) + \frac{D_{HA,T}}{4KC_{f,a}} \left(-1 + \sqrt{1 + 4KC_{f,a}} \right)^2 \right) \tag{4.3.25}$$

$Sh_{f,a}$, Sherwood number of acid in open channel at feed solution side, is defined by

$$Sh_{f,a} = 0.04 Re_f^{3/4} Sc_{f,a}^{1/3} \tag{4.3.26}$$

where Re_f, Reynolds number in feed solution channel, is defined by

$$Re_f = \frac{Lv_f\rho_f}{\mu_f}$$
(4.3.27)

where v_f is the average flow velocity in feed solution channel, μ_f is the dynamic viscosity of feed solution and ρ_f is the density of feed solution, assumed close to water density (ρ_w) according to the empirical equation by Eq. (4.3.27-a):

$$\rho_w = 999.65 + 2.0438{\times}10^{-1}(T - 273) - 6.174{\times}10^{-2}(T - 273)^{3/2}$$
(4.3.27-a)

$Sc_{f,a}$, Schmidt number of acid at feed solution side, is defined by

$$Sc_{f,a} = \frac{\mu_f}{\rho_f D_{f,a}}$$
(4.3.28)

4.3.1.9 Dilutive external concentration polarization and salt mass transfer coefficient at feed solution side

The salt concentration declines away from the membrane active layer surface in the boundary layer at the feed solution side. The dilutive external concentration polarization can be expressed by

$$-J_s = J_w C - \frac{dD_{f,s}C}{dx}$$
(4.3.29)

Integrating Eq. (4.3.29) across the boundary layer thickness (δ)

at $x = 0, C_{mf,s}$ and $x = -\delta, C = C_{f,s}$

where $C_{mf,s}$ is salt concentration at active layer surface and $C_{f,s}$ is salt concentration in bulk solution, yields

$$-\delta = -\frac{D_{f,s}}{k_{f,s}} = \int_{C_{mf,s}}^{C_{f,s}} \frac{1}{(J_w C + J_s)} \, d(D_{f,s}C)$$
(4.3.29-a)

where $D_{f,s}$ is the average salt diffusion coefficient in feed solution and estimated to the salt diffusion coefficient in draw solution $(D_{d,s})$. $D_{f,s}$ is determined by

$$D_{f,s} = 5.05{\times}10^{-11}C^4 - 2.59{\times}10^{-9}C^3 + 4.65{\times}10^{-8}C^2 + 4.03{\times}10^{-7}C + 1.06{\times}10^{-5}$$

(4.3.29-b)

where $k_{f,s}$, the salt mass transfer coefficient at the feed solution side, is defined by

$$k_{f,s} = \frac{Sh_{f,s}D_{f,s}}{d_h}$$

(4.3.30)

where $D_{f,s}$ is determined by

$$D_{f,s} = 5.05{\times}10^{-11}C_{f,s}^4 - 2.59{\times}10^{-9}C_{f,s}^3 + 4.65{\times}10^{-8}C_{f,s}^2 + 4.03{\times}10^{-7}C_{f,s} + 1.06{\times}10^{-5}$$

(4.3.31)

where $Sh_{f,s}$, Sherwood number of salt in open channel at feed solution side, is defined by

$$Sh_{f,s} = 0.04Re_f^{3/4}Sc_{f,s}^{1/3}$$

(4.3.32)

where $Sc_{f,s}$, Schmidt number of salt at feed solution side, is defined by

$$Sc_{f,s} = \frac{\mu_f}{\rho_f D_{f,s}}$$

(4.3.33)

4.3.1.10 CO_2 permeability
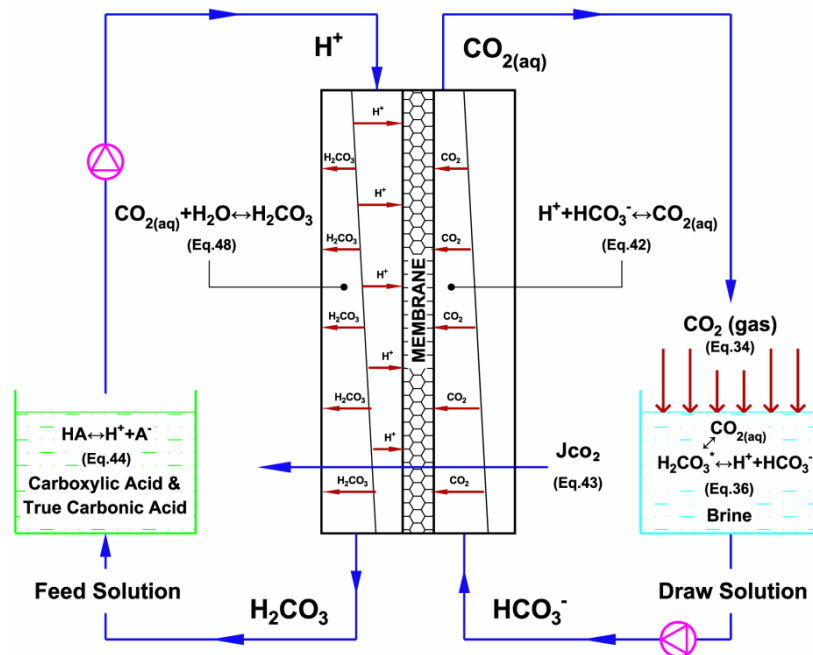
Fig. 4.14 The dynamical system of forward osmosis shows the novel CO_2 transportation mechanism through a FO membrane from NaCl draw solution side to successively generate the true carbonic acid in feed solution side. The reaction pathways of carbonate system with carboxylic acid in the FO process are presented in corresponding with mathematical equations.

4.3.1.11 Draw solution side

In Fig. 4.14, when draw solution tank is exposed to the air, the carbonate species in the NaCl draw solution is in equilibrium with atmospheric CO_2 above the solution. In the atmosphere, the CO_2 partial pressure is $10^{-3.408}$ atm (p_{CO_2}) and equilibrates with draw solution as the following equation:

$$C_{d,CO_2a} = \frac{K_H \, p_{CO_2}}{\gamma_o} \tag{4.3.34}$$

where C_{d,co_2a} is the CO_2 concentration in draw solution tank, $K_H = 0.0387$ molar, the temperature dependence Henry's constant at 298 K and γ_0, activity coefficient of

dissolved CO_2 in the NaCl draw solution as a function of ionic strength (I_d) and temperature $(t, °C)$, is calculated from the following empirical equation (butler, 1991):

$$log\gamma_o = \frac{(33.5-0.109t+0.0014t^2)I_d-(1.5+0.015t+0.004t^2)I_d^2}{t+273} \qquad (4.3.35)$$

The dissociation equilibrium of dissolved CO_2 in draw solution can be written as

$$H_2CO_3^* \leftrightarrow H^+ + HCO_3^-$$
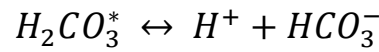
The equilibrium reaction is established to determine bicarbonate ion concentration by the following equation:

$$C_{d,HCO_3^-} = \sqrt{\frac{K_{CO_2,T}(C_{d,CO_2}-C_{d,HCO_3^-})}{\gamma_{d,z1}^2}} \qquad (4.3.36)$$

where $K_{CO_2,T}$ is the first ionization constant of carbonic acid and C_{d,HCO_3^-} is the bicarbonate ion concentration in draw solution. $\gamma_{d,z1}$, single charge activity coefficient of hydrogen, bicarbonate, acid and hydroxide ions in draw solution, is determined by

$$-\log\gamma_{d,z1} = A_I Z^2(\frac{I_d^{\frac{1}{2}}}{1+I_d^{\frac{1}{2}}} - 0.3I_d) \qquad (4.3.37)$$

where H^+, A^-, HCO_3^- and $, OH^-$ ion have a single valence electron (Z=1) and I_d, the ionic strength of draw solution, is determined by

$$I_d = 0.5\left(2C_{d,s} + C_{d,H^+} + C_{d,OH^-} + C_{d,A^-} + C_{d,HCO_3^-}\right) \qquad (4.3.38)$$

where $C_{d,s}, C_{d,H^+}, C_{d,OH^-}$ and C_{d,A^-} are salt, hydrogen ion, hydroxide ion and acid ion concentration in draw solution, respectively.

In equilibrium, weak acid donates its proton into the water and can be written in the short notation as

$$HA \leftrightarrow A^- + H^+$$

The above equilibrium reaction can determine C_{d,A^-} by Eq. (4.3.39):

$$C_{d,A^-} = \frac{K_{a,T}(C_{d,a} - C_{d,A^-})}{C_{d,H^+}\gamma_{d,z1}^2} \tag{4.3.39}$$

The self-ionization of water acts as either an acid or base as the following reaction:

$$H_2O \leftrightarrow H^+ + OH^-$$

C_{d,OH^-} can be determined from the equilibrium equation and written as

$$C_{d,OH^-} = \frac{K_{W,T}}{C_{d,H^+}\gamma_{d,z1}^2} \tag{4.3.40}$$

where $K_{W,T}$ is the water ionization constant.

The charge balance equation in draw solution can be described by

$$C_{d,H^+} = C_{d,A^-} + C_{d,OH^-} + C_{d,HCO_3^-} \tag{4.3.41}$$

In the draw solution channel of FO test cell, the bicarbonate ion species instantly react with hydrogen ions, gradually diffusing from acid feed solution channel, and then can quickly convert to soluble carbon dioxide in the aqueous solution above atmospheric by the following equilibrium reaction:

$$H^+ + HCO_3^- \leftrightarrow CO_{2(aq)} + H_2O$$
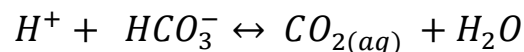
This equilibrium reaction is written to determine the concentration of carbon dioxide in draw solution channel by the following equation:

$$C_{d,CO_2} = \frac{\gamma_{d,z1}^2 C_{d,H^+}}{K_{CO_2,T}} + C_{d,HCO_3^-} \qquad (4.3.42)$$

where C_{d,CO_2} is carbon dioxide concentration, generated by acid permeability, in the draw solution channel.

The transmembrane CO_2 flux, J_{CO_2}, into feed solution channel can be determined by

$$J_{CO_2} = P_{CO_2} C_{d,CO_2} \qquad (4.3.43)$$

where P_{CO_2} is the membrane CO_2 permeability coefficient. As illustrated in Fig.4.14, the carbon dioxides partially diffuse through the membrane to feed solution channel. The rest of them flow into a draw solution tank and then disperse to the atmosphere. By means of inverse problems techniques, the membrane CO_2 permeability coefficient, which cannot directly be observed, is determined by fitting the dynamic model to the experimental pH profiles (in the section 4.2).

4.3.1.12 Feed solution side

In the feed solution tank, weak acid ionizes its proton into the water and can be written as

$$HA \leftrightarrow A^- + H^+$$
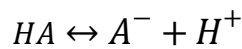
The above equilibrium reaction of weak acid can determine acid ion concentration by Eq. (4.3.44):

$$C_{f,A^-} = \frac{K_{a,T}(C_{f,a} - C_{f,A^-})}{C_{f,H^+} \gamma_{f,z1}^2} \qquad (4.3.44)$$

where C_{f,A^-} is the acid ion concentration in feed solution, $\gamma_{f,z1}$, single charge activity coefficient of hydrogen, acid, bicarbonate and hydroxide ion in feed solution, is determined from

$$-\log \gamma_{f,z1} = A_I Z^2 \left(\frac{I_f^{\frac{1}{2}}}{1+I_f^{\frac{1}{2}}} - 0.3 I_f \right) \tag{4.3.45}$$

where I_f, the ionic strength of feed solution, is determined by

$$I_f = 0.5 \left(2C_{f,si} + C_{f,H^+} + C_{f,A^-} + C_{f,OH^-} + C_{f,HCO_3^-} \right) \tag{4.3.46}$$

where C_{f,H^+} is the hydrogen ion concentration and C_{f,OH^-}, the hydroxide ion concentrations in feed solution, is determined from

$$C_{f,OH^-} = \frac{K_{W,T}}{C_{f,H^+} \gamma_{f,z1}^2} \tag{4.3.47}$$

As illustrated in Fig.4.14, the generated CO_2 in draw solution channel (C_{d,CO_2}) diffuses across a membrane and enters the feed solution channel under the increasingly acidified condition and the rising levels of CO_2 in the aqueous phase above atmospheric pressure in FO module. Under this condition, as the following chemical equilibrium, the position of equilibrium hydration reaction is moved to right side to produce the true carbonic acid in the bulk feed solution (C_{f,CO_2}) according to Lechatelier's principle.

$$CO_2 + H_2O \leftrightarrow H_2CO_3$$
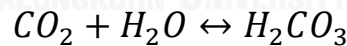
The true carbonic acid can dissociate to form bicarbonate ion (HCO_3^-) by the following chemical equilibrium:

$$H_2CO_3 \leftrightarrow H^+ + HCO_3^-$$
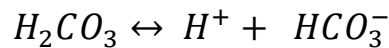
C_{f,HCO_3^-}, the bicarbonate ion concentration in feed solution tank, is determined by

$$C_{f,HCO_3^-} = \frac{K_{H_2CO_3}(C_{f,CO_2} - C_{f,HCO_3^-})}{C_{f,H^+} \gamma_{f,z1}^2} \tag{4.3.48}$$

where $K_{H_2CO_3}$ is the true carbonic acid ionization constant (Pk_a=3.45) (Adamczyk et al., 2009), C_{f,co_2} is true carbonic acid concentration and C_{f,HCO_3^-} is bicarbonate ion concentrations in feed solution.

The charge balance equation in feed solution can be described by

$$C_{f,H^+} = C_{f,A^-} + C_{f,OH^-} + C_{f,HCO_3^-} \qquad (4.3.49)$$

As validating parameter, the pH of feed solution, pH_f, is calculated by

$$pH_f = -\log\left(\gamma_{f,z1}\, C_{f,H^+}\right) \qquad (4.3.50)$$

4.3.1.13 Mole balance equations
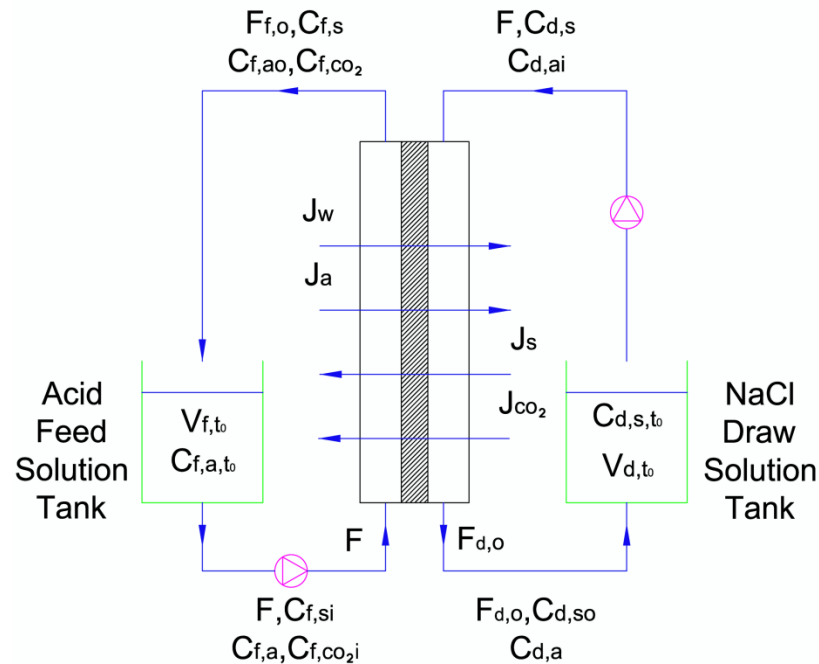
Fig. 4.15 Mole balance diagram of forward osmosis process, using carboxylic acid as feed solution and NaCl as draw solution.

In Fig. 4.15, two recirculating pumps transfer the feed and draw solutions to the inlet of FO test cell, internally installed a FO membrane. Feed and draw solution are then returned back to their storage tanks. Corresponded with elapsed time, the feed and draw solution concentration will respectively be concentrated and diluted.

Making mole balance equations on acid, salt and true carbonic acid in the feed solution tank can be written as

$$V_{f,t_0} C_{f,a,t_0} + C_{f,ao} F_{f,o} t - C_{f,a} Ft = C_{f,a}(V_{f,t_0} + F_{f,o} t - Ft) \qquad (4.3.51)$$

$$C_{f,s} F_{f,o} t - C_{f,si} Ft = C_{f,si}(V_{f,t_0} + F_{f,o} t - Ft) \qquad (4.3.52)$$

$$C_{f,co_2} F_{f,o} t - C_{f,co_2 i} Ft = C_{f,co_2 i}(V_{f,t_0} + F_{f,o} t - Ft) \qquad (4.3.53)$$

where V_{f,t_0} is the initial volume of feed solution, C_{f,a,t_0} is the initial acid concentration in feed solution tank, $C_{f,ao}$ is the acid concentration from the outlet of feed solution channel, F is the flow rate of recirculating pump, t is elapsed time, $C_{f,si}$ is the salt concentration into the inlet of feed solution channel, $C_{f,co_2 i}$ is the true carbonic acid concentration into the inlet of feed solution channel and $F_{f,o}$, the flow rate from the outlet of feed solution channel, is determined by

$$F_{f,o} = F - J_w A_M \qquad (4.3.54)$$

where A_M is the effective membrane area.

Making mole balance equations on acid, salt and true carbonic acid in the feed solution channel can be written as

$$C_{f,a} Ft - J_a A_M t = C_{f,ao} F_{f,o} t \qquad (4.3.55)$$

$$C_{f,si} Ft + J_s A_M t = C_{f,s} F_{f,o} t \qquad (4.3.56)$$

$$C_{f,co_2} Ft + J_{CO_2} A_M t = C_{f,co_2 i} F_{f,o} t \qquad (4.3.57)$$

Making mole balance equations on acid and salt in the draw solution tank can be written as

$$C_{d,a} F_{d,o} t - C_{d,ai} Ft = C_{d,ai}(V_{d,t_0} + F_{d,o} t - Ft) \qquad (4.3.58)$$

$$C_{d,s,t_0}V_{d,t_0} + C_{d,so}F_{d,o}t - C_{d,s}Ft = C_{d,s}\left(V_{d,t_0} + F_{d,o}t - Ft\right)$$

(4.3.59)

where V_{d,t_0} is the initial volume of draw solution, C_{d,s,t_0} is the initial salt concentration in draw solution tank, $C_{d,so}$ is the salt concentration from the outlet of draw solution channel, $C_{d,ai}$ is the acid concentration into the inlet of draw solution channel and $F_{d,o}$, the flow rate from the outlet of draw solution channel, is determined by

$$F_{d,o} = F + J_w A_M$$

(4.3.60)

Making mole balance equations on acid and salt in the draw solution channel can be written as

$$C_{d,ai}Ft + J_a A_M t = C_{d,a}F_{d,o}t$$

(4.3.61)

$$C_{d,s}Ft - J_s A_M t = C_{d,so}F_{d,o}t$$

(4.3.62)

v_f, the average flow velocity in the feed solution channel, is calculated by

$$v_f = \frac{F_{f,o}+F}{WD}$$

(4.3.63)

where W and D are respectively the channel width and depth of FO test cell.

v_d, the average flow velocity in the draw solution channel, is calculated by

$$v_d = \frac{F_{d,o}+F}{WD}$$

(4.3.64)

As validating parameter, the weight change of draw solution, W_d, is determined by

$$W_d = J_w A_M t \, \rho_d$$

(4.3.65)

4.3.2 FO process modeling of a mixture of two carboxylic acids
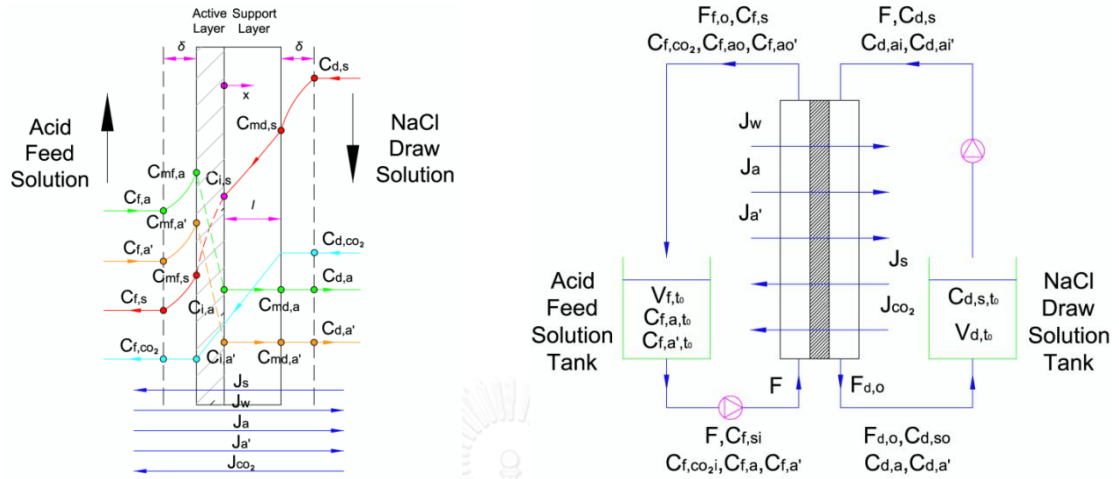
Fig. 4.16 Schematic of a FO membrane cross section during the acid filtration (left side) and mole balance diagram of forward osmosis process (right side), using a mixture of two carboxylic acids as feed solution and NaCl as draw solution.

The modeling of FO process for a single carboxylic acid as feed solution (section 4.3.1) is a part of the modeling of a mixture of two carboxylic acids. The number of sixty-five equations is alike. The ten modified equations and sixteen additional equations are added to developed model of a single carboxylic acid to acquire modeling of a mixture of two carboxylic acids. The ten modified equations are described as follows:

Equation (4.3.6) is modified to $\pi_i = \emptyset_s n_s C_{i,s} RT + \emptyset_a \big(C_{i,a} + C_{i,H^+} \big) RT + \emptyset_{a'} \big(C_{i,a'} + C_{i,H^+} \big) RT$

$$\tag{4.3.6'}$$

Equation (4.3.7) is modified to the charge balance equation: $C_{i,H^+} = C_{i,A^-} + C_{i,A^{-'}} + \dfrac{K_{W,T}}{C_{i,H^+} \gamma_{i,z1} \gamma_{i,z1}}$

$$\tag{4.3.7'}$$

Equation (4.3.9) is modified to $I_i = 0.5\left(2C_{i,s} + C_{i,A^-} + C_{i,A^{-\prime}} + C_{i,H^+} + \dfrac{K_{W,T}}{C_{i,H^+}\gamma_{i,z1}\gamma_{i,z1}}\right)$ (4.3.9')

Equation (4.3.10) is modified to $\pi_{mf} = \emptyset_s n_s C_{mf,s} RT + \emptyset_a\left(C_{mf,a} + C_{mf,H^+}\right)RT + \emptyset_{a\prime}\left(C_{mf,a\prime} + C_{mf,H^+}\right)RT$ (4.3.10')

Equation (4.3.11) is modified to $C_{mf,H^+} = C_{mf,A^-} + C_{mf,A^{-\prime}} + \dfrac{K_{W,T}}{C_{mf,H^+}\gamma_{mf,z1}\gamma_{mf,z1}}$

$$(4.3.11')$$

Equation (4.3.13) is modified to $I_{mf} = 0.5\left(2C_{mf,s} + C_{mf,A^-} + C_{mf,A^{-\prime}} + C_{mf,H^+} + \dfrac{K_{W,T}}{C_{mf,H^+}\gamma_{mf,z1}\gamma_{mf,z1}}\right)$ (4.3.13')

Equation (4.3.37) is modified to $I_d = 0.5\left(2C_{d,s} + C_{d,H^+} + C_{d,OH^-} + C_{d,A^-} + C_{d,A^{-\prime}} + C_{d,HCO_3^-}\right)$ (4.3.37')

Equation (4.3.40) is modified to $C_{d,H^+} = C_{d,A^-} + C_{d,A^{-\prime}} + C_{d,OH^-} + C_{d,HCO_3^-}$

$$(4.3.40')$$

Equation (4.3.45) is modified to $I_f = 0.5\left(2C_{f,si} + C_{f,A^-} + C_{f,A^{-\prime}} + C_{f,H^+} + \dfrac{K_{W,T}}{C_{f,H^+}\gamma_{f,z1}\gamma_{f,z1}} + C_{f,HCO_3^-}\right)$ (4.3.45')

Equation (4.3.47) is modified to $C_{f,H^+} = C_{f,A^-} + C_{f,A^{-\prime}} + \dfrac{K_{W,T}}{C_{f,H^+}\gamma_{f,z1}\gamma_{f,z1}} + C_{f,HCO_3^-}$ (4.3.47')

where the prime symbols (′) above the letter and number represent the carboxylic acid type 2.

The sixteen additional equations of carboxylic acid type 2 are described as follows:

$$J_{a'} = B_{a'}(C_{mf,a'} - C_{i,a'}) \tag{4.3.66}$$

$$C_{d,a'} = C_{i,a'} = C_{md,a'} \tag{4.3.67}$$

$$C_{d,a'} = J_{a'}/J_w \tag{4.3.68}$$

$$C_{mf,A^{-'}} = \sqrt{\frac{K_{a',T}(C_{mf,a'} - C_{mf,A^{-'}})}{\gamma_{mf,z1}\gamma_{mf,z1}}} \tag{4.3.69}$$

$$\delta = \frac{D_{f,a'}}{k_{f,a'}} = \int_{C_{mf,a'}}^{C_{f,a'}} \frac{1}{(J_w C_{a'} - J_{a'})} \, d(D_{f,a'}C_{a'}) \tag{4.3.70-a}$$

$$D_{f,a'} = \left(\frac{D_{ion',T}}{2KC_{a'}}\left(-1 + \sqrt{1 + 4KC_{a'}}\right) + \frac{D_{HA',T}}{4KC_{a'}}\left(-1 + \sqrt{1 + 4KC_{a'}}\right)^2\right) \tag{4.3.70-b}$$

$$D_{ion',T} = \frac{2}{\left(\frac{1}{D_{H^+,T}} + \frac{1}{D_{A^{-'},T}}\right)} \tag{4.3.70-c}$$

$$K' = 1/K_{a',T} \tag{4.3.70-d}$$

$$k_{f,a'} = \frac{Sh_{f,a'}D_{f,a'}}{d_h} \tag{4.3.71}$$

$$D_{f,a'} = \left(\frac{D_{ion',T}}{2K'C_{f,a'}}\left(-1 + \sqrt{1 + 4K'C_{f,a'}}\right) + \frac{D_{HA',T}}{4K'C_{f,a'}}\left(-1 + \sqrt{1 + 4K'C_{f,a'}}\right)^2\right) \tag{4.3.72}$$

$$Sc_{f,a'} = \frac{\mu_f}{\rho_f D_{f,a'}} \tag{4.3.73}$$

$$Sh_{f,a'} = 0.04 Re_f^{3/4} Sc_{f,a'}^{1/3} \tag{4.3.74}$$

$$V_{f,t_0} C_{f,a',t_0} + C_{f,ao'} F_{f,o} t - C_{f,a'} Ft = C_{f,a'}\left(V_{f,t_0} + F_{f,o} t - Ft\right) \tag{4.3.75}$$

$$C_{f,a'} Ft - J_{a'} A_M t = C_{f,ao'} F_{f,o} t \tag{4.3.76}$$

$$C_{d,a'} F_{d,t_0} t - C_{d,ai'} Ft = C_{d,ai'}\left(V_{d,t_0} + F_{d,o} t - Ft\right) \tag{4.3.77}$$

$$C_{d,ai'} Ft + J_{a'} A_M t = C_{d,a'} F_{d,o} t \tag{4.3.78}$$

4.3.2.1 List of all variables and unit used in mathematical model

Table 4.22 Constant variables

Variable	Value	Unit	Ref.
$D_{H^+,298}$	9.311×10^{-5}	cm^2/sec	(Haynes, 2014-2015)
n_s	2	—	-
R	0.08314	$L\ bar\ K^{-1} mol^{-1}$	-
σ	1		-
$\emptyset_s$	0.936	—	(Robinson and Stokes, 1959)

Table 4.23 Diffusion coefficients of carboxylic acid in water at 298 K

Acid Type	$D_{A^-,298}$ (cm^2/sec)	Ref.	$D_{a,298}$ (cm^2/sec)	Ref.	$D_{HA,298}$[a] (cm^2/sec)
Acetic acid	1.089×10^{-5}	(Bidstrup and Geankoplis, 1963)	1.27×10^{-5}	(Haynes, 2014-2015)	1.26×10^{-5}
Butyric acid	0.868×10^{-5}	(Bidstrup and Geankoplis, 1963)	0.918×10^{-5}	(Haynes, 2014-2015)	0.905×10^{-5}
Lactic acid	1.033×10^{-5}	(Bidstrup and Geankoplis, 1963)	0.993×10^{-5}	[Ribeiro et al., 2005]	0.764×10^{-5}
Valeric acid	0.871×10^{-5}	(Bidstrup and Geankoplis, 1963)	0.817×10^{-5}	(Haynes, 2014-2015)	0.80×10^{-5}

Table 4. 24 Fixed variables

Variable	Value	Unit	Variable	Value	Unit
A_M	0.42	dm^2	f	1.58	L/min
D	0.023	dm	T	301	K
d_h	0.0438	dm	W	0.4572	dm

Table 4. 25 Initial independent variables

Variable	Value	Unit	Variable	Value	Unit
$C_{d,s,0}$	0.5	$mole/L$	$V_{d,0}$	0.5	L
$C_{f,a,0}$	0.01,0.005	$mole/L$	$V_{f,0}$	1	L
$C_{f,a',0}{}^a$	0.01,0.005	$mole/L$	t	0 to 30	hr

a $C_{f,a',0}$ is initial acid concentration of acid type 2

Table 4.26 Unknown dependent variables of single carboxylic acid model

	Variable	Unit		Variable	Unit
1.	$C_{d,a}$	$mole/L$	34.	I_f	$mole/L$
2.	C_{d,A^-}	$mole/L$	35.	I_i	$mole/L$
3.	$C_{d,ai}$	$mole/L$	36.	I_{mf}	$mole/L$
4.	C_{d,co_2}	$mole/L$	37.	J_a	$mole/dm^2/min$
5.	C_{d,co_2a}	$mole/L$	38.	J_{CO_2}	$mole/dm^2/min$
6.	C_{d,H^+}	$mole/L$	39.	J_s	$mole/dm^2/min$
7.	C_{d,HCO_3^-}	$mole/L$	40.	J_w	$L/dm^2/min$
8.	C_{d,OH^-}	$mole/L$	41.	k_d	dm/min
9.	$C_{d,s}$	$mole/L$	42.	$k_{f,a}$	dm/min
10.	$C_{d,s2}$	$mole/L$	43.	$k_{f,s}$	dm^2/min
11.	$C_{f,a}$	$mole/L$	44.	pH_f	-
12.	C_{f,A^-}	$mole/L$	45.	Re_d	-
13.	$C_{f,ai}$	$mole/L$	46.	Re_f	-
14.	C_{f,CO_2}	$mole/L$	47.	Sc_d	-
15.	C_{f,CO_2i}	$mole/L$	48.	$Sc_{f,a}$	-
16.	C_{f,H^+}	$mole/L$	49.	$Sc_{f,s}$	-
17.	C_{f,HCO_3^-}	$mole/L$	50.	Sh_d	-
18.	C_{f,OH^-}	$mole/L$	51.	$Sh_{f,a}$	-
19.	$C_{f,s}$	$mole/L$	52.	$Sh_{f,s}$	-
20.	$C_{f,si}$	$mole/L$	53.	v_d	dm/min

	Variable	Unit		Variable	Unit
21.	$C_{i,a}$	$mole/L$	54.	v_f	dm/min
22.	C_{i,H^+}	$mole/L$	55.	W_d	g
23.	$C_{i,s}$	$mole/L$	56.	$\gamma_{d,z1}$	-
24.	$C_{md,s}$	$mole/L$	57.	$\gamma_{f,z1}$	-
25.	$C_{mf,a}$	$mole/L$	58.	$\gamma_{i,z1}$	-
26.	C_{mf,H^+}	$mole/L$	59.	$\gamma_{mf,z1}$	-
27.	$C_{mf,s}$	$mole/L$	60.	γ_o	-
28.	$D_{d,s}$	dm^2/min	61.	μ_d	$g/dm/min$
29.	$D_{f,a}$	dm^2/min	62.	μ_f	$g/dm/min$
30.	$D_{f,s}$	dm^2/min	63.	π_i	bar
31.	f_1	L/min	64.	π_{mf}	bar
32.	f_2	L/min	65.	ρ_d	g/L
33.	I_d	$mole/L$			

Table 4.27 Additional unknown dependent variables of the model of carboxylic acid mixture

Variable	Unit	Variable	Unit
1. $C_{d,a'}$	$mole/L$	8. $C_{mf,a'}$	$mole/L$
2. $C_{d,A^{-\prime}}$	$mole/L$	9. $D_{f,a'}$	dm^2/min
3. $C_{d,ai'}$	$mole/L$	10. $J_{a'}$	$mole/dm^2/min$
4. $C_{f,a'}$	$mole/L$	11. $k_{f,a'}$	dm/min
5. $C_{f,A^{-\prime}}$	$mole/L$	12. $Sc_{f,a'}$	-
6. $C_{f,ai'}$	$mole/L$	13. $Sh_{f,a'}$	-
7. $C_{i,a'}$	$mole/L$		

4.3.3 Temperature dependence of constant variables

4.3.3.1 Equilibrium constants

As a function of temperature, the ionization constants, $K_{a,T}$, can be determined by the equation in Table 4.28.

Table 4.28 Ionization constants as functions of temperature

Acid type	Equation	Reference
Acetic acid	$-\log K_{a,T} = \dfrac{1170.48}{T} - 3.1649 + 0.013399 \times T$	(Robinson and Stokes, 1959)
Butyric acid	$-\log K_{a,T} = \dfrac{1033.39}{T} - 2.6215 + 0.01334 \times T$	(Robinson and Stokes, 1959)
Lactic acid	$-\log K_{a,T} = \dfrac{1286.49}{T} - 4.8607 + 0.014776 \times T$	(Robinson and Stokes, 1959)
Valeric acid	$-\log K_{a,T} = \dfrac{921.38}{T} - 1.8574 + 0.012105 \times T$	(Robinson and Stokes, 1959)
Carbonic acid	$-\log K_{CO_2,T} = \dfrac{3404.71}{T} - 14.8435 + 0.032786 \times T$	(Robinson and Stokes, 1959)
Water	$-\log K_{w,T} = \dfrac{4470.99}{T} - 6.0875 - 0.017060 \times T$	(Harned and Owen, 1958)

4.3.3.2 Solute diffusion coefficients

According to the Stokes-Einstein equation, the temperature correction of the diffusion coefficient can be determined by the following equations:

$$D_{H^+,T} = D_{H^+,298} \times \frac{T}{298} \times \frac{\vartheta_{298}}{\vartheta_T} \tag{4.3.80}$$

$$D_{A^-,T} = D_{A^-,298} \times \frac{T}{298} \times \frac{\vartheta_{298}}{\vartheta_T} \tag{4.3.81}$$

$$D_{A^{-\prime},T} = D_{A^{-\prime},298} \times \frac{T}{298} \times \frac{\vartheta_{298}}{\vartheta_T} \tag{4.3.82}$$

$$D_{HA,T} = D_{HA,298} \times \frac{T}{298} \times \frac{\vartheta_{298}}{\vartheta_T} \tag{4.3.83}$$

$$D_{HA',T} = D_{HA',298} \times \frac{T}{298} \times \frac{\vartheta_{298}}{\vartheta_T} \tag{4.3.84}$$

$$\vartheta_T = \frac{(T-273)+246}{(0.05594 \times (T-273)+5.2842) \times (T-273)+137.37} \tag{4.3.85}$$

where T is absolute temperature, ϑ_T and ϑ_{298} are water viscosity at the given temperature and at 298 kelvins, respectively.

4.3.3.3 Henry's law constant

As a function of temperature, Henry's law constant is determined by

$$K_H = K_{H^\theta} \exp(\frac{-\Delta_{soln}H}{R}(\frac{1}{T} - \frac{1}{T^\theta})) \qquad (4.3.86)$$

where K_{H^θ} is the Henry's law constant at standard condition (T^θ=298.15 kelvins), 3.4×10^{-2} molar/atm, R is the gas constant and $\frac{-\Delta_{soln}H}{R}$ is the temperature dependence, 2400 Kelvins.

4.3.3.4 Model solutions

A set of simultaneous equations were given with the same number of dependent endogenous variables. The system of equations model describes the logical phenomena during the carboxylic acid filtration in forward osmosis process. By Levenberg-Marquardt (LM) algorithm running in MATLAB, the dynamic model was able to simultaneously define the unknown dependent variables at each point in simulating time. The process of model solutions is described in Fig. 4.17. The calculations were inaugurated by inputting constant variables (e.g. acid ionization constant, diffusion coefficient, gas constant), fixed variables (e.g. membrane area, test cell channel sizing, liquid temperature, flow rate of pump) , initial conditions (e.g. initial draw solution volume and concentration, initial feed solution volume and concentration) and the assumed initial dependent variables at $t = 1$. By means of Levenberg-Marquardt algorithm, the dependent variables at $t = 1$ could be simultaneously determined and were applied as the new initial dependent variables at t = 2. The sets of dependent variables at each point in time (t = 1, 2, 3,..., n) could be yielded at the end of each

simulation cycle. The MATLAB code files of each acid model were presented in Appendix F.
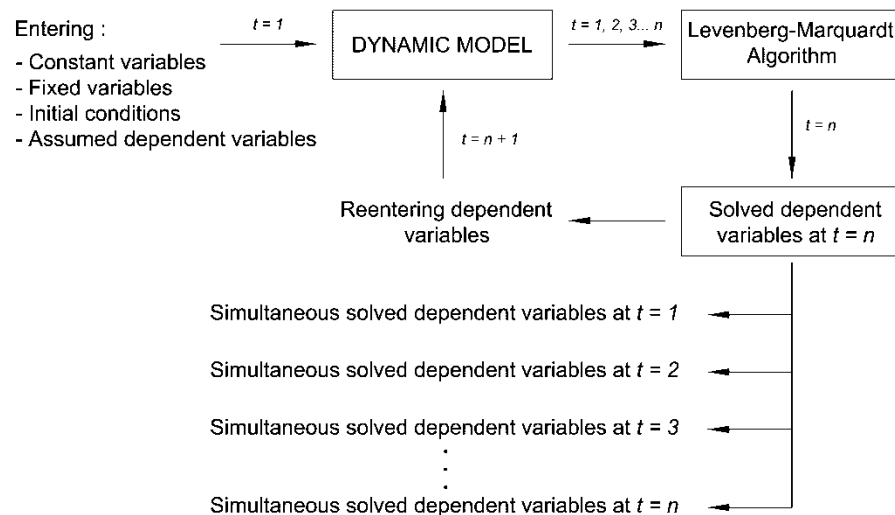
Fig. 4.17 The flow chart of model solution procedures

4.3.4 Results and discussion

4.3.4.1 FO membrane parameter characterization

Table 4.29 Membrane parameters (A, B_s, S), along with the correlated coefficients of determination of water flux $(R^2 - J_w)$, salt flux $(R^2 - J_s)$ and the coefficient of variation (CV), have been calculated by excel error minimization algorithms from Ref. (Tiraferri et al., 2013)

Sample	A $(L/m^2/h/bar)$	B_s $(L/m^2/h)$	S (μm)	$R^2 - J_w$	$R^2 - J_s$	$CV(\%)$
1	0.458	0.262	545	0.997	0.999	2.50
2	0.436	0.249	487	0.989	0.980	2.04
3	0.431	0.256	470	0.984	0.981	3.05
Mean Value	0.442	0.256	500			

Using excel algorithms, the mean values of A, B_s and S calculated from three samples of FO membrane were shown in Table 4.29. The simulation results have been

reported together with the related coefficients of determination of water flux $(R^2 - J_w)$, salt flux $(R^2 - J_s)$ and the coefficient of variation (CV). The mean values of A, B_s and S determined by this method were 0.442 L/m²/h/bar, 0.256 L/m²/h and 500 µm, respectively. The result of structure parameter (500 µm) is in the range 424 µm to 589 µm which is consistent with the reported value from Ref. (Tiraferri et al., 2013).

Table 4.30 Experiment results and calculation report of four acid permeability coefficients (B_a) by excel algorithms

Acid Type	Sample	Stage	$C_{d,a}$ (mM)	$C_{f,a}$ (mM)	J_w (L/m²/h)	J_a (mmole/m²/h)	B_a (L/m²/h)	Mean Value (L/m²/h)
Acetic	1	1	379	8.82	1.6	930.83	1.979	2.10
		2	736	29.8	3.4	1,999.60	2.145	
	2	1	488	19.3	2.2	1,337.28	2.178	
		2	906	42.0	3.9	2,385.42	2.102	
Butyric	1	1	392	2.11	1.8	241.46	0.576	0.64
		2	606	5.89	2.1	343.86	0.539	
	2	1	280	1.09	1.4	239.31	0.767	
		2	521	5.14	2.1	379.28	0.674	
Valeric	1	1	120	3.02	0.5	59.42	0.470	0.49
		2	254	8.97	0.9	123.87	0.468	
	2	1	159	5.19	0.7	85.23	0.510	
		2	292	12.74	1.1	158.02	0.521	
Lactic	1	1	365	0.17	4.3	75.22	0.198	0.15
		2	536	0.55	5.2	94.22	0.170	
	2	1	415	0.51	3.1	54.23	0.128	
		2	677	1.48	4.9	79.96	0.116	

To characterize the acid permeability coefficients of the membrane, two membrane samples were tested for each acid. The mean value of acid permeability coefficient (B_a) was calculated from two membrane samples and determined by two-stage individual calculation, using excel algorithms. The experimental data and simulation results of four different carboxylic acids are reported in Table 4.30. The highest permeability value was acetic acid, 2.10 L/m²/h. Butyric and valeric acid had

relatively close permeability values, 0.64 L/m^2/h and 0.49 L/m^2/h respectively, and the

value of lactic was the lowest at 0.15 L/m^2/h.

4.3.4.2 Validation of the mathematical model

4.3.4.2.1 In case of single a carboxylic acid as feed solution
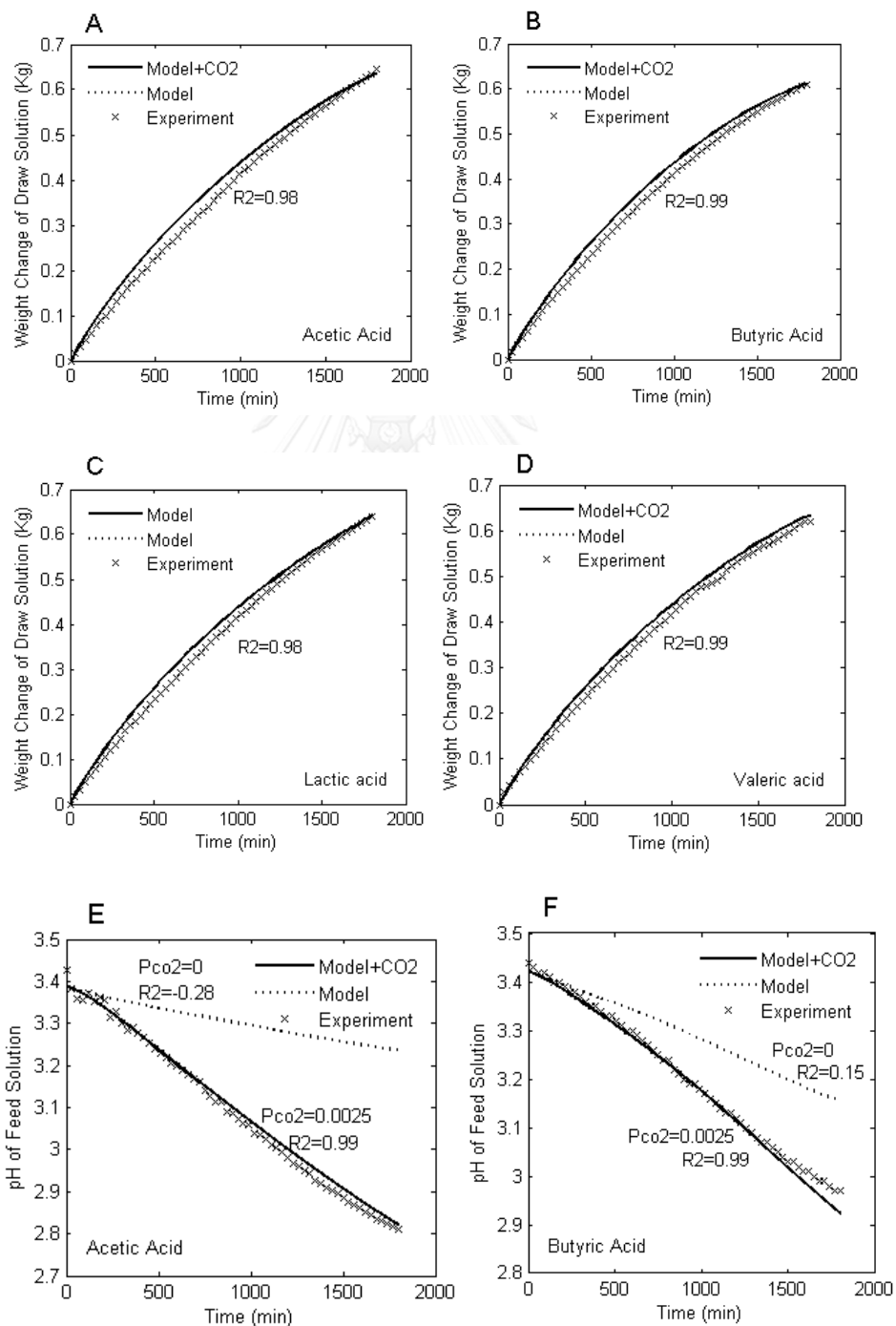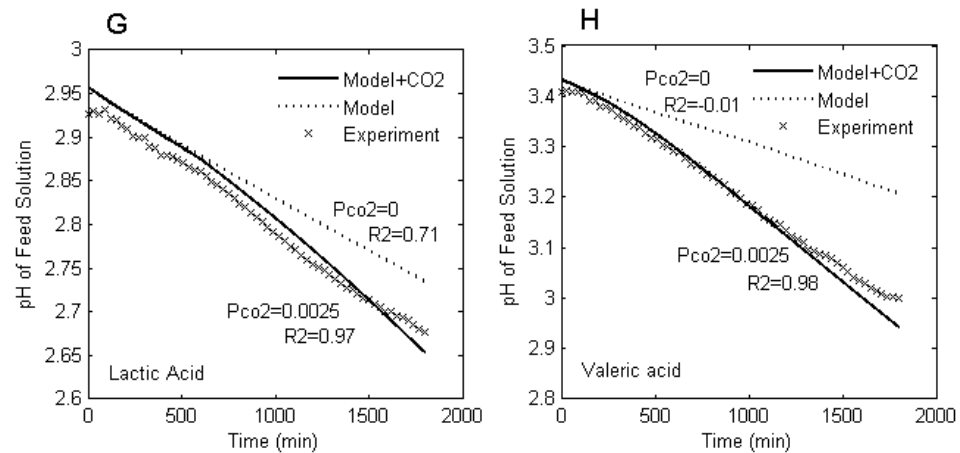
Fig. 4.18 A comparison of the simultaneous results from model predictions of both respective CO_2 permeation phenomena (solid line) and irrespective CO_2 permeation phenomena (dot line) with the experimental data (cross symbols) obtained from the acid concentration FO experiment of four different carboxylic acids. Weight change of draw solution (A, B, C, D) and pH of feed solution (E, F, G, H) are plotted against elapsed time.

In Fig. 4.18A-D, as a result of the different osmotic pressure between feed and draw solutions, the weight change of draw solution (W_d) escalates over the period of experimental time by accumulated permeate volumes in a draw solution tank. According to Fig.4.14 and Henry's law in Eq.(4.3.34), 1M NaCl draw solution, equilibrated with atmosphere, has the low CO_2 concentration (C_{d,co_2a}) in a draw solution tank. This low CO_2 concentration also produces very low bicarbonate ion concentration, determined by Eq.(4.3.36). The bicarbonate ions pass into the draw solution channel of FO test cell unit and then instantly react with hydrogen ions which gradually diffuse across a membrane from acid feed solution. The bicarbonate ion species are quickly converted to soluble carbon dioxides $(C_{d,co_2}$ in Fig.4.13), determined by Eq.(4.3.42). This small amount of the generated CO_2 partially diffuses through a membrane from draw solution to feed solution channel. At the support layer-active layer interface, the permeating CO_2 has a relatively tiny osmotic pressure, compared with the high osmotic pressure of NaCl draw solution. Consequently, this developed model can neglect the generated CO_2 in the osmotic pressure calculation at support layer-active layer

interface (π_i) in Eq.(4.3.6). In feed solution channel, the generated CO_2 diffuses through the boundary layer from membrane active layer surface and enters the acid feed solution channel under the increasingly acidified condition above atmospheric pressure in a FO module. Under this circumstance, the true carbonic acid (C_{f,CO_2} in Fig.4.13) is produced in the bulk feed solution. At membrane active layer surface, since the generated CO_2, which partially diffuses across a membrane from draw solution channel, is expected to be negligibly small, this very low CO_2 concentration and the irrelevant true carbonic acid in the bulk feed solution were thus not addressed in the osmotic pressure calculation at membrane active layer surface (π_{mf}) in Eq.(4.3.10). Depending on the different osmotic pressure across the membrane active layer$(\pi_i - \pi_{mf})$, the impact of CO_2 permeation phenomena on the water flux (J_w) in Eq.(4.3.1) is rationally ignorable. Therefore, the model predictions of irrespective and respective CO_2 permeation phenomena for the weight change of draw solution were alike (Fig. 4.18A-D). Indicated by R^2 of W_d for four carboxylic acids, the agreements between the simulation results from developed model and experimental data were higher than 0.95.

In case of the pH of feed solution, the pH of feed solution declined over the period of experimental time corresponding to the volume reduction of acid feed solution. Considering the acetic acid as feed solution in Fig. 4.18E at the end of experiment, 1800 minutes (cross symbol), the pH of acetic acid is 2.8, which is approximately equivalent to 145 mM acetic acid. By initial concentration of 10 mM, acetic acid was likely concentrated up to a 14.5-fold increase, whereas the weight change in Fig. 4.18A at 1800 minutes was observed to be 0.645 kg of draw solution. Since the decreasing feed solution volume is equal to the increasing draw solution volume, by initial feed solution volume of 1 L, acetic acid should possibly have the concentration up only to a 2.8-fold increase which is equivalent to pH of 3.15, determined by 28 mM acetic acid. This pH result could be performed by the irrespective CO_2 permeation model (dot line). Nonetheless, at 1800 minutes, the experimental pH of 2.8 was noticeably much lower than 3.15 of acetic acid feed solution. This discrepancy can be explained by the formation of true carbonic acid

which is a fairly strong acid (Pk_a=3.45) in a composition of acid feed solution. This phenomenon could be described by the respective CO_2 permeation model that was stated in section 2.10 (CO_2 permeability). Regardless of CO_2 permeation phenomena (dot line), the CO_2 permeation is neglected ($J_{CO_2} = 0$ in Eq.(4.3.43)) as CO_2 permeability coefficient is zero ($P_{CO_2} = 0$). Indicated by R^2 of -0.28, the pH predictions of irrespective CO_2 permeation model were obviously much far from the experimental data by considering only acetic acid component in feed solution. However, to reflect the consequence of CO_2 permeability phenomena during acid filtration, is to define the unobserved membrane CO_2 permeability coefficient (P_{CO_2}), in the Eq.(4.3.43). By means of inverse problems techniques, the process of calculating from a set of experimental data to infer the actual values of the model parameter, the respective CO_2 permeation model was fitted to experimental pH profiles (solid line), where R^2 was adjusted from -0.28 to 0.99. A P_{CO_2} value of 0.0025 $L/m^2/h$ was determined. This model parameter was also applied to the remaining acid models to validate against the experimental pH profiles in Fig. 4.18F-H. The goodness of fit of butyric acid, lactic acid and valeric acid models was obviously improved where R^2 were increased from 0.15 to 0.99, from 0.71 to 0.97 and from -0.01 to 0.98, respectively. Despite the much different R^2 values between the irrespective CO_2 model predictions (P_{CO_2}= 0) and the experimental pH profiles for each acid model, a P_{CO_2} value of 0.0025 $L/m^2/h$ could evidently improve the goodness of fit of all acid models to the experimental data. The agreement between the simulated results from proposed irrespective and respective CO_2 permeation model with experimental data have been evaluated by statistical factors, as shown in Table 4.31.

Regarding the model prediction, the targeted acid could be characterized by inputting three specific acid parameters, ionization constant (K_a), diffusion coefficient (D_a) and membrane permeability coefficient (B_a), into the model. In case of the acetic acid model which performs the lowest R^2 of -0.28, the lowest R^2 infers the highest impact from CO_2 permeation phenomena. As a result of high acid permeability property of 2.10 $L/m^2/h$ or high acid flux of acetic acid, CO_2 is increasingly generated in draw solution channel and then transports across a membrane to feed solution

channel to unignorably form the higher true carbonic acid content (Pk_a=3.45) which is more acidic than acetic acid (Pk_a=4.75). The consequence of the dominant true carbonic in the mixture of them could lead to the abnormally low pH results of acetic acid feed solution. Conversely, the lactic acid model could perform the highest R^2 of 0.71, interpreted as the lowest impact from CO_2 permeation phenomena due to the lowest corresponding acid permeability of 0.15 $L/m^2/h$. This reveals how much the CO_2 permeation phenomena influence to the pH of feed solution during carboxylic acid filtration. It relies on its acid permeability coefficient.

While using NaCl as draw solution, the impact of generated true carbonic acid on the increasing acidity in feed solution shows the distinguishable mechanism during the carboxylic acid filtration. Unlike using NH_4Cl as draw solution in section 4.1, since the CO_2 permeation phenomena did not appear, the experimental pH of the targeted acids were normally corresponding to their concentrations. This filtration mechanism was comparably demonstrated by the previous irrespective CO_2 permeation model. The diversity of chemical properties of draw solution can lead to the different mechanisms in the forward osmosis membrane processes. A NH_4Cl solution exhibits as a weak acid whereas a NaCl solution, common salt, acts like a CO_2 absorber.

Table 4.31 Quantitative comparisons of model predictions to experimental data

| Acid | Irrespective CO_2 permeation model | | | | | | Respective CO_2 permeation model | | | | | |
| | Weight VS. Time | | | pH VS. Time | | | Weight VS. Time | | | pH VS. Time | | |
	R^2	*RMSE*	*SEP%*	R^2	*RMSE*	*SEP%*	R^2	*RMSE*	*SEP%*	R^2	*RMSE*	*SEP%*
Acetic	0.98	0.0281	6.36	-0.28	0.2698	8.79	0.98	0.0281	6.36	0.99	0.0159	0.51
Butyric	0.99	0.0192	5.43	0.15	0.1010	3.43	0.99	0.0192	5.43	0.98	0.0153	0.48
Valeric	0.99	0.0172	4.76	-0.11	0.1272	3.96	0.99	0.0172	4.76	0.98	0.0195	0.61
Lactic	0.99	0.0186	5.14	0.71	0.0400	1.43	0.99	0.0186	5.14	0.97	0.0192	0.68

4.3.4.2.2 In case of a mixture of two carboxylic acids as feed solution
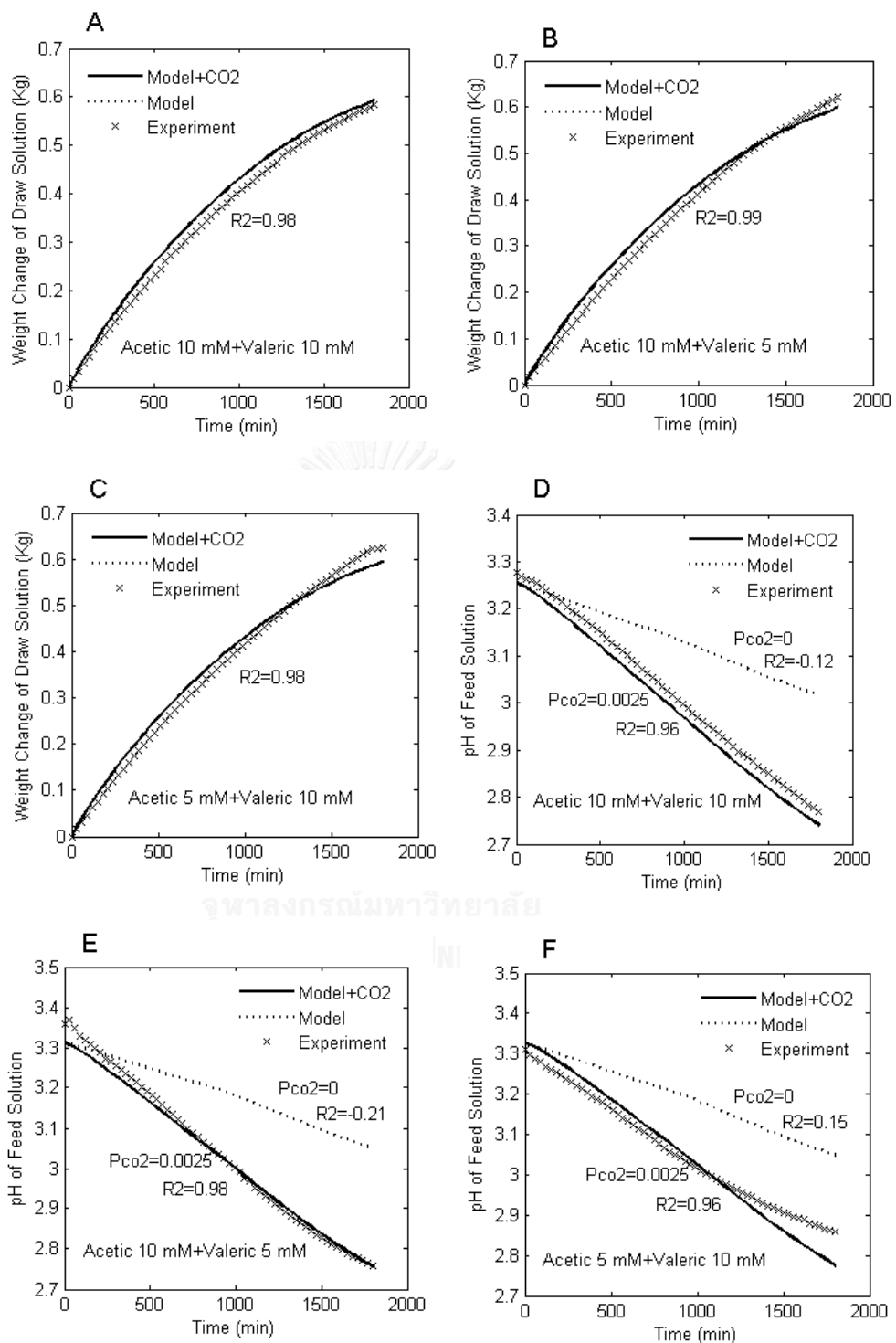
Fig. 4.19 A comparison of the simultaneous results from model predictions of respective CO_2 permeation phenomena (solid line) and irrespective CO_2 permeation phenomena (dot line) with the experimental data (cross symbols) obtained from acid

filtration FO experiment of three different compositions of acetic and valeric acid. Weight change of draw solution (A, B, C) and pH of feed solution (D, E, F) are plotted against elapsed time.

Likewise explaining in case of a single carboxylic acid as feed solution, in Fig.4.19 A-C, weight changes of draw solution of three different compositions of the acetic and valeric acid mixture, the prediction of irrespective CO_2 permeation model and respective CO_2 permeation model results for the weight change of draw solution were identical. Indicated by R^2 of weight change of draw solution, the agreements between the simulation results from developed model and experimental data for all different acid mixtures were higher than 0.95. In case of pH of feed solution, Fig.4.19D-F, the acetic acid as the composition of feed solution, has the high acid permeability as well as the high impact to the pH reduction of feed solution that stated in section 4.3.4.2. In Fig. 4.19E, the model prediction of irrespective CO_2 permeation (P_{CO_2} = 0) for the highest 10 mM acetic acid to 5 mM valeric acid ratio presents the lowest R^2 of pH of feed solution (R^2 = -0.21) compared with experimental pH profiles. Since a higher acetic acid ratio leads to produce a higher true carbonic acid content in the three-component system of acetic acid-valeric acid-true carbonic acid, the lowest R^2 of irrespective CO_2 permeation model was therefore observed. By inputting a P_{CO_2} value of 0.0025 into the model, the goodness of fit of all acid mixture models could be improved. Compared with the experimental pH profiles, the goodness of fit of respective CO_2 permeation model, for a mixture of 10 mM acetic acid+10 mM valeric acid, 10 mM acetic acid+5 mM valeric acid and 5 mM acetic acid+10 mM valeric acid, could be increased in term of R^2 from -0.12 to 0.96, from -0.21 to 0.98, and from 0.15 to 0.96, respectively. This is evidence that the P_{CO_2} value of 0.0025 $L/m^2/h$ could successfully be applied not only in the model of a single carboxylic but also a mixture of two carboxylic acids. The agreement between the simulated results from proposed

irrespective and respective CO_2 permeation model with experimental data have been evaluated by statistical factors, as given in Table 4.32.

Table 4.32 Quantitative comparisons of model predictions to experimental data

| | Irrespective CO_2 permeation | | | | | | Respective CO_2 permeation | | | | | |
| | Weight VS. Time | | | pH VS. Time | | | Weight VS. Time | | | pH VS. Time | | |
Acid	R^2	RMSE	SEP%	R^2	RMSE	SEP%	R^2	RMSE	SEP%	R^2	RMSE	SEP%
Acetic 10mM+Valeric 10mM	0.98	0.0204	5.90	-0.12	0.1400	4.62	0.98	0.0204	5.90	0.97	0.0286	0.95
Acetic 10mM+Valeric 5mM	0.99	0.0194	5.45	-0.21	0.1824	6.00	0.99	0.0170	4.85	0.99	0.0200	0.66
Acetic 5mM+Valeric 10mM	0.99	0.0192	5.31	0.15	0.1450	4.74	0.99	0.0192	5.31	0.96	0.0352	1.15

        4.3.4.4 Simulation results of acetic acid feed solution at 30 hour operations of FO process.

Table 4.33 List of the simulation results on dependent variables of respective CO_2 permeation model for acetic acid feed solution at 30 hour system operation. The variable results were solved by Levenberg-Marquardt algorithm

	Variable	Value	unit		Variable	Value	Unit
1.	$C_{d,a}$	5.2735×10^{-3}	$mole/L$	34.	I_f	3.7560×10^{-2}	$mole/L$
2.	C_{d,A^-}	3.9860×10^{-4}	$mole/L$	35.	I_i	2.8768×10^{-1}	$mole/L$
3.	$C_{d,ai}$	5.2725×10^{-3}	$mole/L$	36.	I_{mf}	3.8920×10^{-2}	$mole/L$
4.	C_{d,co_2}	1.3750×10^{-3}	$mole/L$	37.	J_a	4.3850×10^{-6}	$mole/dm^2/min$
5.	C_{d,co_2a}	1.3048×10^{-5}	$mole/L$	38.	J_{CO_2}	3.4374×10^{-6}	$mole/dm^2/min$
6.	C_{d,H^+}	4.0155×10^{-4}	$mole/L$	39.	J_s	1.1404×10^{-5}	$mole/dm^2/min$
7.	C_{d,HCO_3^-}	2.9538×10^{-6}	$mole/L$	40.	J_w	8.3153×10^{-4}	$L/dm^2/min$
8.	C_{d,OH^-}	5.8623×10^{-11}	$mole/L$	41.	k_d	1.4851×10^{-1}	dm/min
9.	$C_{d,s}$	4.3879×10^{-1}	$mole/L$	42.	$k_{f,a}$	1.5219×10^{-1}	dm/min
10.	$C_{d,so}$	4.3869×10^{-1}	$mole/L$	43.	$k_{f,s}$	1.4446×10^{-1}	dm^2/min
11.	$C_{f,ao}$	1.9953×10^{-2}	$mole/L$	44.	pH_f	2.8207	-
12.	C_{f,A^-}	2.7344×10^{-4}	$mole/L$	45.	Re_d	26,954	-
13.	$C_{f,a}$	1.9957×10^{-2}	$mole/L$	46.	Re_f	25,450	-
14.	C_{f,CO_2}	7.0042×10^{-3}	$mole/L$	47.	Sc_d	541.08	-
15.	C_{f,CO_2i}	7.0017×10^{-3}.	$mole/L$	48.	$Sc_{f,a}$	532.74	-
16.	C_{f,H^+}	1.8126×10^{-3}	$mole/L$	49.	$Sc_{f,s}$	576.05	-
17.	C_{f,HCO_3^-}	1.5391×10^{-3}	$mole/L$	50.	Sh_d	685.67	-
18.	C_{f,OH^-}	9.9616×10^{-12}	$mole/L$	51.	$Sh_{f,a}$	653.38	-
19.	$C_{f,s}$	3.7957×10^{-2}	$mole/L$	52.	$Sh_{f,s}$	670.63	-

	Variable	Value	Unit		Variable	Value	Unit
20.	$C_{f,si}$	3.7944×10⁻²	mole/L	53.	v_d	150.27	dm/min
21.	$C_{i,a}$	5.2735×10⁻³	mole/L	54.	v_f	150.24	dm/min
22.	C_{i,H^+}	3.9820×10⁻⁴	mole/L	55.	W_d	637.14	g
23.	$C_{i,s}$	2.8728×10⁻¹	mole/L	56.	$\gamma_{d,z1}$	7.3017×10⁻¹	-
24.	$C_{md,s}$	4.3645×10⁻¹	mole/L	57.	$\gamma_{f,z1}$	8.3370×10⁻¹	-
25.	$C_{mf,a}$	2.0051×10⁻²	mole/L	58.	$\gamma_{i,z1}$	7.3364×10⁻¹	-
26.	C_{mf,H^+}	6.9715×10⁻⁴	mole/L	59.	$\gamma_{mf,z1}$	8.3490×10⁻¹	-
27.	$C_{mf,s}$	3.8223×10⁻²	mole/L	60.	γ_o	1.1035	-
28.	$D_{d,s}$	9.4864×10⁻⁶	dm²/min	61.	μ_d	5.2023	g/dm/min
29.	$D_{f,a}$	1.0202×10⁻⁵	dm²/min	62.	μ_f	5.4145	g/dm/min
30.	$D_{f,s}$	9.4350×10⁻⁶	dm²/min	63.	π_i	13.600	bar
31.	$F_{f,o}$	1.5797	L/min	64.	π_{mf}	2.3099	bar
32.	$F_{d,o}$	1.5803	L/min	65.	ρ_d	1013.5	g/L
33.	I_d	4.3909×10⁻¹	mole/L				

.

Table 4.34 List of dependent variable for a mixture of 10 mM of acetic and valeric acid as feed solution at 1800 minute system operation, solved by LM algorithm

	Variable	Value	Unit		Variable	Value	Unit
1.	$C_{d,a}$	6.8443×10⁻³	mole/L	40.	$F_{f,o}$	1.5797	L/min
2.	$C_{d,a'}$	2.8844×10⁻³	mole/L	41.	$F_{d,o}$	1.5803	L/min
3.	C_{d,A^-}	3.9511×10⁻⁴	mole/L	42.	I_d	4.6631×10⁻¹	mole/L
4.	$C_{d,A^{-'}}$	1.36691×10⁻⁴	mole/L	43.	I_f	6.4337×10⁻²	mole/L
5.	$C_{d,ai}$	6.8437×10⁻³	mole/L	44.	I_i	3.1490×10⁻¹	mole/L
6.	$C_{d,ai'}$	2.8841×10⁻³	mole/L	45.	I_{mf}	6.3663×10⁻²	mole/L
7.	C_{d,co_2}	1.8267×10⁻³	mole/L	46.	J_a	5.2798×10⁻⁶	mole/dm²/min
8.	C_{d,co_2a}	1.2975×10⁻⁵	mole/L	47.	$J_{a'}$	2.2250×10⁻⁶	mole/dm²/min
9.	C_{d,H^+}	5.3474×10⁻⁴	mole/L	48.	J_{CO_2}	4.5665×10⁻⁶	mole/dm²/min
10.	C_{d,HCO_3^-}	2.9416×10⁻⁶	mole/L	49.	J_s	1.1163×10⁻⁵	mole/dm²/min
11.	C_{d,OH^-}	4.3924×10⁻¹¹	mole/L	50.	J_w	7.7141×10⁻⁴	L/dm²/min
12.	$C_{d,s}$	4.6587×10⁻¹	mole/L	51.	k_d	1.4828×10⁻¹	dm/min
13.	$C_{d,si}$	4.6578×10⁻¹	mole/L	52.	$k_{f,a}$	1.5278×10⁻¹	dm/min
14.	$C_{f,a}$	2.3086×10⁻²	mole/L	53.	$k_{f,a'}$	1.4874×10⁻¹	dm/min
15.	$C_{f,a'}$	3.7927×10⁻²	mole/L	54.	$k_{f,s}$	1.4415×10⁻¹	dm²/min
16.	C_{f,A^-}	2.7270×10⁻⁴	mole/L	55.	pH_f	2.7407	-
17.	$C_{f,A^{-'}}$	3.6461×10⁻⁴	mole/L	56.	Re_d	26,916	-
18.	$C_{f,ao}$	2.3090×10⁻²	mole/L	57.	Re_f	25,451	-
19.	$C_{f,ao'}$	3.7936×10⁻²	mole/L	58.	Sc_d	542.59	-
20.	C_{f,CO_2}	8.2912×10⁻³	mole/L	59.	$Sc_{f,a}$	529.67	-
21.	C_{f,CO_2i}	8.2883×10⁻³	mole/L	60.	$Sc_{f,a'}$	552.86	-
22.	C_{f,H^+}	2.2544×10⁻³	mole/L	61.	$Sc_{f,s}$	577.89	-
23.	C_{f,HCO_3^-}	1.6171×10⁻³	mole/L	62.	Sh_d	685.57	-
24.	C_{f,OH^-}	8.5701×10⁻¹²	mole/L	63.	$Sh_{f,a}$	652.13	-
25.	$C_{f,s}$	6.2101×10⁻²	mole/L	64.	$Sh_{f,a'}$	661.52	-
26.	$C_{f,si}$	6.2082×10⁻²	mole/L	65.	$Sh_{f,s}$	671.35	-
27.	$C_{i,a}$	6.8443×10⁻³	mole/L	66.	v_d	150.27	dm/min
28.	$C_{i,a'}$	2.8844×10⁻³	mole/L	67.	v_f	150.24	dm/min

29. C_{i,H^+}	$5.3268{\times}10^{-4}$	$mole/L$	68. W_d	591.66	g
30. $C_{i,s}$	$3.1437{\times}10^{-1}$	$mole/L$	69. $\gamma_{d,z1}$	$7.3098{\times}10^{-1}$	-
31. $C_{md,s}$	$4.6357{\times}10^{-1}$	$mole/L$	70. $\gamma_{f,z1}$	$8.0597{\times}10^{-1}$	-
32. $C_{mf,a}$	$2.3168{\times}10^{-2}$	$mole/L$	71. $\gamma_{i,z1}$	$7.3176{\times}10^{-1}$	-
33. $C_{mf,a'}$	$3.8264{\times}10^{-2}$	$mole/L$	72. $\gamma_{mf,Z1}$	$8.0658{\times}10^{-1}$	-
34. C_{mf,H^+}	$1.1963{\times}10^{-3}$	$mole/L$	73. ρ_f	999.0	g/L
35. $C_{mf,s}$	$6.2467{\times}10^{-2}$	$mole/L$	74. μ_d	5.2149	$g/dm/min$
36. $D_{d,s}$	$9.4735{\times}10^{-6}$	dm^2/min	75. μ_f	5.4295	$g/dm/min$
37. $D_{f,a}$	$1.0261{\times}10^{-5}$	dm^2/min	76. π_i	14.998	bar
38. $D_{f,a'}$	$9.8306{\times}10^{-6}$	dm^2/min	77. π_{mf}	4.5241	bar
39. $D_{f,s}$	$9.4048{\times}10^{-6}$	dm^2/min	78. ρ_d	1014.5	g/L

Based on Levenberg-Marquardt algorithm, Table 4.33 and Table 4.34 illustrate the simulation results of 65 and 78 dependent process variables at the end of simulation, 1800 minutes, for a single acetic acid and a mixture of 10 mM acetic and 10 mM valeric acid, respectively. As given in table, the proposed model could predict all dependent variables, the hydrodynamic parameters, physical and chemical properties of both feed and draw solutions in the FO process at 1800 minute system operation. Furthermore, the immeasurable parameters, the concentrations and osmotic pressures at membrane surface and at support layer-active layer interface depend on the individual membrane properties. These membrane parameters could be predicted and also reflect comprehensively the membrane performance. Due to the true carbonic acid formation in forward osmosis process, the developed model can express the influence of permeating CO_2 on the incongruous pH results with its concentration. Regarding the pH operating condition, this developed model can help to forecast the unexpected pH drop, caused by the undesired true carbonic acid in forward osmosis system. By simulating the targeted acid as feed solution, the results can practically be used to select proper material in FO system in designing phase. Even in the operating phase, the operator can also use this information to plan more protective maintenance procedures on FO system to minimize corrosion of system hardware.

Nomenclature

A	water permeability coefficient of the membrane
A_I	temperature-dependent constant
A_M	effective membrane area
B_a	acid permeability coefficient of the membrane
$B_{a'}$	acid type 2 permeability coefficient of the membrane
B_s	salt permeability coefficient of the membrane
C	salt concentration in the boundary layer
C_a	acid concentration in the boundary layer or support layer
$C_{d,a}$	acid concentration in bulk draw solution
$C_{d,a'}$	acid type 2 concentration in bulk draw solution
C_{d,A^-}	acid ion concentration in draw solution
$C_{d,A^{-'}}$	acid ion type 2 concentration in draw solution
$C_{d,ai}$	acid concentration into the inlet of draw solution channel
$C_{d,ai'}$	acid type 2 concentration into the inlet of draw solution channel
C_{d,CO_2}	carbon dioxide concentration in the draw solution channel
C_{d,CO_2a}	CO_2 concentration in draw solution tank
$C_{d,b}$	bulk draw solution concentration
C_{d,H^+}	hydrogen ion concentration in draw solution
C_{d,HCO_3^-}	bicarbonate ion concentration in draw solution
C_{d,OH^-}	hydroxide ion concentration in draw solution
$C_{d,s}$	salt concentration in bulk draw solution
$C_{d,so}$	salt concentration from the outlet of draw solution channel
C_{d,s,t_0}	initial salt concentration in draw solution tank
$C_{f,a}$	acid concentration in feed solution
$C_{f,a'}$	acid type 2 concentration in feed solution
C_{f,A^-}	acid ion concentration in feed solution
$C_{f,A^{-'}}$	acid ion type 2 concentration in feed solution
$C_{f,ao}$	acid concentration from the outlet of feed solution channel
$C_{f,ao'}$	acid type 2 concentration from the outlet of feed solution channel

C_{f,a,t_0}	initial acid concentration in feed solution tank
C_{f,a',t_0}	initial acid type 2 concentration in feed solution tank
$C_{f,b}$	bulk feed solution concentration
C_{f,CO_2}	true carbonic acid concentration
C_{f,CO_2i}	true carbonic acid concentration into the inlet of feed solution channel
C_{f,H^+}	hydrogen ion concentration in feed solution
C_{f,HCO_3^-}	bicarbonate ion concentration in feed solution
C_{f,OH^-}	hydroxide ion concentration in feed solution
$C_{f,s}$	salt concentration in bulk feed solution
$C_{f,si}$	salt concentration into the inlet of feed solution channel
$C_{i,a}$	acid concentration at support layer-active layer interface
$C_{i,a'}$	acid type 2 concentration at support layer-active layer interface
C_{i,H^+}	hydrogen ion concentration at support layer-active layer interface
$C_{i,s}$	salt concentration at support layer-active layer interface
$C_{md,a}$	acid concentration at support layer surface
$C_{md,a'}$	acid type 2 concentration at support layer surface
$C_{md,s}$	salt concentration at support layer surface
$C_{mf,a}$	acid concentration at active layer surface
$C_{mf,a'}$	acid type 2 concentration at active layer surface
C_{mf,H^+}	hydrogen ion concentration at active layer surface
$C_{mf,s}$	salt concentration at active layer surface
C_s	salt concentration in membrane support layer
CV	coefficient of variation
D	channel depth of FO test cell
D_a	average weak acid diffusion coefficient
$D_{a'}$	average weak acid type 2 diffusion coefficient
$D_{a,298}$	average weak acid diffusion coefficient at 298 kelvins
$D_{A^-,T}$	acid ion diffusion coefficient
$D_{A^-',T}$	acid ion type 2 diffusion coefficient
$D_{A^-,298}$	acid ion diffusion coefficient at 298 kelvins

$D_{d,s}$	average salt diffusion coefficient in draw solution
D_{eff}	effective salt diffusion coefficient in porous support layer
$D_{f,a}$	apparent weak acid diffusion coefficient in feed solution
$D_{f,a'}$	apparent weak acid type 2 diffusion coefficient in feed solution
$D_{f,s}$	average salt diffusion coefficient in feed solution
d_h	hydraulic diameter
D_{H^+}	hydrogen ion diffusion coefficient at 298 kelvins
$D_{H^+,T}$	hydrogen ion diffusion coefficient
$D_{HA,T}$	acid molecule diffusion coefficient
$D_{HA',T}$	acid molecule type 2 diffusion coefficient
$D_{HA,298}$	acid molecule diffusion coefficient at 298 kelvins
$D_{ion,T}$	average acid ion diffusion coefficient
D_s	salt diffusion coefficient
F	flow rate of recirculating pump
$F_{f,o}$	flow rate from the outlet of feed solution channel
$F_{d,o}$	flow rate from the outlet of draw solution channel
I_d	ionic strength of draw solution
I_f	ionic strength of feed solution
I_i	ionic strength at support layer- active layer interface
I_{mf}	ionic strength at active layer surface
J_a	acid flux
$J_{a'}$	acid type 2 flux
J_{CO_2}	transmembrane CO_2 flux into feed solution channel
J_s	reverse salt flux
$J_{s,l}$	reverse salt flux in stage l
J_w	water flux
$J_{w,l}$	water flux in stage l
k	mass transfer coefficient
$K_{a,T}$	acid ionization constant
$K_{a',T}$	acid type 2 ionization constant
$K_{CO_2,T}$	first ionization constant of carbonic acid

$K_{H_2CO_3}$	true carbonic acid ionization constant
k_d	salt mass transfer coefficient at draw solution side
$k_{f,a}$	acid mass transfer coefficient at feed solution side
$k_{f,a'}$	acid type 2 mass transfer coefficient at feed solution side
$k_{f,s}$	salt mass transfer coefficient at the feed solution side
K_H	temperature dependence Henry's constant
$K_{w,T}$	water ionization constant
l	actual thickness of support layer yields
L	channel length of FO test cell
l_{eff}	effective thickness of support layer
n_s	number of salt ion species
P_{CO_2}	membrane CO_2 permeability coefficient
p_{CO_2}	CO_2 partial pressure
pH_f	pH of feed solution
R	gas constant
R^2	coefficient of determination
$R^2 - J_s$	coefficient of determination of salt flux
$R^2 - J_w$	coefficient of determination of water flux
Re_d	Reynolds number in draw solution channel
Re_f	Reynolds number in feed solution channel
S	structure parameter of membrane support layer
Sc_d	Schmidt number of salt at draw solution side
$Sc_{f,a}$	Schmidt number of acid at feed solution side
$Sc_{f,a'}$	Schmidt number of acid type 2 at feed solution side
$Sc_{f,s}$	Schmidt number of salt at feed solution side
Sh_d	Sherwood number of salt in open channel at draw solution side
$Sh_{f,a}$	Sherwood number of acid in open channel at feed solution side
$Sh_{f,a'}$	Sherwood number of acid type 2 in open channel at feed solution side
$Sh_{f,s}$	Sherwood number of salt in open channel at feed solution side
t	elapsed time

T	absolute temperature
v_d	average flow velocity in draw solution channel
v_f	average flow velocity in feed solution channel
V_{f,t_0}	initial volume of feed solution
V_{d,t_0}	initial volume of draw solution
W	channel width of FO test cell
W_d	weight change of draw solution
Z	charge of ion

Greek symbols

$\gamma_{d,Z1}$	single charge activity coefficient in draw solution
$\gamma_{f,Z1}$	single charge activity coefficient in feed solution
$\gamma_{i,Z1}$	single charge activity coefficient at support layer-active layer interface
$\gamma_{mf,z1}$	single charge activity coefficient at active layer surface
γ_0	activity coefficient of dissolved CO_2 in the NaCl draw solution
δ	boundary layer thickness
ε	support layer porosity
μ_d	dynamic viscosity of NH_4Cl solution
μ_f	dynamic viscosity of feed solution
$\pi_{d,a}$	osmotic pressure of acid draw solution
$\pi_{d,b}$	osmotic pressure of bulk draw solution
$\pi_{f,a}$	osmotic pressure of acid feed solution
$\pi_{f,b}$	osmotic pressure of bulk feed solution
π_i	osmotic pressure at interface of support layer-active layer
π_{mf}	osmotic pressure at the active layer surface
ρ_d	density of draw solution
ρ_f	density of feed solution
ρ_w	water density
σ	reflection coefficient
τ	tortuosity of support layer
$\emptyset_a$	osmotic coefficient of acid solution

$\emptyset_s$	osmotic coefficient of salt solution

Subscripts

0	initial condition
298	298 kelvins
a	acid species
a'	acid type 2 species
A^-	acid ion species
$A^{-'}$	acid ion type 2 species
ao	acid species from the outlet of feed solution channel
ao'	acid type 2 species from the outlet of feed solution channel
ai	acid species into the inlet of draw solution channel
ai'	acid type 2 species into the inlet of draw solution channel
b	bulk solution
CO_2	carbon dioxide or true carbonic acid species
CO_2a	carbon dioxide species from the air
CO_2i	true carbonic acid species into the inlet of feed solution channel
d	draw solution
eff	effective
f	feed solution
h	hydraulic
H	Henry's constant
H^+	hydrogen ion species
HA	acid molecule species
HA'	acid molecule type 2 species
HCO_3^-	bicarbonate ion species
H_2CO_3	true carbonic acid species
I	ionic strength
i	membrane support layer- active layer interface
ion	ion species

ion'	ion type 2 species
l	number of experiment stages
M	membrane
md	membrane support layer surface
mf	membrane active layer surface
OH^-	hydroxide ion species
s	salt species
si	salt species into the inlet of feed solution channel
so	salt species from the outlet of feed solution channel
T	absolute temperature
t_0	zero time
w	water species
$z1$	single charge

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY
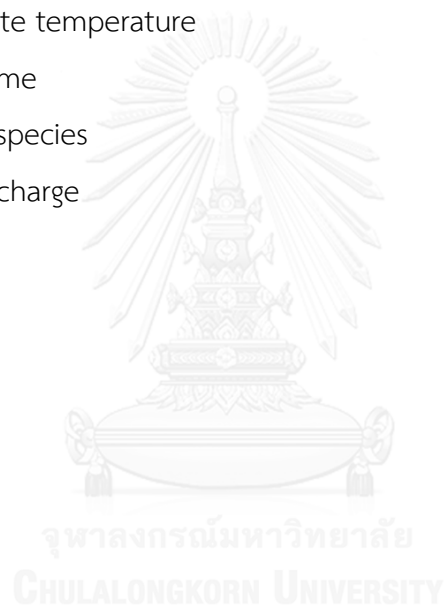
CHAPTER V

CONCLUSIONS

5.1 In case of a single carboxylic acid as feed solution and NH_4Cl as draw solution

In the membrane testing, the mean values of the A, B_s and S obtained from excel algorithms were 0.413 $L/m^2/h/bar$, 0.460 $L/m^2/h$ and 491 µm respectively. Butyric and valeric acid had relatively close permeability value, 0.64 $L/m^2/h$ and 0.49 $L/m^2/h$, respectively. The highest permeability value was acetic acid, 2.10 $L/m^2/h$ and the value of lactic was lowest at 0.15 $L/m^2/h$.

Formulating the forty-five logical phenomena equations during concentrating carboxylic acid by forward osmosis process, the developed mathematical model of FO process was proposed. With well-controlled experiments, the model has been validated against four selected carboxylic acid as feed solutions at the similar initial condition. R^2, $RMSE$ and SEP have been determined to act as validation metrics against the experimental data. The goodness of the model can be measured by R-squared (R^2) which had the value in the range from 0.98 to 0.99 and 0.64 to 0.77 for weight change of draw solution and pH of feed solution respectively. $RMSE$ and SEP are used to quantify the accuracy of the model. The developed model had root mean square error $(RMSE)$ for weight change of draw solution and pH of feed solution in the range from 0.0149 to 0.0253 and 0.0171 to 0.0375 respectively and standard error of prediction (SEP) for weight change of draw solution and pH of feed solution in the range from 4.19% to 6.99% and 0.52% to 1.13% respectively.

Regarding the rejection and concentration performance of the FO process, acetic acid performs the lowest acid rejection (71%) and concentration performance (1.65 fold increase) due to its minimum molar mass, corresponding to the highest acid

permeability coefficient among the other carboxylic acids. Butyric acid and valeric acid have the same pKa value but valeric acid has higher molar mass than butyric acid, causing acid rejection and concentration performance of valeric acid (91.5% and 2.3 fold-increase) to be higher than those of butyric acid (89% and 2.2 fold-increase). Comparing lactic acid to butyric acid, the molar mass of lactic acid is close to butyric acid but lactic acid has significantly lower pKa than butyric acid. As lactic acid can induce the increasing of membrane surface charge, it performed the highest acid rejection of 97.3% and concentration performance of 2.4 fold increase corresponding to the lowest acid permeability coefficient.

The developed simulation model not only could quantify acid rejection and concentration performance of forward osmosis system at any point in simulating time (Dynamic Model) but also could suggest the optimal region of initial operating condition to attain the nearby maximum system performance at the end of acid concentration process.

In sensitivity analysis, the effects of change in four variables of initial condition on the rejection rate and concentration performance have been determined in terms of percentage change. It can be used as a guideline to configure the starting condition of the system so that the optimal rejection rate and concentration performance can be obtained by considering both operational and economical aspects as a whole.

5.2 In case of a mixture of two carboxylic acids as feed solution and NH$_4$Cl as draw solution

Integrating by the ninety-seven logical phenomena equations during concentrating carboxylic acid by forward osmosis process, the developed mathematical model of the carboxylic acid mixture as feed solution and NH4Cl as draw solution has been accomplished. With dynamic experiments, the model has been validated against the three selected carboxylic acid mixture as feed solutions at the same initial condition. The accuracy of model prediction, R-squared (R^2) had the value

in the range from 0.97 to 0.99 and 0.61 to 0.68 for weight change of draw solution and pH of feed solution respectively. The developed model had root mean square error ($RMSE$) for weight change of draw solution and pH of feed solution in the range from 0.0202 to 0.0235 and 0.0199 to 0.0278 respectively and standard error of prediction (SEP) for weight change of draw solution and pH of feed solution in the range from 5.88% to 9.34% and 0.68% to 0.88% respectively.

5.3 In case of a single carboxylic acid and a mixture of two carboxylic acids as feed solution and NaCl as draw solution

The mean values of A, B_s and S determined by FO protocol were 0.442 L/m^2/h/bar, 0.256 L/m^2/h and 500 µm, respectively. By simulating the filtration of carboxylic acid in forward osmosis, the novel CO_2 transportation mechanism, across a FO membrane from NaCl draw solution to sequentially form the true carbonic acid in acid feed solution, was proposed and could be demonstrated by the dynamic process modeling. The developed model is formulated from the simultaneous sixty-five equations for a single carboxylic acid and seventy-eight equations for a carboxylic acid mixture, respectively. The dissolution of CO_2 in brine draw solution plays the significant role in generating the true carbonic acid and substantially affects the lowering of pH of acid feed solution. The greater acid permeability value, the higher impact of CO_2 permeation phenomena occurs. Normally, the membrane CO_2 permeability (0.0025 L/m^2/h) could not directly be observed, but could be determined by inverse problems techniques. To simulate the experiment results, all initial condition variables and required constant variables must be inputted into the model to determine the time-dependent variables via Levenberg-Marquardt algorithm.

The accuracy of model predictions for a single carboxylic acid, R-squared (R^2) had the value in the range from 0.98 to 0.99 and 0.98 to 0.99 for weight change of draw solution and pH of feed solution respectively The developed model had root

mean square error ($RMSE$) for weight change of draw solution and pH of feed solution in the range from 0.0172 to 0.0281 and 0.0153 to 0.0195 respectively and standard error of prediction (SEP) for weight change of draw solution and pH of feed solution in the range from 4.76% to 6.36% and 0.48% to 0.68% respectively.

In case of a mixture of acetic acid and valeric acid as feed solution, the higher acetic acid ratio can lead to the more influential CO_2 transport phenomena, affecting the pH reduction as a result of the higher true carbonic acid content in a mixture of them. The P_{CO_2} value of 0.0025 L/m^2/h could be effectively applied not only in the model of a single carboxylic but also a mixture of two carboxylic acids. The accuracy of model predictions, R-squared (R^2) had the value in the range from 0.98 to 0.99 and 0.96 to 0.99 for weight change of draw solution and pH of feed solution respectively. The developed model had root mean square error ($RMSE$) for weight change of draw solution and pH of feed solution in the range from 0.0192 to 0.0204 and 0.02 to 0.0352 respectively and standard error of prediction (SEP) for weight change of draw solution and pH of feed solution in the range from 5.31% to 5.90% and 0.66% to 1.15% respectively.

In the simulations, either the sixty-five dependent variables of a single carboxylic acid model or the seventy-eight dependent variables of a mixture of two carboxylic acid model were solved by Levenberg-Marquardt (LM) algorithm that could exactly determine any concerning variables at each point in simulating time by inputting the initial condition variables, the fixed variables and the constant variables into the model. By changing the three acid parameters: acid ionization constant (K_a), acid diffusion coefficient (D_a) and acid permeability coefficient (B_a), the models could be validated against various types of carboxylic acids. Under similar environment and system configuration, these models can comprehensively predict the behavior of other carboxylic acids by inputting those three characteristic

parameters of targeted acids into the model. Due to the true carbonic acid formation in forward osmosis process, the developed model can express the influence of permeating CO_2 on the incongruous pH results with its concentration. Regarding the pH operating condition, this developed model can help to forecast the unexpected pH drop, caused by the undesired true carbonic acid in forward osmosis system. By simulating the targeted acid as feed solution, the results can practically be used to select proper material in FO system in designing phase. Even in the operating phase, the operator can also use this information to plan more protective maintenance procedures on FO system to minimize corrosion of system hardware.

Practically, the wastewater characteristic is the combination of various contaminants. The forthcoming research should subsequently focus on a mixture of various organic acids and inorganic salts to simulate more feasible environments. The achievement on such a research will offer the further profound understanding and advancing knowledge in this area, which can be contributed to future application of acid concentration of FO process in more practical manner

REFERENCES

Aaberg, R.J. 2003. Osmotic power: A new and powerful renewable energy source *Refocus*, **4**(6), 48-50.

Acetic Acid. 2015. Wikipedia, free encyclopedia. Available from: http://en.wikipedia.org/wiki/Acetic acid.

Achilli, A., Cath, T.Y., Marchand, E.A., Childress, A.E. 2009. The forward osmosis membrane bioreactor: A low fouling alternative to MBR processes. *Desalination*, **239**(1–3), 10-21.

Anderson, D.K., University of Rhode, I. 1977. Concentration of dilute industrial wastes by Direct osmosis, University of Rhode Island. [Kingston].

Baker, R.W. 2004. Reverse Osmosis. in: *Membrane Technology and Applications*, John Wiley & Sons, Ltd, pp. 191-235.

Beaudry, E.G. and Lampi, K A. 1990. Membrane technology for direct osmosis concentration of fruit juices. *Food Technology,* **44**, 121.

Beaudry, E.G., Herron, J.R. 1997. Direct Osmosis for Concentrating Wastewater, SAE International.

Bhatt, P.P. 2004. Osmotic drug delivery systems for poorly soluble drugs. Pharma Ventures Ltd., Oxford, UK. Available from: http://www.shirelabs.com/news/11_30_04.pdf.

BioMatnet. 2000. Production of fatty acid esters usable as fuels by fermentation of biomass. Available from: http://www.biomatnet.org/secure/Fair/S503.html.

Butler, J. N. 1991. Carbon Dioxide: Equilibria and Their Application. Lewis Publishers INC., Michigan, pp. 33-34.

Butyric acid. 2015. Wikipedia, free encyclopedia. Available from: http://en.wikipedia.org/wiki/Butyric_acid

Cath, T.Y., Adams, D., Childress, A.E. 2005a. Membrane contactor processes for wastewater reclamation in space: II. Combined direct osmosis, osmotic distillation, and membrane distillation for treatment of metabolic wastewater. *Journal of Membrane Science*, **257**(1–2), 111-119.

Cath, T.Y., Childress, A.E., Elimelech, M. 2006. Forward osmosis: Principles, applications, and recent developments. *Journal of Membrane Science*, **281**(1–2), 70-87.

Cath, T.Y., Gormly, S., Beaudry, E.G., Flynn, M.T., Adams, V.D., Childress, A.E. 2005b. Membrane contactor processes for wastewater reclamation in space: Part I. Direct osmotic concentration as pretreatment for reverse osmosis. *Journal of Membrane Science*, **257**(1–2), 85-98.

Cartinella, J.L., Cath, T.Y., Flynn, M.T., Miller, GC., Hunter, K.W., Childress, A.E. 2006. Removal of natural steroid hormones from wastewater using membrane contactor Processes. *Environment Science and Technology* **40**(23), 7381- 7386.

Coday, B. D., Almaraz, N., Cath, T.Y. 2015. Forward osmosis desalination of oil and gas wastewater: Impacts of membrane selection and operating conditions on process performance. *Journal of Membrane Science,* **488**, 40-55.

Compilation of Henry's Law Constant for Inorganic and Organic Species of Potential Importance in Environmental Chemistry. Available from: http://www.mpch-mainz.mpg.de/~sander/res/henry. html.

Cornelissen, E.R., Harmsen, D., de Korte, K.F., Ruiken, C.J., Qin, J.-J., Oo, H., Wessels, L.P. 2008. Membrane fouling and process performance of forward osmosis membranes on activated sludge. *Journal of Membrane Science*, **319**(1–2), 158-168.

Cussler, E.L. 1997. Diffusion: Mass Transfer in Fluid System. second ed., Cambridge University Press, New York.

Einstein, A. 1926. Investigation on the Theory of the Brownian Movement. Dover, New York.

Elimelech, M., Bhattacharjee, S. 1998. A novel approach for modeling concentration polarization in crossflow membrane filtration based on the equivalence of osmotic pressure model and filtration theory. *Journal of Membrane Science*, **145**(2), 223-241.

Formic Acid. 2012. Wikipedia, free encyclopedia. Available from: http://en.wikipedia.org/wiki/Acetic_acid

Garcia-Castello, E.M., McCutcheon, J.R., Elimelech, M. 2009. Performance evaluation of sucrose concentration using forward osmosis. *Journal of Membrane Science*, **338**(1–2), 61-66.

Gray, G.T., McCutcheon, J.R., Elimelech, M. 2006. Internal concentration polarization in forward osmosis: role of membrane orientation. *Desalination*, **197**(1–3), 1-8.

Harned, H.S. and Owen, B.B. 1958. The Physical Chemistry of Electrolytic Solution. third ed., Reinhold Book Corporation, New York.

Haynes, W.M. 2014-2015. CRC Handbook of Chemistry and Physics. ninety-fifth ed., CRC Press Taylor & Francis Group, Boca Raton, London, New York.

Holloway, R.W., Childress, A.E., Dennett, K.E., Cath, T.Y. 2007. Forward osmosis for concentration of anaerobic digester centrate. *Water Research*, **41**(17), 4005-4014.

HTI, Military/X Pack. [Online] 2005. Available from: http://www.hydrationtech.com

Jellinek, H.H.G., Masuda, H. 1981. Osmo-power. Theory and performance of an osmo-power pilot plant. *Ocean Engineering*, **8**(2), 103-128.

Jiao, B., Cassano, A., Drioli, E. 2004. Recent advances on membrane processes for the concentration of fruit juices: a review. *Journal of Food Engineering*, **63**(3), 303-324.

Jin, X., Tang, C.Y., Gu, Y., She, Q., Qi, S. 2011. Boric Acid Permeation in Forward Osmosis Membrane Processes: Modeling, Experiments, and Implications. *Environmental Science & Technology*, **45**(6), 2323-2330.

Jung, J.W., Lee, H.S., Kim, K.-J. 2008. Purification of Acetic Acid Wastewater using Layer Melt Crystallization. *Separation Science and Technology*, **43**(5), 1021-1033.

Kessler, J.O. and Moody, C.D. 1976. Drinking water from sea water by forward osmosis. *Desalination,* **18**, 297-306.

Kravath, R.E. and Davis, J.A. 1975. Desalination of seawater by direct osmosis. *Desalination,* **16**, 151-155.

Lactic acid. Wikipedia, free encyclopedia. 2015. Available from: http://en.wikipedia.org/wiki/Lactic acid.

Latorre, M. 2005. Environmental impact of brine disposal on Posidinia Seagrasses. *Desalination* **182**, 517-524.

Lee, K.L., Baker, R.W., Lonsdale, H.K. 1981. Membranes for power generation by pressure-retarded osmosis. *Journal of Membrane Science*, **8**(2), 141-171.

Loeb, S. 1971. Desalination by Reverse Osmosis : John McDermott (Ed.), Noyes Data Corporation, Park Ridge, N.J., 1970, $35.00. *Desalination*, **9**(1), 96.

Loeb, S. 1998. Energy production at the Dead Sea by pressure-retarded osmosis: challenge or chimera? *Desalination*, **120**(3), 247-262.

Loeb, S. 2001. One hundred and thirty benign and renewable megawatts from Great Salt Lake? The possibilities of hydroelectric power by pressure-retarded osmosis. *Desalination*, **141**(1), 85-91.

Loeb, S. 2002. One hundred and thirty benign and renewable megawatts from Great Salt Lake? The possibilities of hydroelectric power by pressure-retarded osmosis with spiral module membranes: Desalination, 141 (2001) 85–91. *Desalination,* **142**(2), 207.

Loeb, S. 1976. Production of energy from concentrated brines by pressure-retarded osmosis : I. Preliminary technical and economic correlations. *Journal of Membrane Science*, **1**, 49-63.

Loeb, S. 1992. The tiltpipe liquid thermosiphon. *Heat Recovery Systems and CHP*, **12**(1), 1-9.

Marc, L. 2007. Model for Calculating the Viscosity of Aqueous Solutions, *Journal of Chemical Engineering Data,* **52**, 321-335.

McCutcheon, J.R., Elimelech, M. 2006. Influence of concentrative and dilutive internal concentration polarization on flux behavior in forward osmosis. *Journal of Membrane Science*, **284**(1–2), 237-247.

McCutcheon, J.R., McGinnis, R.L., Elimelech, M. 2006. Desalination by ammonia–carbon dioxide forward osmosis: Influence of draw and feed solution concentrations on process performance. *Journal of Membrane Science*, **278**(1–2), 114-123.

McCutcheon, J.R., McGinnis, R.L., Elimelech, M. 2005. A novel ammonia—carbon dioxide forward (direct) osmosis desalination process. *Desalination*, **174**(1), 1-11.

Mehta, G.D., Loeb, S. 1978. Internal polarization in the porous substructure of a semipermeable membrane under pressure-retarded osmosis. *Journal of Membrane Science*, **4**, 261-265.

Merkel, B., Planer-Friedrich, B., Nordstrom, D.K. 2008. *Groundwater geochemistry : a practical guide to modeling of natural and contaminated aquatic systems*. *2nd ed*. Springer, Berlin.

Mulder, M. 1997. Basic Principles of Membrane Technology. 2nd ed. Dordrecht, The Netherlands, Kluwer Academic Publishers.

Nano Magnatic. 2005. Water purification. Available from: http://www.nanomagnetics.com/water_purification.asp.

Nayak, C.A., Rastogi, N.K. 2010. Forward osmosis for the concentration of anthocyanin from Garcinia indica Choisy. *Separation and Purification Technology*, **71**(2), 144-151.

Novotny, P., Sohnel, O. 1988. Densities of binary aqueous solutions of 306 inorganic substances. *Journal of Chemical & Engineering Data*, **33**(1), 49-55.

OLI Stream Analyzer 2.0. 2005. OLI Systems Inc., Morris Plains, NJ. Available from: http://www.olisystems.com/.

Phillip, W.A., Yong, J.S., Elimelech, M. 2010. Reverse Draw Solute Permeation in Forward Osmosis: Modeling and Experiments. *Environmental Science & Technology*, **44**(13), 5170-5176.

Phuntsho, S., Shon, H.K., Hong, S., Lee, S., Vigneswaran, S. 2011. A novel low energy fertilizer driven forward osmosis desalination for direct fertigation: Evaluating the performance of fertilizer draw solutions. *Journal of Membrane Science*, **375**(1–2), 172-181.

Robinson, R.A. and Stokes, R.H. 1959. Electrolyte Solutions. second ed. (revised), Butterworths & Co., London: pp. 517-521.

Rodríguez, M., Luque, S., Alvarez, J., Coca, J. 2000. A comparative study of reverse osmosis and freeze concentration for the removal of valeric acid from wastewaters. *Desalination*, **127**(1), 1-11.

Sablani, S.S., Goosen, M.F.A., Al-Belushi, R., Wilf, M. 2001. Concentration polarization in ultrafiltration and reverse osmosis: a critical review. *Desalination*, **141**(3), 269-289.

Sant'Anna, V., Marczak, L.D.F., Tessaro, I.C. 2012. Membrane concentration of liquid foods by forward osmosis: Process and quality view. *Journal of Food Engineering*, **111**(3), 483-489.

Shibuya, M., Yasukawa, M., Takahashi, T., Miyoshi, T., Higa, M., Matsuyama, H. 2015. Effect of operating conditions on osmotic-driven membrane performances of cellulose triacetate forward osmosis hollow fiber membrane. *Desalination*, **362**, 34-42.

Song, L., Elimelech, M. 1995. Theory of concentration polarization in crossflow filtration. *Journal of the Chemical Society, Faraday Transactions*, **91**(19), 3389-3398.

Sourirajan, S. 1970. *Reverse osmosis*. Logos, London,.

Tang, C.Y., She, Q., Lay, W.C.L., Wang, R., Fane, A.G. 2010. Coupled effects of internal concentration polarization and fouling on flux behavior of forward osmosis membranes during humic acid filtration. *Journal of Membrane Science*, **354**(1–2), 123-133.

Timmer, J.M.K., Kromkamp, J., Robbertsen, T. 1994. Lactic acid separation from fermentation broths by reverse osmosis and nanofiltration. *Journal of Membrane Science*, **92**(2), 185-197.

Tiraferri, A., Yip, N.Y., Straub, A.P., Romero-Vargas Castrillon, S., Elimelech, M. 2013. A method for the simultaneous determination of transport and structural parameters of forward osmosis membranes. *Journal of Membrane Science*, **444**, 523-538.

Valeric acid. 2015. Wikipedia, free encyclopedia. Available from: http://en.wikipedia.org/wiki/Valeric_acid.

Vertova, A., Aricci, G., Rondinini, S., Miglio, R., Carnelli, L., D'Olimpio, P. 2009. Electrodialytic recovery of light carboxylic acids from industrial aqueous wastes. *Journal of Applied Electrochemistry*, **39**(11), 2051.

Votta, F., Barnett, S.M., Anderson, D.K. 1974. Concentration of industrial waste by direct osmosis. completion report, Providence, RI.

Wang, W., Zhang, Y., Esparra-Alvarado, M., Wang, X., Yang, H., Xie, Y. 2014. Effects of pH and temperature on forward osmosis membrane flux using rainwater as the makeup for cooling water dilution. *Desalination*, **351**, 70-76.

Wieland, P.O. 1994. Design for human presence in space: an introduction to environment control and life support systems. NASA George C. Marshall Space

Flight Center, Alabama. Available from: http://flightprojects.msfc.nasa.gov/book/rp1324.pdf.

Wick, G.L. 1978. Power from salinity gradients. *Energy*, **3**(1), 95-100.

Wrolstad, R.E., McDaniel, M.R., Durst, R.W., Micheals, N., Lampi, K.A., Beaudry, E.G. 1993. Composition and Sensory Characterization of Red Raspberry Juice Concentrated by Direct-Osmosis or Evaporation. *Journal of Food Science*, **58**(3), 633-637.

York, R.J., Thiel, R.S., Beaudry, E.G. 1999. Full-Scale Experience of Direct Osmosis Concentration Applied to Leachate Management. Proceedings of the Seventh International Waste Management and Landfill Symposium (Sardinia '99), S. Margherita di Pula, Cagliari, Sardinia, Italy.

Yang, K., Yu, Y., Hwang, S. 2003. Selective optimization in thermophilic acidogenesis of cheese-whey wastewater to acetic and butyric acids: partial acidification and methanation. *Water Research*, **37**(10), 2467-2477.

Zhang, F., Brastad, K.S., He, Z. 2011. Integrating Forward Osmosis into Microbial Fuel Cells for Wastewater Treatment, Water Extraction and Bioelectricity Generation. *Environmental Science & Technology*, **45**(15), 6690-6696.

Zhao, S., Zou, L., Tang, C.Y., Mulcahy, D. 2012. Recent developments in forward osmosis: Opportunities and challenges. *Journal of Membrane Science*, **396**, 1-21.

Zhu, H., Zhang, L., Wen, X., Huang, X. 2012. Feasibility of applying forward osmosis to the simultaneous thickening, digestion, and direct dewatering of waste activated sludge. *Bioresource Technology*, **113**, 207-213.

APPENDIX

Appendix A

Derivation of water flux and acid flux equations for acid permeability coefficient determination
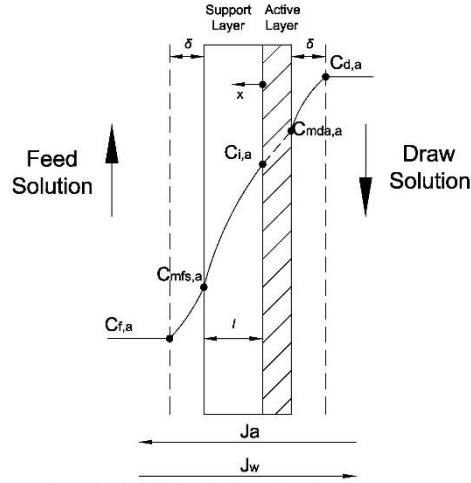
The figure shows the acid concentration profile across FO membrane layers operating in PRO mode. Carboxylic acids enter and penetrate through the porous support layer difficultly resulting in the internal concentration polarization (ICP). The acid flux (J_a) can be written in term of convective water flux and direct diffusion, according to the following equation:

$$-J_a = J_w C_a - D_{eff,a} \frac{dC_a}{dx} \tag{A1}$$

where C_a is the acid concentration in membrane support layer and $D_{eff,a}$ is the effective acid diffusion coefficient in porous support layer with porosity (ε) and related to the apparent weak acid diffusion coefficient (D_a), by $D_{eff,a} = \varepsilon D_a$.

The acid flux, J_a, across rejection layer is expressed by

$$J_a = B_a(C_{mda,a} - C_{i,a}) \tag{A2}$$

where B_a is acid permeability coefficient of the membrane, $C_{mda,a}$ is the acid concentration at active layer surface and $C_{i,a}$ is the acid concentration at support layer-active layer interface. At steady state, the acid flux across the support layer and rejection layer are equal. Substituting Eq. (A2) for J_a in Eq. (A1) yields

$$-B_a(C_{md,a} - C_{i,a}) = J_w C_a - D_{eff,a}\frac{dC_a}{dx} \tag{A3}$$

Integrating Eq. (A3) across the membrane support layer thickness (l_{eff})

at $x = 0, C_a = C_{i,a}$ and at $x = -l_{eff} = -\tau l, C_a = C_{mfs,a}$

where τ is the tortuosity of support layer and l is the actual thickness of support layer and $C_{mfs,a}$ is the acid concentration at support layer surface, yields

$$C_{i,a} = C_{mfs,a}e^{\left(\frac{J_w S}{D_a}\right)} + \frac{B_a(C_{mda,a}-C_{i,a})}{J_w}\left[e^{\left(\frac{J_w S}{D_a}\right)} - 1\right] \tag{A4}$$

where S is the structure parameter of the membrane support layer and defined by $S = \frac{\tau l}{\varepsilon}$.

The dilutive external concentration polarization (ECP) at active layer surface can be expressed by

$$-J_a = J_w C_a - \frac{dD_a C_a}{dx} \tag{A5}$$

Substituting Eq. (A2) for J_a in Eq. (A5) yields

$$-B_a(C_{mda,a} - C_{i,a}) = J_w C_a - D_a\frac{dC_a}{dx} \tag{A6}$$

Integrating Eq. (A6) across the boundary layer thickness (δ)

at $x = 0, C_a = C_{mda,a}$ and $x = \delta = \frac{D_a}{k_d}$, $C_a = C_{d,a}$, yields

$$C_{mda,a} = \frac{\left[C_{d,a}+\frac{B_a(C_{mda,a}-C_{i,a})}{J_w}\right]}{e^{\left(\frac{J_w}{k_{d,a}}\right)}} - \frac{B_a(C_{mda,a}-C_{i,a})}{J_w} \tag{A7}$$

where $k_{d,a}$ is acid mass transfer coefficient at the draw solution side

Subtracting Eq. (A4) from Eq. (A7) results in

$$C_{mda,a} - C_{i,a} = \frac{C_{d,a}e^{\left(-\frac{J_w}{k_{d,a}}\right)} - C_{mfs,a}e^{\left(\frac{J_w S}{D_a}\right)}}{1 - \frac{B_a}{J_w}(e^{\left(-\frac{J_w}{k_{d,a}}\right)} - e^{\left(\frac{J_w S}{D_a}\right)})} \tag{A8}$$

As supplementary assumption, the effect of concentrative external concentration polarization at feed solution side is insignificant by comparing to membrane support layer thickness, hence $C_{mfs,a}$ is estimated to $C_{f,a}$, $C_{mfs,a} \approx C_{f,a}$, and then substituting Eq. (A8) into Eq. (A2) results in acid flux equation for determination of acid permeability coefficient in the PRO mode below:

$$J_a = B_a \left[\frac{C_{d,a}e^{\left(-\frac{J_w}{k_{d,a}}\right)} - C_{f,a}e^{\left(\frac{J_w S}{D_a}\right)}}{1 - \frac{B_a}{J_w}\left(e^{\left(-\frac{J_w}{k_{d,a}}\right)} - e^{\left(\frac{J_w S}{D_a}\right)}\right)} \right] \tag{A9}$$

J_w across the active layer depends on the effective difference of osmotic pressure and can be expressed as

$$J_w = A\left(\pi_{mda,a} - \pi_{i,a}\right) \tag{A10}$$

where A is the water permeability coefficient of the membrane, $\pi_{mda,a}$ is the osmotic pressure of acid solution at the membrane active layer surface and $\pi_{i,a}$ is the osmotic pressure of acid solution at support layer-active layer interface.

By applying Van't Hoff equation, the differential of osmotic pressure is a linear function of concentration gradient. According to Eq. (A8), concentration gradient, $(C_{mda,a} - C_{i,a})$, is equivalent to different osmotic pressure, $\left(\pi_{mda,a} - \pi_{i,a}\right)$, and then substituting into Eq. (A10) yields the water flux equation for determination of acid permeability coefficient in the PRO mode:

$$J_w = A \left[\frac{\pi_{d,a} e^{\left(-\frac{J_w}{k_{d,a}}\right)} - \pi_{f,a} e^{\left(\frac{J_w S}{D_a}\right)}}{1 - \frac{B_a}{J_w}\left(e^{\left(-\frac{J_w}{k_{d,a}}\right)} - e^{\left(\frac{J_w S}{D_a}\right)}\right)} \right] \tag{A11}$$

where $\pi_{d,a}$ is the osmotic pressure of acid draw solution and $\pi_{f,a}$ is the osmotic pressure of acid feed solution. $\pi_{d,a}$ and $\pi_{f,a}$ can be determined by Eqs. (A12) and (A13) respectively:

$$\pi_{d,a} = \left(C_{d,a} + C_{d,H^+}\right)RT \tag{A12}$$

$$\pi_{f,a} = \left(C_{f,a} + C_{f,H^+}\right)RT \tag{A13}$$

where C_{f,H^+} and C_{d,H^+} are the hydrogen ion concentration in feed solution and draw solution and can be calculated from Eqs. (A14) and (A15), respectively:

$$C_{f,H^+} = \frac{\left(-K_a + (K_a^2 + 4K_a C_{f,a})^{1/2}\right)}{2} \tag{A14}$$

$$C_{d,H^+} = \frac{\left(-K_a + (K_a^2 + 4K_a C_{d,a})^{1/2}\right)}{2} \tag{A15}$$

Appendix B

Calculation of water flux $(J_{w,i})$ and acid flux $(J_{a,i})$ in Table 4.18

$J_{w,i}$ is a water flux in stage i and determined by the following equation:

$$J_{w,i} = \frac{W_d}{A_M t} \tag{B1}$$

where W_d is the weight change of draw solution, t is elapsed time and A_M is membrane area.

$J_{a,i}$ is acid flux in stage i and calculated from the following equation:

$$J_{a.i} = \frac{C_{F,i}(V_{F0,i} - J_{w,i}A_m t) - C_{F0,i}V_{F0,i}}{A_m t} \tag{B2}$$

where $C_{F,i}$ is the concentration of acid feed solution in stage i at time (t), $V_{F0,i}$ is the initial volume of feed solution, $C_{F0,i}$ is the initial concentration of feed solution in stage i and $V_{F0,i}$ is the initial volume of feed solution in stage i.

In stage one for a sample no. 1 of acetic feed solution

where W_d= 6.72×10^{-3} kg, A_M= 42 cm² and t= 60min.

$$J_{w,1} = \frac{6.72 \times 10^{-3} \times 10^4 \times 60}{42 \times 60} = 1.6 \text{ L/m}^2/\text{h}$$

where $C_{F,i}$= 8.82 mM, $V_{F0,i}$= 0.987 L and $C_{F0,i}$ = 4.8 mM

$$J_{a.1} = \frac{8.82\left(0.987 - \frac{1.6 \times 42 \times 10^{-4} \times 60}{60}\right) - 4.8 \times 0.987}{42 \times 10^{-4} \times \frac{60}{60}} = 930.6 \frac{\text{mmmole}}{\text{m}^2\text{h}}$$

Acid Permeability Coefficient Calculation by Excel Algorithm

	Draw				Feed										Js in mm and µmol	Jw/Js – L/mmol
Stage	C_D (mmol/L)	C_{Db} (mmol/L)	C_D (mmol/µL)	C_{Db} (mmol/µL)	C_F (mmol/L)	C_{Fb} (mmol/L)	C_F (mmol/µL)	C_{Fb} (mmol/µL)	J_W L m-2 h-1	J_W µL mm-2 h-1	πF	πD	J_S mmol m-2 h-1			
1	379	0.002566994			8.82	0.000384275			1.600	1.6	0.221	9.485	0.00882	930.83	0.93083	0.001719
2	736	0.003580643			298	0.000713553			3.400	3.4	0.746	18.419	0.0298	1999.6	1.9996	0.001700
														CV	0.77%	

Graph Fit

Jw – Eq.

Stage	Jw(exp) Lm-2h-1	Jw(cal) µL mm-2 h-1	Jw(cal) Lm-2h-1	(Yexp-Ycal)^2
1	1.600	4.97E+00	4.974571369	11.38773192
2	3.400	9.63E+00	9.633293548	38.85394845

Js – Eq.

Stage	Js(exp) mmol m-2 h-1	Js(cal) µmol mm-2 h-1	Js(cal) mmol m-2 h-1	(Yexp-Ycal)^2
1	930.83	1.032574445606	1032.574445605926	10351.93434
2	1999.6	1.9995919198332	1999.5919832208	6.42671E-05

R	0.08314	L bar-1 K-1 mol-1
	0.00008314	L bar-1 K-1 mmol-1
T	28	C
	301	K
kd	0.023016667	cm/s
	8.29E+02	mm/h
ka	0.0000175	
D	1.36E-05	cm2/sec
	4.90E+00	mm2/h

B-Acid		
	1.9989974423	
	2.1453328315	
B AVG	2.062616514	

	A	0.4130000000	µL mm-2 h-1 bar-1
	B	2.1453328315	µL mm-2 h-1
	S	491.0000000000	x 10^-3 mm

1096332207

3885401272

Appendix C
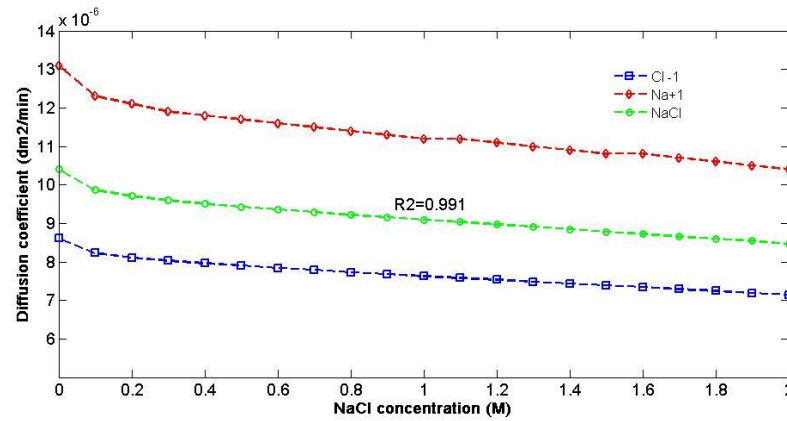
Fig. C The values of average NaCl diffusion coefficient at 28 °C (D_{NaCl}), correlated with its molar concentration (circle symbols). D_{NaCl} were calculated from Na$^+$ and Cl$^-$ diffusion coefficient data (Cussler, 1997), provided by OLI software. The curved fitting of average NaCl diffusion coefficient as a function of molar concentration, C, is represented by the following polynomial

$$D_{NaCl} = 5.05{\times}10^{-11}C^4 - 2.59{\times}10^{-9}C^3 + 4.65{\times}10^{-8}C^2 + 4.03{\times}10^{-7}C + 1.06{\times}10^{-5} \tag{C1}$$

Appendix D
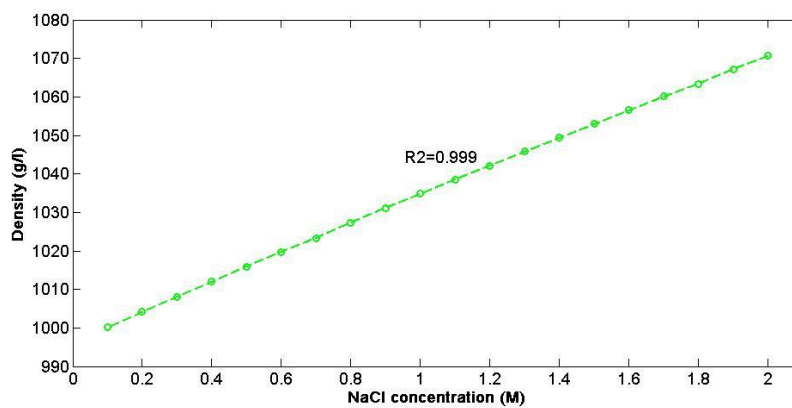
Fig. D The density of NaCl solution at different molar concentrations was calculated by using OLI software and the results are plotted. The correlation can be represented by the following equation:

$$\rho_{NaCl} = 37.0166C + 997.2911 \qquad \text{(D1)}$$

Appendix E
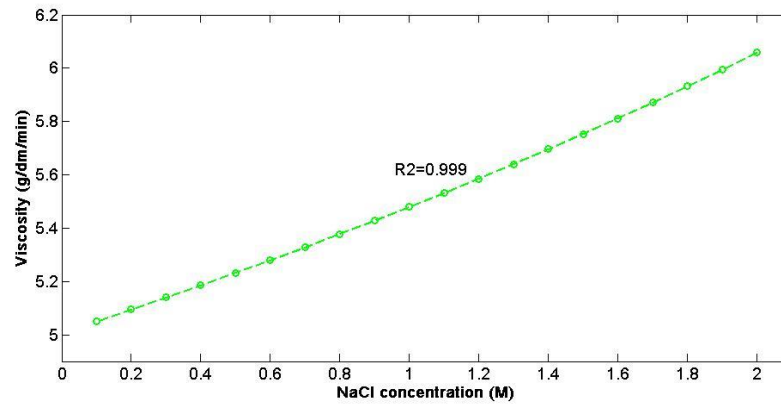
Fig. E The viscosity of NaCl solution at different molar concentrations was calculated by using OLI software and the results are plotted. The correlation can be represented by the following equation:

$$\mu_{NaCl} = 0.0538C^2 + 0.4159C + 5.0095 \tag{E1}$$

Appendix F

F.1 - The curved fitting of molar concentration of acetic acid as a function of conductivity.

Concentration(millimolar)	Conductivity(µS/cm)
1.00	38.6
0.90	36.9
0.80	33.8
0.70	32.0
0.60	28.5
0.50	26.5
0.40	23.1
0.30	19.6
0.20	15.6
0.10	10.6

$y = -2.3285E\text{-}06x^3 + 6.6841E\text{-}04x^2 + 3.6306E\text{-}03x - 1.0824E\text{-}02$
$R^2 = 9.9855E\text{-}01$
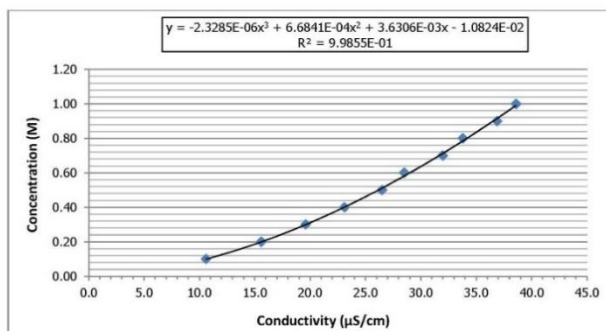
Concentration(millimolar)	Conductivity(µS/cm)
10.00	148.0
9.00	142.4
8.00	133.5
7.00	125.4
6.00	115.5
5.00	105.0
4.00	92.7
3.00	81.4
2.00	64.1
1.00	46.0

$y = 1.8646E\text{-}06x^3 - 9.3138E\text{-}05x^2 + 4.7953E\text{-}02x - 1.1960E\text{+}00$
$R^2 = 9.9949E\text{-}01$
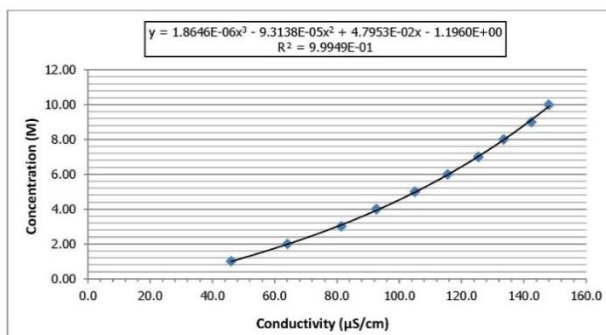
Concentration(millimolar)	Conductivity(µS/cm)
100.00	420.0
90.00	400.0
80.00	370.0
70.00	356.0
60.00	326.0
50.00	302.0
40.00	266.0
30.00	234.0
20.00	187.0
10.00	137.0

$y = 4.7504E\text{-}08x^3 + 5.5270E\text{-}04x^2 - 2.2269E\text{-}03x + 8.2325E\text{-}02$
$R^2 = 9.9853E\text{-}01$
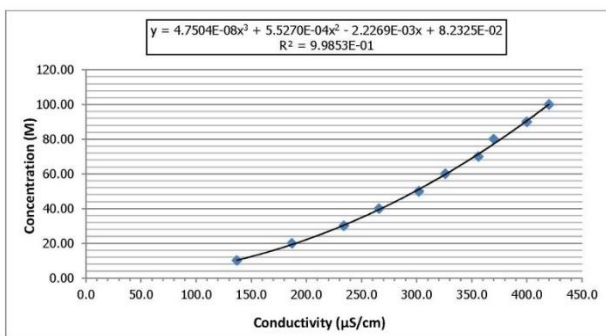
F.2 – The curved fitting of molar concentration of butyric acid as a function of conductivity.

Concentration (millimolar)	Conductivity(µS/cm)
100.00	377.0
90.00	358.0
80.00	343.0
70.00	319.0
60.00	296.5
50.00	270.0
40.00	242.0
30.00	214.0
20.00	165.2
10.00	125.9

$y = 1.3559E\text{-}06x^3 - 2.4637E\text{-}04x^2 + 2.0267E\text{-}01x - 1.3972E+01$
$R^2 = 9.9927E\text{-}01$
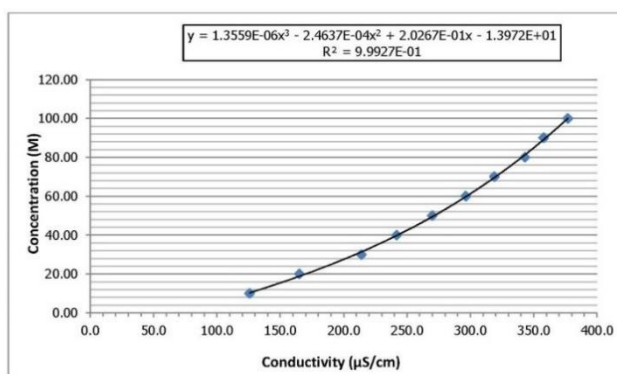
Concentration (millimolar)	Conductivity(µS/cm)
10.00	114.7
9.00	112.2
8.00	103.8
7.00	97.8
6.00	89.5
5.00	82.0
4.00	72.0
3.00	63.0
2.00	50.0
1.00	36.4

$y = 3.9706E\text{-}06x^3 - 2.0312E\text{-}04x^2 + 6.8490E\text{-}02x - 1.4222E+00$
$R^2 = 9.9780E\text{-}01$
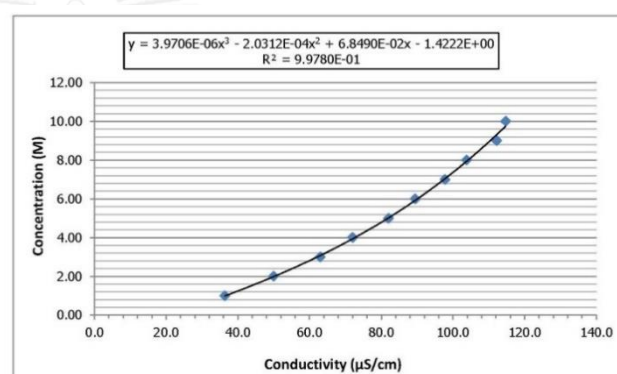
Concentration (millimolar)	Conductivity(µS/cm)
1.00	37.3
0.90	35.8
0.80	33.2
0.70	30.3
0.60	28.2
0.50	24.8
0.40	23.7
0.30	19.8
0.20	15.5
0.10	12.0

$y = -5.4682E\text{-}06x^3 + 8.9516E\text{-}04x^2 + 1.5590E\text{-}03x - 3.2963E\text{-}02$
$R^2 = 9.9638E\text{-}01$
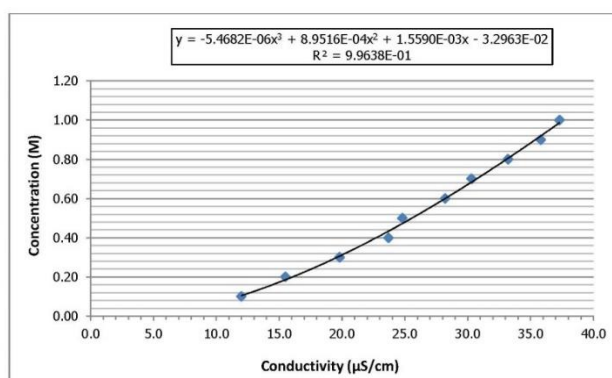
F.3 – The curved fitting of molar concentration of valeric acid as a function of conductivity.

Concentration (millimolar)	Conductivity(µS/cm)
10.00	120.6
9.00	116.2
8.00	107.6
7.00	102.1
6.00	92.1
5.00	84.1
4.00	78.6
3.00	68.1
2.00	54.7
1.00	36.8

$y = -4.0229E-06x^3 + 1.6539E-03x^2 - 7.2906E-02x + 1.6464E+00$
$R^2 = 9.9761E-01$
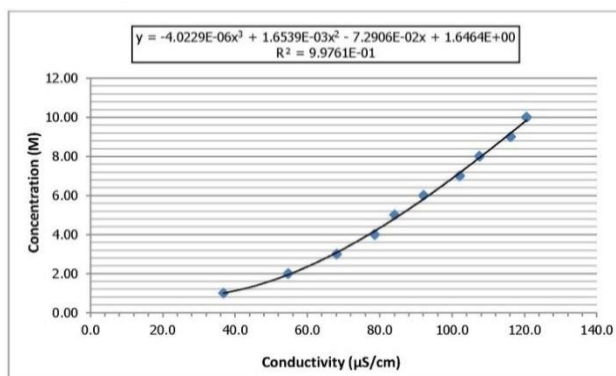
Concentration (millimolar)	Conductivity(µS/cm)
1.00	40.5
0.90	37.8
0.80	35.6
0.70	33.8
0.60	30.7
0.50	27.9
0.40	23.2
0.30	20.8
0.20	17.4
0.10	14.2

$y = 2.4460E-05x^3 - 1.6720E-03x^2 + 6.7046E-02x - 5.8528E-01$
$R^2 = 9.9778E-01$
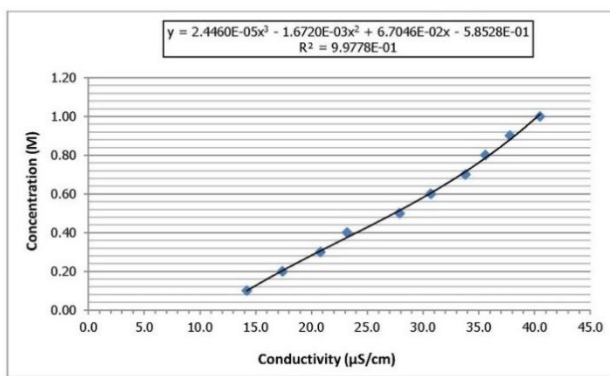
F.4 - The curved fitting of molar concentration of lactic acid as a function of conductivity.

| Concentration (millimolar)	Conductivity(µS/cm)
10.00	470.0
9.00	450.0
8.00	420.0
7.00	381.0
6.00	349.0
5.00	312.0
4.00	272.0
3.00	233.0
2.00	170.6
1.00	117.5

$y = 2.1610E\text{-}08x^3 + 1.0913E\text{-}05x^2 + 1.2428E\text{-}02x - 6.2835E\text{-}01$
$R^2 = 9.9887E\text{-}01$
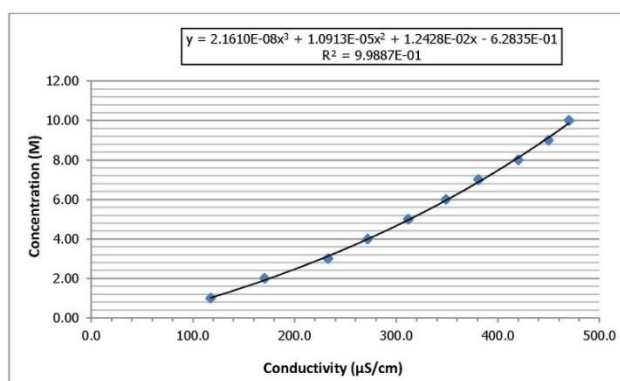
Concentration (millimolar)	Conductivity(µS/cm)
1.00	104.0
0.90	99.7
0.80	91.4
0.70	82.7
0.60	73.3
0.50	63.7
0.40	59.7
0.30	48.3
0.20	36.0
0.10	21.7

$y = -3.3221E\text{-}07x^3 + 1.0809E\text{-}04x^2 + 1.6113E\text{-}03x + 1.5840E\text{-}02$
$R^2 = 9.9643E\text{-}01$

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY
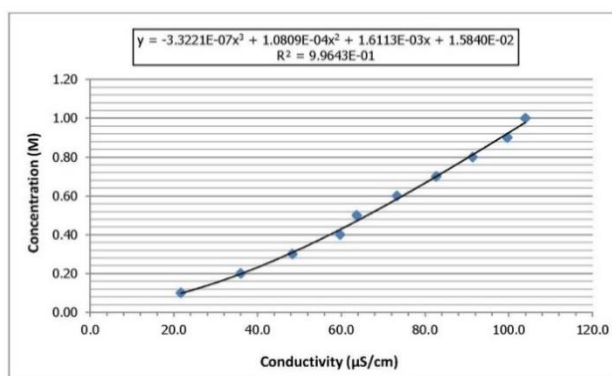
F.5 – The curved fitting of molar concentration of NaCl as a function of conductivity.

| Concentration(millimolar)	Conductivity(μS/cm)
100.00	8530.0
90.00	7540.0
80.00	6780.0
70.00	5740.0
60.00	4930.0
50.00	3880.0
40.00	3060.0
30.00	2010.0
20.00	1140.0
10.00	1029.0

$y = 8.6445E\text{-}11x^3 - 1.3814E\text{-}06x^2 + 1.7566E\text{-}02x - 2.2728E\text{+}00$
$R^2 = 9.9475E\text{-}01$
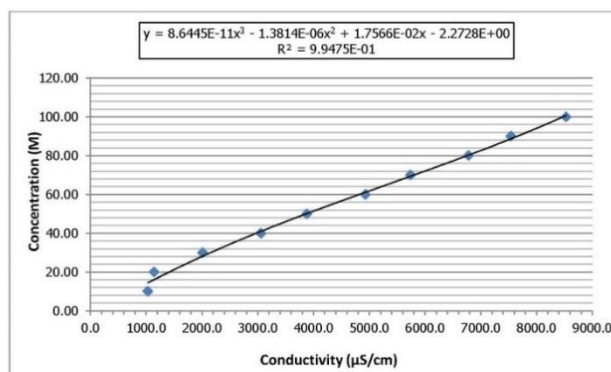
Concentration(millimolar)	Conductivity(μS/cm)
9.00	927.0
8.00	829.0
7.00	705.0
6.00	605.0
5.00	470.0
4.00	371.0
3.00	239.0
2.00	124.1
1.00	111.2

$y = 7.6712E\text{-}09x^3 - 1.2898E\text{-}05x^2 + 1.5227E\text{-}02x - 1.2225E\text{-}01$
$R^2 = 9.9328E\text{-}01$
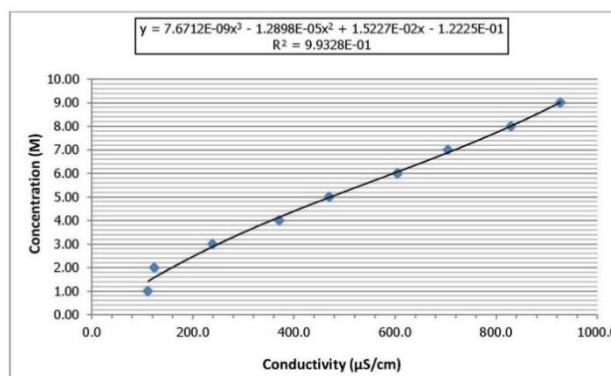
Concentration(millimolar)	Conductivity(μS/cm)
0.90	101.5
0.80	87.5
0.70	78.7
0.60	64.0
0.50	51.4
0.40	41.3
0.30	28.1
0.20	15.2
0.10	13.7

$y = 4.0820E\text{-}07x^3 - 8.3112E\text{-}05x^2 + 1.3428E\text{-}02x - 2.8518E\text{-}02$
$R^2 = 9.9309E\text{-}01$
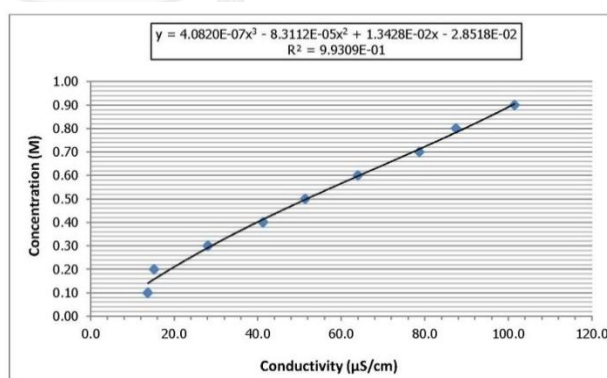
Concentration(millimolar)	Conductivity(μS/cm)
0.09	13.0
0.08	11.4
0.07	10.3
0.06	8.4
0.05	7.2
0.04	6.1
0.03	4.7
0.02	3.9
0.01	2.1

$y = -1.6591E\text{-}05x^3 + 2.9345E\text{-}04x^2 + 6.2769E\text{-}03x - 5.2565E\text{-}03$
$R^2 = 9.9682E\text{-}01$
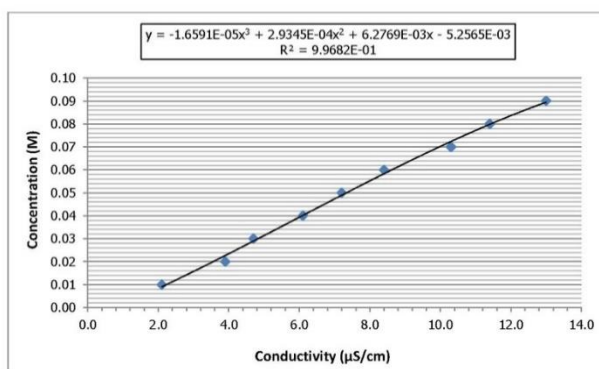
F.6 – The curved fitting of molar concentration of NH_4Cl as a function of conductivity.

Concentration(Molar)	Conductivity(µS/cm)
0.10	13160.0
0.20	23300.0
0.30	35300.0
0.40	44600.0
0.50	56166.7
0.60	64766.7
0.70	76266.7
0.80	85066.7
0.90	96066.7
1.00	104400.0

$y = 3.3840E\text{-}17x^3 + 2.0145E\text{-}12x^2 + 9.1513E\text{-}06x - 1.9663E\text{-}02$
$R^2 = 9.9955E\text{-}01$
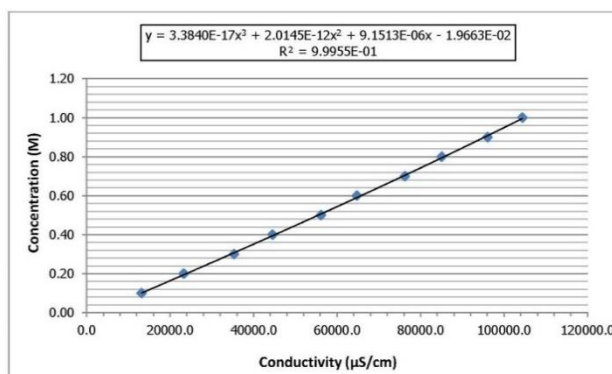
Concentration(millimolar)	Conductivity(µS/cm)
10.00	588.0
9.00	554.0
8.00	492.0
7.00	421.0
6.00	377.0
5.00	316.0
4.00	253.0
3.00	179.8
2.00	123.4
1.00	72.5

$y = 2.0563E\text{-}08x^3 - 1.8014E\text{-}05x^2 + 2.0966E\text{-}02x - 3.9309E\text{-}01$
$R^2 = 9.9845E\text{-}01$
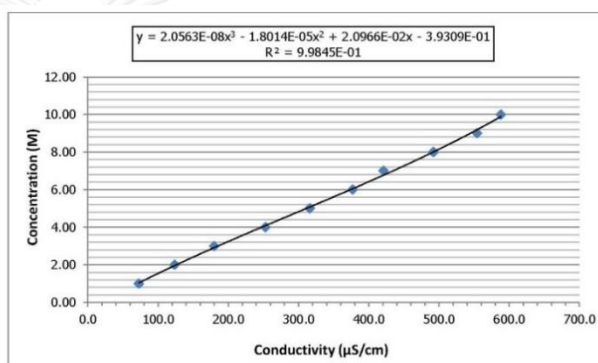
Concentration(millimolar)	Conductivity(µS/cm)
1.00	60.8
0.90	55.6
0.80	50.4
0.70	42.5
0.60	36.0
0.50	32.7
0.40	28.9
0.30	22.5
0.20	15.8
0.10	7.8

$y = -3.6083E\text{-}06x^3 + 3.8795E\text{-}04x^2 + 5.4236E\text{-}03x + 3.1626E\text{-}02$
$R^2 = 9.9559E\text{-}01$
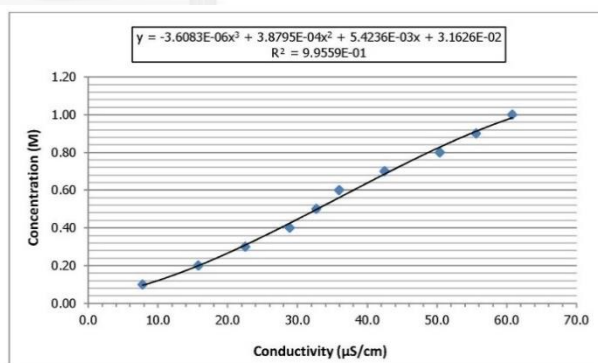
Appendix G MATLAB Code

G.1 Acetic acid as feed solution and NH$_4$Cl as draw solution

```
 1 clc
 2 clear
 3 [g]=[1.608e-03     2.3629e+01 2.748e-01     4.033e-05 5.263e-01     2.341e-04↙
6.598e-11    1.013e-02    3.656e-08...
 4 3.531e-08       3.5274e-08 7.339e-01     7.339e-01 5.263e-01     9.997e-01     4.245e-04↙
9.712e-01    9.712e-01...
 5 6.586e-04       9.9100e-01 1.880e-01     6.372e+02 3.700e+02     2.894e+04     1.503e+02↙
1.012e+03    4.839e+00...
 6 1.581e+00       1.1698e-05 1.724e-01     1.000e-02 6.456e+02     2.759e+04     1.502e+02↙
1.579e+00    4.285e+02...
 7 1.843e-01       1.4116e-05 1.001e-02     3.388e-06 3.075e-08     9.993e-01     9.767e-01↙
9.767e-01    4.193e-04...
 8 9.767e-01       4.2273e-04 5.621e-12     6.245e+02 3.879e+02     1.367e-04     4.193e-04↙
9.767e-01    9.767e-01...
 9 1.657e-08       2.1502e-15 9.099e-01     3.388e-06 3.137e-11     9.5820e-03     9.9624↙
e+02 3.3877e+00    9.7668e-01];
10 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
11 ts=0.2:0.2:1800;
12 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
13 for ti=1:numel(ts);
14 t=ts(ti);
15 options = optimset('Algorithm','Levenberg-Marquardt','TolFun',1*10^-13,'TolX',1*10^-↙
13 ...
16     ,'Maxiter',25,'Maxfun',15000);
17
18 [r,fval,exitflag,output] = fsolve(@(z)Acetic_NH4Cl(z,t),[g],options)
19
20 j(ti)=r(1,1);
21 c(ti)=r(1,31);
22 d(ti)=r(1,51);
23 p(ti)=r(1,62);
24 T(ti)=t;
25 e(ti)=r(1,52);
26 % For Rejection
27 O(ti)=(1-r(1,10)/r(1,31))*100;
28 K4(ti)=r(1,31)/0.01;
29 K1(ti)=r(1,1);
30 [g]=[r];
31 %for Plotting
32 [A(ti)]=t;
33 [B(ti)]=d(ti);
34 [C(ti)]=p(ti);
35 % For Rejection
36 [M(ti)]=O(ti);
37 [N(ti)]=K1(ti);
38 [K5(ti)]=K4(ti); %Cfa/Cf0
39 disp(ti)
40 fprintf('\n    Time(min)        J_w(L/dm2/min)         pH         Cf,a(mole/l)↙
W(kg)\n');
41
42 fprintf('\n%11.4f%20.4e%19.4f%19.4e%19.4e\n' ,T(ti),j(ti),p(ti),c(ti),d(ti));
43
44 format short e
45
46 %Result of Best guess
47 fprintf('\n%8.3e%15.4e%10.3e%14.3e%11.3e%13.3e%13.3e%13.3e%13.3e\n'...
```

```
48      ,r(1,1),r(1,2),r(1,3),r(1,4),r(1,5),r(1,6),r(1,7),r(1,8),r(1,9));
49
50  fprintf('\n%8.3e%15.4e%10.3e%14.3e%11.3e%13.3e%13.3e%13.3e%\n'...
51      ,r(1,10),r(1,11),r(1,12),r(1,13),r(1,14),r(1,15),r(1,16),r(1,17),r(1,18));
52
53  fprintf('\n%8.3e%15.4e%10.3e%14.3e%11.3e%13.3e%13.3e%13.3e%13.3e%\n'...
54      ,r(1,19),r(1,20),r(1,21),r(1,22),r(1,23),r(1,24),r(1,25),r(1,26),r(1,27));
55
56  fprintf('\n%8.3e%15.4e%10.3e%14.3e%11.3e%13.3e%13.3e%13.3e%13.3e%\n'...
57      ,r(1,28),r(1,29),r(1,30),r(1,31),r(1,32),r(1,33),r(1,34),r(1,35),r(1,36));
58
59  fprintf('\n%8.3e%15.4e%10.3e%14.3e%11.3e%13.3e%13.3e%13.3e%13.3e%13.3e%\n'...
60      ,r(1,37),r(1,38),r(1,39),r(1,40),r(1,41),r(1,42),r(1,43),r(1,44),r(1,45));
61
62  fprintf('\n%8.3e%15.4e%10.3e%14.3e%11.3e%13.3e%13.3e%13.3e%13.3e%\n'...
63      ,r(1,46),r(1,47),r(1,48),r(1,49),r(1,50),r(1,51),r(1,52),r(1,53),r(1,54));
64
65  fprintf('\n%8.3e%15.4e%10.3e%14.3e%13.3e%13.4e%13.4e%13.4e%13.4e%\n'...
66      ,r(1,55),r(1,56),r(1,57),r(1,58),r(1,59),r(1,60),r(1,61),r(1,62),r(1,63));
67  end
68
69  %Time Interval%%%%%%%%%%%%%%%%%%%%%%%%%%%%
70  Ai=0:30:1800;
71  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
72  format short
73  Ci=spline(A,C,Ai);
74  format short e
75  Bi=spline(A,B,Ai);
76
77  F=[ 0      0.0150     0.0310     0.0470     0.0630     0.0780     0.0920     0.1060     0.1210
0.1340     0.1470     0.1600...
78  0.1730     0.1860     0.1980     0.2100     0.2220     0.2330     0.2450     0.2610
0.2720     0.2830     0.2940     0.3050...
79  0.3150     0.3260     0.3360     0.3460     0.3560     0.3660     0.3750     0.3840
0.3940     0.4030     0.4120     0.4210...
80  0.4300     0.4390     0.4470     0.4560     0.4640     0.4730     0.4850     0.4930
0.5010     0.5080     0.5160     0.5240...
81  0.5310     0.5390     0.5460     0.5530     0.5600     0.5670     0.5740     0.5810
0.5880     0.5950     0.6010     0.6060...
82  0.611];
83
84  G=[3.4000     3.3950     3.3930     3.3900     3.3850     3.3840     3.3820     3.3780
3.3740     3.3730     3.3690     3.3660...
85  3.3640     3.3600     3.3570     3.3550     3.3520     3.3490     3.3470     3.3440
3.3410     3.3390     3.3370     3.3340...
86  3.3330     3.3290     3.3270     3.3240     3.3220     3.3190     3.3160     3.3140
3.3110     3.3090     3.3060     3.3030...
87  3.2980     3.2940     3.2910     3.2870     3.2840     3.2800     3.2770     3.2740
3.2730     3.2690     3.2670     3.2640...
88  3.2620     3.2600     3.2580     3.2590     3.2570     3.2550     3.2530     3.2520
3.2510     3.2500     3.2490     3.2480...
89  3.247];
90  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
91  F_m=mean(F);
92  G_m=mean(G);
93
```

```matlab
 94 sum1=0;sum2=0;
 95 sum3=0;sum4=0;
 96 %%% Number of Data %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 97 for k=1:1:61
 98 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 99 sum1=(Bi(1,k)-F(1,k))^2+sum1;
100 sum2=(Bi(1,k)-F_m)^2+sum2;
101 sum3=(Ci(1,k)-G(1,k))^2+sum3;
102 sum4=(Ci(1,k)-G_m)^2+sum4;
103 end
104
105 format short
106 R2_W=1-sum1/sum2;
107 R2_pH=1-sum3/sum4;
108 MSE_W=sum1/k;
109 MSE_pH=sum3/k;
110 RMSE_W=sqrt(MSE_W)
111 RMSE_pH=sqrt(MSE_pH)
112 SEP_W=RMSE_W/F_m*100
113 SEP_pH=RMSE_pH/G_m*100
114
115 % Plotting
116 subplot(2,3,1)
117 plot([A]/60,[B],'-g',[Ai]/60,[F],'xr')
118 xlabel('Time (hr)')
119 ylabel('Weight (Kg)')
120 title('Weight Change as a Function of Time')
121 legend('Model','Experiment',2)
122
123 subplot(2,3,2)
124 plot([A]/60,[C],'-g',[Ai]/60,[G],'xr')
125 xlabel('Time (hr)')
126 ylabel('pH')
127 title('pH as a Function of Time')
128 legend('Model','Experiment',1)
129
130 subplot(2,3,3)
131 plot([A]/60,[M],'-g')
132 xlabel('Time (hr)')
133 ylabel('%Rejection')
134 title('%Rejection as a Function of Time')
135 legend('Model',1)
136
137 subplot(2,3,4)
138 plot([N],[M],'-g')
139 xlabel('Jw (L/dm2/min)')
140 ylabel('%Rejection')
141 title('%Rejection as a Function of Jw')
142 legend('Model',1)
143
144 subplot(2,3,5)
145 plot([A]/60,[K5],'-g')
146 xlabel('Time (hr)')
147 ylabel('Cfa/Cf0')
148 title('Cfa/Cf0 as a Function of Time')
149 legend('Model',1)
```

```matlab
150
151 subplot(2,3,6)
152 plot([N],[K5],'-g')
153 xlabel('Jw (L/dm2/min)')
154 ylabel('Cfa/Cf0')
155 title('Cfa/Cf0 as a Function of Jw')
156 legend('Model',1)
157
158 fprintf('\n     \n')
159 fprintf('\n     \n')
160 fprintf('\n CONCLUSION    \n')
161 fprintf('\n       R2_W          MSE_W                    R2_pH          MSE_pH\n');
162 fprintf('\n%11.3f%20.4e%19.4f%19.4e%\n',R2_W,MSE_W,R2_pH,MSE_pH)
163 fprintf('\n     \n')
164
165 fprintf('\n     Time(min)       J_w(L/dm2/min)            pH           Cf,a(mole/l) ⬎
W(kg)        C_f_H\n');
166 for x=5:5:200000000;
167 z=x*4;
168 fprintf('\n%11.3f%20.4e%19.4f%19.4e%19.4e%19.4e%\n',T(z),j(z),p(z),c(z),d(z),e(z));
169 end
170
171
```

```matlab
1
2  function fu=Acetic_NH4Cl(z,t)
3  %Independent Variable
4  V_f_0=1;C_d_s_0=1;C_f_a_0=10*10^(-3);V_d_0=0.5;
5  %Fixed Variable
6  T=301;f=1.58;
7  L=0.9207;D=0.023;W=0.4572;A_m=42*10^(-2);d_h=0.0438;
8  %Constant Variable
9  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
10 B_a=2.102/60/100;
11 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
12 B_s=0.460/60/100;
13 A=0.413/60/100;
14 o_s=0.897;%amonium chloride coefficient
15 S=491*10^-5;
16 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
17 D_NH4_298=117.42e-7;D_Cl_298=121.92e-7;%amonium & chloride diffusion coefficient
18 den_w=999.65+2.0438/10*(T-273)-6.174*10^-2*(T-273)^1.5;
19 D_A_298=6.534*10^(-6);D_HA_298=7.5429*10^(-6);%Acetic Physicocmical Property
20 D_H_298=5.587*10^(-5);
21 visco_298=((298-273)+246)/((0.05594*(298-273)+5.2842)*(298-273)+137.37);
22 visco_T=((T-273)+246)/((0.05594*(T-273)+5.2842)*(T-273)+137.37);
23 mul_f=visco_T*6;
24 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
25 o_a=1;n_s=2;R=0.08314;
26 %Temperature Correction
27 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
28 K_a_T=10^-(1170.48/T-3.1649+0.013399*T);%Acetic acid
29 K_CO2_T=10^-(3404.71/T-14.8435+0.032786*T);
30 K_HCO3_T=10^-(2902.39/T-6.4980+0.02379*T);
31 K_w_T=10^(-4470.99/T+6.0875-0.017060*T);
32 K_NH4CL=10^-(2835.76/T-0.6322+0.001225*T);%Amonium Chloride
33 E=2727.586+0.6224107*T-466.9151*log(T)-52000.87/T;
34 d=1-(((T-273)-3.9863)^2*((T-273)+288.9414))/(508929.2*((T-273)+68.12963))...
35 +0.011445*exp((-374.3)/(T-273));
36 A_I=(1.82483*10^6*d^0.5)/(E*T)^1.5;
37 K=1/K_a_T;
38
39 D_H_T=D_H_298*T/298*visco_298/visco_T ;
40 D_A_T=D_A_298*T/298*visco_298/visco_T;
41 D_HA_T=D_HA_298*T/298*visco_298/visco_T;
42 D_NH4_T=D_NH4_298*T/298*visco_298/visco_T;
43 D_Cl_T=D_Cl_298*T/298*visco_298/visco_T;
44
45 D_1_T=2/(1/D_H_T +1/D_A_T);
46 D_S_T=2/(1/D_NH4_T +1/D_Cl_T);
47
48 D_d_s=D_S_T;
49 D_f_s=D_S_T;
50
51 K_H=0.034*exp(-2400*(1/T-1/298));%************************
52 C_f_co2=10^-3.408*K_H;%***************************
53 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
54 %Unknown Variable
55 J_w=z(1);P_i=z(2);P_m=z(3);J_s=z(4);C_i_s=z(5);
56 C_mf_s=z(6);J_a=z(7);C_mf_a=z(8);C_i_a=z(9);C_d_a=z(10);
```

```matlab
57  C_d_H=z(11);gamma_d_H=z(12);gamma_d_A=z(13);I_i=z(14);C_d_s=z(15);
58  C_mf_H=z(16);gamma_m_H=z(17);gamma_m_A=z(18);I_m=z(19);C_md_s=z(20);
59  k_d=z(21);Sh_d=z(22);Sc_d=z(23);Re_d=z(24);
60  v_d=z(25);den_d=z(26);mul_d=z(27);f_2=z(28);D_f_a=z(29);
61  k_f_a=z(30);C_f_a=z(31);Sh_f_a=z(32);Re_f=z(33);v_f=z(34);
62  f_1=z(35);Sc_f_a=z(36);k_f_s=z(37);C_f_s=z(38);
63  C_f_ai=z(39);C_f_si=z(40);C_d_ai=z(41);C_d_si=z(42);gamma_f_Ac=z(43);
64  gamma_f_H=z(44);C_f_Ac=z(45);gamma_f_A=z(46);I_f=z(47);C_f_NH3=z(48);
65  Sh_f_s=z(49);Sc_f_s=z(50);w=z(51);C_f_H=z(52);gamma_f_HCO3=z(53);
66  gamma_f_OH=z(54);C_f_HCO3=z(55);C_f_CO3=z(56);
67  gamma_f_CO3=z(57);C_f_NH4=z(58);C_f_OH=z(59);C_f_HA=z(60);den_f=z(61);
68  pH_f=z(62);gamma_f_NH4=z(63);
69
70  format short e
71  %Math Model
72  fu(1)=A*(P_i-P_m )-J_w;
73  fu(2)= B_s *(C_i_s-C_mf_s)-J_s;
74  fu(3)= B_a*(C_mf_a-C_i_a)-J_a;
75  fu(4)=C_d_a-C_i_a;
76  fu(5)=J_a/J_w-C_d_a;
77  fu(6)= o_s*n_s*C_i_s*R*T+o_a*(C_d_a+C_d_H)*R*T-P_i;
78  fu(7)=(-K_a_T+((K_a_T)^2+4*K_a_T*gamma_d_H*gamma_d_A*C_d_a)^(1/2))...
79      /(2*gamma_d_H*gamma_d_A)-C_d_H;%*****************
80  fu(8)=A_I*(I_i^(1/2)/(1+I_i^(1/2))- 0.3*I_i)+log10(gamma_d_H);
81  fu(9)=A_I*(I_i^(1/2)/(1+I_i^(1/2))- 0.3*I_i)+log10(gamma_d_A);
82  fu(10)=0.5*(2*C_i_s+2*C_d_H)- I_i;
83  fu(11)=o_s*n_s*C_mf_s*R*T+o_a*(C_mf_a+C_mf_H)*R*T-P_m;
84  fu(12)=(-K_a_T+((K_a_T)^2+4*K_a_T*gamma_m_H*gamma_m_A*C_mf_a)^(1/2))...
85      /(2*gamma_m_H*gamma_m_A)-C_mf_H;%*****************
86  fu(13)=A_I*(I_m^(1/2)/(1+I_m^(1/2))- 0.3*I_m)+log10(gamma_m_H);
87  fu(14)=A_I*(I_m^(1/2)/(1+I_m^(1/2))- 0.3*I_m)+log10(gamma_m_A);
88  fu(15)=0.5*(2*C_mf_s+2*C_mf_H)-I_m;
89  fu(16)=(C_md_s-C_i_s)/2*D_S_T*(1/(J_w*(0.789*C_i_s+0.211*C_md_s)+J_s)...
90  +1/(J_w*(0.211*C_i_s+0.789*C_md_s)+ J_s))-S;
91  fu(17)=(C_d_s-C_md_s)/2*D_S_T*(1/(J_w*(0.789*C_md_s+0.211*C_d_s)+J_s)...
92  +1/(J_w*(0.211*C_md_s+0.789*C_d_s)+ J_s))-D_d_s/k_d;
93  fu(18)=Sh_d*D_d_s/d_h-k_d;
94  fu(19)=L*v_d*den_d/mul_d-Re_d;
95  fu(20)=(f_2+f)/2/W/D-v_d;
96  fu(21)=f+J_w*A_m-f_2;
97  % amonium chloride density and viscosity
98  fu(22)=6*exp((12.396*(C_d_s*53.491/den_d)^1.5039-1.7756)/(0.23471*(T-273)+1)/...
99  (-2.7591*(C_d_s*53.491/den_d)^2.8408+1))-mul_d;
100
101 fu(23)=den_w+0.2061e2*C_d_s-0.1577*C_d_s*(T-273)+1.553e-3*C_d_s*(T-273)^2-...
102 2.556*C_d_s^1.5+5.67e-2*C_d_s^1.5*(T-273)-5.082e-4*C_d_s^1.5*(T-273)^2-den_d;
103 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
104 fu(24)= 0.04*Re_d^(3/4)*Sc_d^(1/3)-Sh_d;
105 fu(25)= mul_d/den_d/D_d_s-Sc_d;
106 fu(26)=(0.5*(C_f_a-C_mf_a)*((D_1_T-D_HA_T)/...
107     (sqrt(1+4*K*(0.789*C_mf_a+0.211*C_f_a))*(J_w*(0.789*C_mf_a+0.211*C_f_a)-J_a))...
108     +(D_1_T-D_HA_T)/(sqrt(1+4*K*(0.211*C_mf_a+0.789*C_f_a))*(J_w*(0.211*C_mf_a...
109     +0.789*C_f_a)-J_a)+ D_HA_T/(J_w*(0.789*C_mf_a+0.211*C_f_a)-J_a)...
110     +D_HA_T/((J_w*(0.211*C_mf_a+0.789*C_f_a)-J_a))))+D_f_a/k_f_a;
111 fu(27)=Sh_f_a*D_f_a/d_h-k_f_a;
112 fu(28)=(D_1_T/(2*K*C_f_a)*(-1+sqrt(1+4*K*C_f_a))+D_HA_T/...
```

```matlab
113      (4*K*C_f_a)*(-1+sqrt(1+4*K*C_f_a))^2)-D_f_a;
114 fu(29)=L*v_f*den_f/mul_f -Re_f;
115 fu(30)=(f+f_1)/W/D/2-v_f;
116 fu(31)=f-J_w*A_m-f_1;
117 fu(32)= 0.04*Re_f^(3/4)*Sc_f_a^(1/3)-Sh_f_a;
118 fu(33)= mul_f/den_f/D_f_a-Sc_f_a;
119 fu(34)=(C_f_s-C_mf_s)/2*D_S_T*(1/(J_w*(0.789*C_mf_s+0.211*C_f_s)+J_s)...
120 +1/(J_w*(0.211*C_mf_s+0.789*C_f_s)+ J_s))+D_f_s/k_f_s;
121 fu(35)=Sh_f_s*D_f_s/d_h-k_f_s;
122 %Mole balance Equation
123 fu(36)=C_f_a*(V_f_0+f_1*t-f*t)-(V_f_0*C_f_a_0+C_f_ai*f_1*t-C_f_a*f*t);
124 fu(37)=C_f_si*(V_f_0+f_1*t-f*t)-C_f_s*f_1*t+C_f_si*f*t;
125 fu(38)=C_f_a*f-J_a*A_m-C_f_ai*f_1;
126 fu(39)=C_f_si*f+J_s*A_m-C_f_s*f_1;
127
128 fu(40)=C_d_s_0*V_d_0+C_d_si*f_2*t-C_d_s*f*t-C_d_s*(V_d_0+f_2*t-f*t);
129 fu(41)=C_d_a*f_2*t-C_d_ai*f*t-C_d_ai*(V_d_0+f_2*t-f*t);
130 fu(42)=C_d_s*f-J_s*A_m-C_d_si*f_2;
131 fu(43)=C_d_ai*f+J_a*A_m-C_d_a*f_2;
132 % Adjust Part
133 %additional Equation
134 fu(44)= 0.04*Re_f^(3/4)*Sc_f_s^(1/3)-Sh_f_s;
135 fu(45)= mul_f/den_f/D_f_s-Sc_f_s;
136 fu(46)=J_w*A_m*t*den_d/1000-w;
137 %pH Equation..
138 %ionic strenght in feed solution%%%%%%%%%%%%%%%%%%%%%
139 fu(47)=0.5*(C_f_si+C_f_NH4+C_f_H+C_f_OH+C_f_Ac+C_f_HCO3+4*C_f_CO3)- I_f;
140 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
141 fu(48)=A_I*(I_f^(1/2)/(1+I_f^(1/2))- 0.3*I_f)+log10(gamma_f_H);
142 fu(49)=A_I*(I_f^(1/2)/(1+I_f^(1/2))- 0.3*I_f)+log10(gamma_f_A);
143 fu(50)=A_I*(I_f^(1/2)/(1+I_f^(1/2))- 0.3*I_f)+log10(gamma_f_HCO3);
144 fu(51)=A_I*(I_f^(1/2)/(1+I_f^(1/2))- 0.3*I_f)+log10(gamma_f_OH);
145 %Charge Balance Equation %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
146 fu(52)=C_f_HCO3+2*C_f_CO3+C_f_Ac+C_f_OH+C_f_si-C_f_NH4-C_f_H;
147 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
148 %New Equation for charge balance
149 fu(53)=K_a_T*C_f_HA/gamma_f_H/gamma_f_Ac/C_f_H-C_f_Ac;
150 fu(54)=K_CO2_T/gamma_f_H/gamma_f_HCO3/C_f_H*(C_f_co2-C_f_HCO3)-C_f_HCO3;%******
151 fu(55)=K_HCO3_T*gamma_f_HCO3/gamma_f_H/gamma_f_CO3/C_f_H*(C_f_HCO3-C_f_CO3)-C_f_CO3;%↙
******
152 fu(56)=A_I*4*(I_f^(1/2)/(1+I_f^(1/2))- 0.3*I_f)+log10(gamma_f_CO3);
153 fu(57)=A_I*(I_f^(1/2)/(1+I_f^(1/2))- 0.3*I_f)+log10(gamma_f_Ac);
154 fu(58)=K_w_T/C_f_H/gamma_f_OH/gamma_f_H-C_f_OH;
155 fu(59)=C_f_NH4+C_f_NH3-C_f_si;
156 fu(60)=C_f_HA+C_f_Ac-C_f_a;
157 fu(61)=gamma_f_H*C_f_H*C_f_NH3/gamma_f_NH4/K_NH4CL-C_f_NH4;
158 fu(62)=pH_f+log10(gamma_f_H*C_f_H);
159 fu(63)=A_I*(I_f^(1/2)/(1+I_f^(1/2))- 0.3*I_f)+log10(gamma_f_NH4);
160
161
162
```

G.2 Butyric acid as feed solution and NH_4Cl as draw solution

```
 1 clc
 2 clear
 3 [g]=[1.608e-03    2.3632e+01 2.752e-01    4.035e-05 5.264e-01    2.325e-04↙
1.658e-11    1.019e-02    8.682e-09...
 4 8.567e-09    1.1650e-08 7.339e-01    7.339e-01 5.264e-01    9.999e-01    3.940e-04↙
9.718e-01    9.718e-01...
 5 6.264e-04    9.9113e-01 1.880e-01    6.372e+02 3.700e+02    2.894e+04    1.503e+02↙
1.012e+03    4.839e+00...
 6 1.581e+00    1.1694e-05 1.724e-01    1.000e-02 6.457e+02    2.759e+04    1.502e+02↙
1.579e+00    4.287e+02...
 7 1.843e-01    1.2425e-05 1.000e-02    1.695e-06 7.604e-09    9.994e-01    9.776e-01↙
9.776e-01    3.878e-04...
 8 9.776e-01    3.8955e-04 3.039e-12    6.245e+02 3.879e+02    6.835e-05    3.879e-04↙
9.776e-01    9.776e-01...
 9 1.243e-08    9.0797e-15 9.133e-01    1.695e-06    3.386e-11    9.6128e-03    9.9624↙
e+02    3.4212e+0];
10 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
11 ts=0.1:0.1:1800;
12 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
13 for ti=1:numel(ts);
14 t=ts(ti);
15 options = optimset('Algorithm','Levenberg-Marquardt','TolFun',1*10^-13,'TolX',1*10^-↙
13 ...
16     ,'Maxiter',25,'Maxfun',15000);
17
18 [r,fval,exitflag,output] = fsolve(@(z)Butyric_NH4Cl(z,t),[g],options)
19
20 j(ti)=r(1,1);
21 c(ti)=r(1,31);
22 d(ti)=r(1,51);
23 p(ti)=r(1,62);
24 T(ti)=t;
25 e(ti)=r(1,52);
26 % For Rejection
27 O(ti)=(1-r(1,10)/r(1,31))*100;
28 K4(ti)=r(1,31)/0.01;
29 K1(ti)=r(1,1);
30 [g]=[r];
31 %for Plotting
32 [A(ti)]=t;
33 [B(ti)]=d(ti);
34 [C(ti)]=p(ti);
35 % For Rejection
36 [M(ti)]=O(ti);
37 [N(ti)]=K1(ti);
38 [K5(ti)]=K4(ti); %Cfa/Cf0
39 disp(ti)
40 fprintf('\n    Time(min)        J_w(L/dm2/min)          pH          Cf,a(mole/l)↙
W(kg)\n');
41
42 fprintf('\n%11.4f%20.4e%19.4f%19.4e%19.4e%\n' ,T(ti),j(ti),p(ti),c(ti),d(ti));
43
44 format short e
45
46 %Result of Best guess
47 fprintf('\n%8.3e%15.4e%10.3e%14.3e%11.3e%13.3e%13.3e%13.3e%13.3e%\n'...
```

```matlab
48     ,r(1,1),r(1,2),r(1,3),r(1,4),r(1,5),r(1,6),r(1,7),r(1,8),r(1,9));
49
50  fprintf('\n%8.3e%15.4e%10.3e%14.3e%11.3e%13.3e%13.3e%13.3e%13.3e%\n'...
51     ,r(1,10),r(1,11),r(1,12),r(1,13),r(1,14),r(1,15),r(1,16),r(1,17),r(1,18));
52
53  fprintf('\n%8.3e%15.4e%10.3e%14.3e%11.3e%13.3e%13.3e%13.3e%13.3e%\n'...
54     ,r(1,19),r(1,20),r(1,21),r(1,22),r(1,23),r(1,24),r(1,25),r(1,26),r(1,27));
55
56  fprintf('\n%8.3e%15.4e%10.3e%14.3e%11.3e%13.3e%13.3e%13.3e%13.3e%\n'...
57     ,r(1,28),r(1,29),r(1,30),r(1,31),r(1,32),r(1,33),r(1,34),r(1,35),r(1,36));
58
59  fprintf('\n%8.3e%15.4e%10.3e%14.3e%11.3e%13.3e%13.3e%13.3e%13.3e%13.3e%\n'...
60     ,r(1,37),r(1,38),r(1,39),r(1,40),r(1,41),r(1,42),r(1,43),r(1,44),r(1,45));
61
62  fprintf('\n%8.3e%15.4e%10.3e%14.3e%11.3e%13.3e%13.3e%13.3e%13.3e%\n'...
63     ,r(1,46),r(1,47),r(1,48),r(1,49),r(1,50),r(1,51),r(1,52),r(1,53),r(1,54));
64
65  fprintf('\n%8.3e%15.4e%10.3e%14.3e%13.3e%13.4e%13.4e%13.4e%13.4e%\n'...
66     ,r(1,55),r(1,56),r(1,57),r(1,58),r(1,59),r(1,60),r(1,61),r(1,62),r(1,63));
67  end
68
69  %Time Interval%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
70  Ai=0:30:1800;
71  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
72  format short
73  Ci=spline(A,C,Ai);
74  format short e
75  Bi=spline(A,B,Ai);
76
77  F=[ 0     0.0160    0.0330    0.0480    0.0640    0.0790    0.0930    0.1070    0.1210↙
0.1340    0.1470...
78  0.1600    0.1730    0.1850    0.1980    0.2090    0.2210    0.2320    0.2440↙
0.2610    0.2720    0.2830...
79  0.2940    0.3040    0.3140    0.3240    0.3350    0.3440    0.3540    0.3630↙
0.3730    0.3820    0.3910...
80  0.4000    0.4090    0.4180    0.4270    0.4350    0.4440    0.4520    0.4600↙
0.4680    0.4810    0.4890...
81  0.4970    0.5050    0.5120    0.5190    0.5270    0.5340    0.5410    0.5480↙
0.5550    0.5610    0.5680...
82  0.5740    0.5810    0.5870    0.5940    0.6000    0.606];
83
84  G=[ 3.4431    3.4440    3.4428    3.4445    3.4426    3.4408    3.4376    3.4337↙
3.4298    3.4266    3.4228    3.4189    3.4145...
85  3.4092    3.4047    3.3995    3.3956    3.3913    3.3875    3.3837    3.3791↙
3.3761    3.3739    3.3716    3.3687    3.3664...
86  3.3642    3.3613    3.3591    3.3569    3.3539    3.3518    3.3489    3.3467↙
3.3453    3.3424    3.3395    3.3373    3.3352...
87  3.3331    3.3302    3.3281    3.3252    3.3231    3.3202    3.3174    3.3145↙
3.3117    3.3096    3.3068    3.3047    3.3019...
88  3.2999    3.2978    3.2965    3.2937    3.2924    3.2897    3.2877    3.2857↙
3.2837];
89
90  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
91  F_m=mean(F);
92  G_m=mean(G);
93
```

```matlab
 94 sum1=0;sum2=0;
 95 sum3=0;sum4=0;
 96 %%% Number of Data %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 97 for k=1:1:61
 98 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 99 sum1=(Bi(1,k)-F(1,k))^2+sum1;
100 sum2=(Bi(1,k)-F_m)^2+sum2;
101 sum3=(Ci(1,k)-G(1,k))^2+sum3;
102 sum4=(Ci(1,k)-G_m)^2+sum4;
103 end
104
105 format short
106 R2_W=1-sum1/sum2;
107 R2_pH=1-sum3/sum4;
108 MSE_W=sum1/k;
109 MSE_pH=sum3/k;
110 RMSE_W=sqrt(MSE_W)
111 RMSE_pH=sqrt(MSE_pH)
112 SEP_W=RMSE_W/F_m*100
113 SEP_pH=RMSE_pH/G_m*100
114
115 % Plotting
116 subplot(2,3,1)
117 plot([A]/60,[B],'-g',[Ai]/60,[F],'xr')
118 xlabel('Time (hr)')
119 ylabel('Weight (Kg)')
120 title('Weight Change as a Function of Time')
121 legend('Model','Experiment',2)
122
123 subplot(2,3,2)
124 plot([A]/60,[C],'-g',[Ai]/60,[G],'xr')
125 xlabel('Time (hr)')
126 ylabel('pH')
127 title('pH as a Function of Time')
128 legend('Model','Experiment',1)
129
130 subplot(2,3,3)
131 plot([A]/60,[M],'-g')
132 xlabel('Time (hr)')
133 ylabel('%Rejection')
134 title('%Rejection as a Function of Time')
135 legend('Model',1)
136
137 subplot(2,3,4)
138 plot([N],[M],'-g')
139 xlabel('Jw (L/dm2/min)')
140 ylabel('%Rejection')
141 title('%Rejection as a Function of Jw')
142 legend('Model',1)
143
144 subplot(2,3,5)
145 plot([A]/60,[K5],'-g')
146 xlabel('Time (hr)')
147 ylabel('Cfa/Cf0')
148 title('Cfa/Cf0 as a Function of Time')
149 legend('Model',1)
```

```
150
151 subplot(2,3,6)
152 plot([N],[K5],'-g')
153 xlabel('Jw (L/dm2/min)')
154 ylabel('Cfa/Cf0')
155 title('Cfa/Cf0 as a Function of Jw')
156 legend('Model',1)
157
158 fprintf('\n      \n')
159 fprintf('\n      \n')
160 fprintf('\n CONCLUSION    \n')
161 fprintf('\n      R2_W          MSE_W                R2_pH          MSE_pH\n');
162 fprintf('\n%11.3f%20.4e%19.4f%19.4e%\n',R2_W,MSE_W,R2_pH,MSE_pH)
163 fprintf('\n      \n')
164
165 fprintf('\n    Time(min)        J_w(L/dm2/min)          pH              Cf,a(mole/l) ↙
W(kg)         C_f_H\n');
166 for x=5:5:200000000;
167 z=x*4;
168 fprintf('\n%11.3f%20.4e%19.4f%19.4e%19.4e%19.4e%\n',T(z),j(z),p(z),c(z),d(z),e(z));
169 end
170
171
172
```

```matlab
 1
 2 function fu=Butyric_NH4Cl(z,t)
 3 %Independent Variable
 4 V_f_0=1;C_d_s_0=1;C_f_a_0=10*10^(-3);V_d_0=0.5;
 5 %Fixed Variable
 6 T=301;f=1.58;
 7 % T=300..normal f=1.58...normal
 8 L=0.9207;D=0.023;W=0.4572;A_m=42*10^(-2);d_h=0.0438;
 9 %Constant Variable
10 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
11 B_a=0.639/60/100;% B of Butyric Acid
12 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
13 A=0.413/60/100;B_s=0.46/60/100;
14 S=491*10^(-5);
15 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
16 o_s=0.897;%amonium chloride coefficient
17 D_NH4_298=117.42e-7;D_Cl_298=121.92e-7;%amonium & chloride diffusion coefficient
18 D_A_298=5.208*10^(-6);D_HA_298=5.429*10^(-6);%Butyric Physicocmical Property
19 D_H_298=5.587*10^(-5);
20 den_w=999.65+2.0438/10*(T-273)-6.174*10^-2*(T-273)^1.5;
21 visco_298=((298-273)+246)/((0.05594*(298-273)+5.2842)*(298-273)+137.37);
22 visco_T=((T-273)+246)/((0.05594*(T-273)+5.2842)*(T-273)+137.37);
23 mul_f=visco_T*6;
24 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
25 n_s=2;o_a=1;R=0.08314;
26 %Temperature Correction
27 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
28 K_a_T=10^-(1033.39/T-2.6215+0.013334*T);%Butyric Acid
29 K_CO2_T=10^-(3404.71/T-14.8435+0.032786*T);
30 C_f_co2=1e-5;
31 K_HCO3_T=10^-(2902.39/T-6.4980+0.02379*T);
32 K_w_T=10^(-4470.99/T+6.0875-0.017060*T);
33 K_NH4CL=10^-(2835.76/T-0.6322+0.001225*T);%Amonium Chloride
34 E=2727.586+0.6224107*T-466.9151*log(T)-52000.87/T;
35 d=1-(((T-273)-3.9863)^2*((T-273)+288.9414))/(508929.2*((T-273)+68.12963))...
36 +0.011445*exp((-374.3)/(T-273));
37 A_I=(1.82483*10^6*d^0.5)/(E*T)^1.5;
38 K=1/K_a_T;
39 D_H_T=D_H_298*T/298*visco_298/visco_T ;
40 D_A_T=D_A_298*T/298*visco_298/visco_T;
41 D_HA_T=D_HA_298*T/298*visco_298/visco_T;
42 D_NH4_T=D_NH4_298*T/298*visco_298/visco_T;
43 D_Cl_T=D_Cl_298*T/298*visco_298/visco_T;
44 D_1_T=2/(1/D_H_T +1/D_A_T);
45 D_S_T=2/(1/D_NH4_T +1/D_Cl_T);
46 D_d_s=D_S_T;
47 D_f_s=D_S_T;
48 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
49 %Unknown Variable
50 J_w=z(1);P_i=z(2);P_m=z(3);J_s=z(4);C_i_s=z(5);
51 C_mf_s=z(6);J_a=z(7);C_mf_a=z(8);C_i_a=z(9);C_d_a=z(10);
52 C_d_H=z(11);gamma_d_H=z(12);gamma_d_A=z(13);I_i=z(14);C_d_s=z(15);
53 C_mf_H=z(16);gamma_m_H=z(17);gamma_m_A=z(18);I_m=z(19);C_md_s=z(20);
54 k_d=z(21);Sh_d=z(22);Sc_d=z(23);Re_d=z(24);
55 v_d=z(25);den_d=z(26);mul_d=z(27);f_2=z(28);D_f_a=z(29);
56 k_f_a=z(30);C_f_a=z(31);Sh_f_a=z(32);Re_f_a=z(33);v_f=z(34);
```

```matlab
57  f_1=z(35);Sc_f_a=z(36);k_f_s=z(37);C_f_s=z(38);
58  C_f_ai=z(39);C_f_si=z(40);C_d_ai=z(41);C_d_si=z(42);gamma_f_Ac=z(43);
59  gamma_f_H=z(44);C_f_Ac=z(45);gamma_f_A=z(46);I_f=z(47);C_f_NH3=z(48);
60  Sh_f_s=z(49);Sc_f_s=z(50);w=z(51);C_f_H=z(52);gamma_f_HCO3=z(53);
61  gamma_f_OH=z(54);C_f_HCO3=z(55);C_f_CO3=z(56);
62  gamma_f_CO3=z(57);C_f_NH4=z(58);C_f_OH=z(59);C_f_HA=z(60);den_f=z(61);
63  pH_f=z(62);
64
65  format short e
66  %Math Model
67  fu(1)=A*(P_i-P_m )-J_w;
68  fu(2)= B_s *(C_i_s-C_mf_s)-J_s;
69  fu(3)= B_a*(C_mf_a-C_i_a)-J_a;
70  fu(4)=C_d_a-C_i_a;
71  fu(5)=J_a/J_w-C_d_a;
72  fu(6)= o_s*n_s*C_i_s*R*T+o_a*(C_d_a+C_d_H)*R*T-P_i;
73  fu(7)=(-(gamma_d_H*K_a_T)+((gamma_d_H*K_a_T)^2+4*K_a_T*gamma_d_H*gamma_d_A*C_d_a)^↙
(1/2))...
74       /(2*gamma_d_H*gamma_d_A)-C_d_H;
75  fu(8)=A_I*(I_i^(1/2)/(1+I_i^(1/2))- 0.3*I_i)+log10(gamma_d_H);%"Davies"
76  fu(9)=A_I*(I_i^(1/2)/(1+I_i^(1/2))- 0.3*I_i)+log10(gamma_d_A);%"Davies"
77  fu(10)=0.5*(2*C_i_s+2*C_d_H)- I_i;
78  fu(11)=o_s*n_s*C_mf_s*R*T+o_a*(C_mf_a+C_mf_H)*R*T-P_m;
79  fu(12)=(-(gamma_m_H*K_a_T)+((gamma_m_H*K_a_T)^2+4*K_a_T*gamma_m_H*gamma_m_A*C_mf_a)^↙
(1/2))...
80       /(2*gamma_m_H*gamma_m_A)-C_mf_H;
81  fu(13)=A_I*(I_m^(1/2)/(1+I_m^(1/2))- 0.3*I_m)+log10(gamma_m_H);%"Davies"
82  fu(14)=A_I*(I_m^(1/2)/(1+I_m^(1/2))- 0.3*I_m)+log10(gamma_m_A);%"Davies"
83  fu(15)=0.5*(2*C_mf_s+2*C_mf_H)-I_m;
84  fu(16)=(C_md_s-C_i_s)/2*D_S_T*(1/(J_w*(0.789*C_i_s+0.211*C_md_s)+J_s)...
85  +1/(J_w*(0.211*C_i_s+0.789*C_md_s)+ J_s))-S;
86  fu(17)=(C_d_s-C_md_s)/2*D_S_T*(1/(J_w*(0.789*C_md_s+0.211*C_d_s)+J_s)...
87  +1/(J_w*(0.211*C_md_s+0.789*C_d_s)+ J_s))-D_d_s/k_d;
88  fu(18)=Sh_d*D_d_s/d_h-k_d;
89  fu(19)=L*v_d*den_d/mul_d-Re_d;
90  fu(20)=(f_2+f)/2/W/D-v_d;
91  fu(21)=f+J_w*A_m-f_2;
92  % amonium chloride density and viscosity
93  fu(22)=6*exp((12.396*(C_d_s*53.491/den_d)^1.5039-1.7756)/(0.23471*(T-273)+1)/...
94  (-2.7591*(C_d_s*53.491/den_d)^2.8408+1))-mul_d;
95  fu(23)=den_w+0.2061e2*C_d_s-0.1577*C_d_s*(T-273)+1.553e-3*C_d_s*(T-273)^2-...
96  2.556*C_d_s^1.5+5.67e-2*C_d_s^1.5*(T-273)-5.082e-4*C_d_s^1.5*(T-273)^2-den_d;
97  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
98  fu(24)= 0.04*Re_d^(3/4)*Sc_d^(1/3)-Sh_d;
99  fu(25)= mul_d/den_d/D_d_s-Sc_d;
100 fu(26)=(0.5*(C_f_a-C_mf_a)*((D_1_T-D_HA_T)/...
101      (sqrt(1+4*K*(0.789*C_mf_a+0.211*C_f_a))*(J_w*(0.789*C_mf_a+0.211*C_f_a)-J_a))...
102      +(D_1_T-D_HA_T)/(sqrt(1+4*K*(0.211*C_mf_a+0.789*C_f_a))*(J_w*(0.211*C_mf_a...
103      +0.789*C_f_a)-J_a)+ D_HA_T/((J_w*(0.789*C_mf_a+0.211*C_f_a)-J_a)...
104      +D_HA_T/((J_w*(0.211*C_mf_a+0.789*C_f_a)-J_a))))+D_f_a/k_f_a;
105 fu(27)=Sh_f_a*D_f_a/d_h-k_f_a;
106 fu(28)=(D_1_T/(2*K*C_f_a)*(-1+sqrt(1+4*K*C_f_a))+D_HA_T/...
107      (4*K*C_f_a)*(-1+sqrt(1+4*K*C_f_a))^2)-D_f_a;
108 fu(29)=L*v_f*den_f/mul_f -Re_f;
109 fu(30)=(f+f_1)/W/D/2-v_f;
110 fu(31)=f-J_w*A_m-f_1;
```

```
111 fu(32)= 0.04*Re_f^(3/4)*Sc_f_a^(1/3)-Sh_f_a;
112 fu(33)= mul_f/den_f/D_f_a-Sc_f_a;
113 fu(34)=(C_f_s-C_mf_s)/2*D_S_T*(1/(J_w*(0.789*C_mf_s+0.211*C_f_s)+J_s)...
114 +1/(J_w*(0.211*C_mf_s+0.789*C_f_s)+ J_s))+D_f_s/k_f_s;
115 fu(35)=Sh_f_s*D_f_s/d_h-k_f_s;
116 %Mole balance Equation
117 fu(36)=C_f_a*(V_f_0+f_1*t-f*t)-(V_f_0*C_f_a_0+C_f_ai*f_1*t-C_f_a*f*t);
118 fu(37)=C_f_si*(V_f_0+f_1*t-f*t)-C_f_s*f_1*t+C_f_si*f*t;
119 fu(38)=C_f_a*f-J_a*A_m-C_f_ai*f_1;
120 fu(39)=C_f_si*f+J_s*A_m-C_f_s*f_1;
121
122 fu(40)=C_d_s_0*V_d_0+C_d_si*f_2*t-C_d_s*f*t-C_d_s*(V_d_0+f_2*t-f*t);
123 fu(41)=C_d_a*f_2*t-C_d_ai*f*t-C_d_ai*(V_d_0+f_2*t-f*t);
124 fu(42)=C_d_s*f-J_s*A_m-C_d_si*f_2;
125 fu(43)=C_d_ai*f+J_a*A_m-C_d_a*f_2;
126 %additional Equation
127 fu(44)= 0.04*Re_f^(3/4)*Sc_f_s^(1/3)-Sh_f_s;
128 fu(45)= mul_f/den_f/D_f_s-Sc_f_s;
129 fu(46)=J_w*A_m*t*den_d/1000-w;
130 %ionic strenght in feed solution%%%%%%%%%%%%%%%%%%%
131 fu(47)=0.5*(C_f_si+C_f_NH4+C_f_H+C_f_OH+C_f_Ac+C_f_HCO3+2*C_f_CO3)- I_f;
132 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
133 fu(48)=A_I*(I_f^(1/2)/(1+I_f^(1/2))- 0.3*I_f)+log10(gamma_f_H);%Davis
134 fu(49)=A_I*(I_f^(1/2)/(1+I_f^(1/2))- 0.3*I_f)+log10(gamma_f_A);%Davis
135 fu(50)=A_I*(I_f^(1/2)/(1+I_f^(1/2))- 0.3*I_f)+log10(gamma_f_HCO3);%Davis
136 fu(51)=A_I*(I_f^(1/2)/(1+I_f^(1/2))- 0.3*I_f)+log10(gamma_f_OH);%Davis
137 %Charge Balance Equation %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
138 fu(52)=C_f_HCO3+2*C_f_CO3+C_f_Ac+C_f_OH+C_f_si-C_f_NH4-C_f_H;
139 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
140 %New Equation for charge balance
141 fu(53)=K_a_T*C_f_HA/gamma_f_H/gamma_f_Ac/C_f_H-C_f_Ac;
142 fu(54)=K_CO2_T/gamma_f_H/gamma_f_HCO3/C_f_H*C_f_co2-C_f_HCO3;
143 fu(55)=K_HCO3_T*gamma_f_HCO3/gamma_f_H/gamma_f_CO3/C_f_H*C_f_HCO3-C_f_CO3;
144 fu(56)=A_I*4*(I_f^(1/2)/(1+I_f^(1/2))- 0.3*I_f)+log10(gamma_f_CO3);%Davis
145 fu(57)=A_I*(I_f^(1/2)/(1+I_f^(1/2))- 0.3*I_f)+log10(gamma_f_Ac);%Davis
146 fu(58)=K_w_T/C_f_H/gamma_f_OH/gamma_f_H-C_f_OH;
147 fu(59)=C_f_NH4+C_f_NH3-C_f_si;
148 fu(60)=C_f_HA+C_f_Ac-C_f_a;
149 fu(61)=C_f_H*C_f_NH3/K_NH4CL-C_f_NH4;
150 fu(62)=pH_f+log10(gamma_f_H*C_f_H);
151
152
153
154
```

G.3 Valeric acid as feed solution and NH$_4$Cl as draw solution

```matlab
1  clc
2  clear
3
4  [g]=[ 1.608e-03    2.3632e+01 2.755e-01    4.034e-05 5.264e-01    2.324e-04↙
1.285e-11    1.021e-02    6.716e-09...
5  6.647e-09    9.0535e-09 7.339e-01    7.339e-01 5.264e-01    9.999e-01    3.847e-04↙
9.720e-01    9.720e-01...
6  6.171e-04    9.9113e-01 1.880e-01    6.372e+02 3.700e+02    2.894e+04    1.503e+02↙
1.012e+03    4.839e+00...
7  1.581e+00    1.1701e-05 1.725e-01    1.000e-02 6.455e+02    2.759e+04    1.502e+02↙
1.579e+00    4.284e+02...
8  1.843e-01    1.2422e-05 1.000e-02    1.694e-06 5.900e-09    9.994e-01    9.781e-01↙
9.781e-01    3.711e-04...
9  9.781e-01    3.7277e-04 3.176e-12    6.245e+02 3.879e+02    6.835e-05    3.711e-04↙
9.781e-01    9.781e-01...
10 1.298e-08    1.8932e-15 9.151e-01    1.694e-06    3.536e-11    9.6296e-03    9.9624↙
e+02    3.4402e+00];
11
12 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
13 ts=0.1:0.1:1800;    %ts=0.1:0.1:1800;
14 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
15 for ti=1:numel(ts);
16 t=ts(ti);
17 options = optimset('Algorithm','Levenberg-Marquardt','TolFun',1*10^-15,'TolX',1*10^-↙
15 ...
18      ,'Maxiter',25,'Maxfun',15000);
19
20 [r,fval,exitflag,output] = fsolve(@(z)Valeric_NH4Cl(z,t),[g],options)
21
22 j(ti)=r(1,1);
23 c(ti)=r(1,31);
24 d(ti)=r(1,51);
25 p(ti)=r(1,62);
26 T(ti)=t;
27 e(ti)=r(1,52);
28 % For Rejection
29 O(ti)=(1-r(1,10)/r(1,31))*100;
30 K4(ti)=r(1,31)/0.01;
31 K1(ti)=r(1,1);
32 [g]=[r];
33 %for Plotting
34 [A(ti)]=t;
35 [B(ti)]=d(ti);
36 [C(ti)]=p(ti);
37 % For Rejection
38 [M(ti)]=O(ti);
39 [N(ti)]=K1(ti);
40 [K5(ti)]=K4(ti); %Cfa/Cf0
41 disp(ti)
42 fprintf('\n    Time(min)        J_w(L/dm2/min)            pH            Cf,a(mole/l)↙
W(kg)\n');
43
44 fprintf('\n%11.4f%20.4e%19.4f%19.4e%19.4e%\n' ,T(ti),j(ti),p(ti),c(ti),d(ti));
45
46 format short e
47
```

```matlab
48 %Result of Best guess
49 fprintf('\n%8.3e%15.4e%10.3e%14.3e%11.3e%13.3e%13.3e%13.3e%13.3e%\n'...
50     ,r(1,1),r(1,2),r(1,3),r(1,4),r(1,5),r(1,6),r(1,7),r(1,8),r(1,9));
51
52  fprintf('\n%8.3e%15.4e%10.3e%14.3e%11.3e%13.3e%13.3e%13.3e%13.3e%\n'...
53     ,r(1,10),r(1,11),r(1,12),r(1,13),r(1,14),r(1,15),r(1,16),r(1,17),r(1,18));
54
55 fprintf('\n%8.3e%15.4e%10.3e%14.3e%11.3e%13.3e%13.3e%13.3e%13.3e%\n'...
56     ,r(1,19),r(1,20),r(1,21),r(1,22),r(1,23),r(1,24),r(1,25),r(1,26),r(1,27));
57
58 fprintf('\n%8.3e%15.4e%10.3e%14.3e%11.3e%13.3e%13.3e%13.3e%13.3e%\n'...
59     ,r(1,28),r(1,29),r(1,30),r(1,31),r(1,32),r(1,33),r(1,34),r(1,35),r(1,36));
60
61 fprintf('\n%8.3e%15.4e%10.3e%14.3e%11.3e%13.3e%13.3e%13.3e%13.3e%13.3e%13.3e%\n'...
62     ,r(1,37),r(1,38),r(1,39),r(1,40),r(1,41),r(1,42),r(1,43),r(1,44),r(1,45));
63
64 fprintf('\n%8.3e%15.4e%10.3e%14.3e%11.3e%13.3e%13.3e%13.3e%13.3e%\n'...
65     ,r(1,46),r(1,47),r(1,48),r(1,49),r(1,50),r(1,51),r(1,52),r(1,53),r(1,54));
66
67 fprintf('\n%8.3e%15.4e%10.3e%14.3e%13.3e%13.4e%13.4e%13.4e%\n'...
68     ,r(1,55),r(1,56),r(1,57),r(1,58),r(1,59),r(1,60),r(1,61),r(1,62));
69 end
70
71 %Time Interval%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
72 Ai=0:30:1800;
73 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
74 format short
75 Ci=spline(A,C,Ai);
76 format short e
77 Bi=spline(A,B,Ai);
78
79 F=[  0     0.0060    0.0210    0.0350    0.0490    0.0640    0.0780    0.0920↙
0.1070    0.1200    0.1340...
80 0.1490    0.1630    0.1760    0.1890    0.2020    0.2150    0.2280    0.2400↙
0.2530    0.2660    0.2780...
81 0.2900    0.3030    0.3130    0.3260    0.3370    0.3490    0.3600    0.3720↙
0.3830    0.3930    0.4040...
82 0.4140    0.4270    0.4350    0.4460    0.4570    0.4660    0.4760    0.4860↙
0.4960    0.5050    0.5150...
83 0.5240    0.5340    0.5420    0.5510    0.5600    0.5690    0.5780    0.5860↙
0.5940    0.6030    0.6110...
84 0.6200    0.6270    0.6350    0.6430    0.6490    0.6550];
85
86 G=[3.4300    3.4060    3.4168    3.4176    3.4139    3.4000    3.4025    3.3921↙
3.3943    3.3909    3.3803    3.3781    3.3736...
87 3.3688    3.3704    3.3610    3.3552    3.3550    3.3476    3.3486    3.3439↙
3.3431    3.3347    3.3252    3.3288    3.3235...
88 3.3197    3.3163    3.3124    3.3091    3.3004    3.3012    3.2962    3.2894↙
3.2902    3.2854    3.2749    3.2727    3.2706...
89 3.2717    3.2635    3.2579    3.2552    3.2532    3.2422    3.2436    3.2445↙
3.2448    3.2428    3.2396    3.2366    3.2308...
90 3.2232    3.2206    3.2186    3.2168    3.2145    3.2126    3.2116    3.2097↙
3.2082];
91
92
93 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```matlab
94
95 F_m=mean(F);
96 G_m=mean(G);
97
98 sum1=0;sum2=0;
99 sum3=0;sum4=0;
100 %%% Number of Data %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
101 for k=1:1:61
102 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
103 sum1=(Bi(1,k)-F(1,k))^2+sum1;
104 sum2=(Bi(1,k)-F_m)^2+sum2;
105 sum3=(Ci(1,k)-G(1,k))^2+sum3;
106 sum4=(Ci(1,k)-G_m)^2+sum4;
107 end
108
109 format short
110 R2_W=1-sum1/sum2;
111 R2_pH=1-sum3/sum4;
112 MSE_W=sum1/k;
113 MSE_pH=sum3/k;
114 RMSE_W=sqrt(MSE_W)
115 RMSE_pH=sqrt(MSE_pH)
116 SEP_W=RMSE_W/F_m*100
117 SEP_pH=RMSE_pH/G_m*100
118
119 % Plotting
120 subplot(2,3,1)
121 plot([A]/60,[B],'-g',[Ai]/60,[F],'xr')
122 xlabel('Time (hr)')
123 ylabel('Weight (Kg)')
124 title('Weight Change as a Function of Time')
125 legend('Model','Experiment',2)
126
127 subplot(2,3,2)
128 plot([A]/60,[C],'-g',[Ai]/60,[G],'xr')
129 xlabel('Time (hr)')
130 ylabel('pH')
131 title('pH as a Function of Time')
132 legend('Model','Experiment',1)
133
134 subplot(2,3,3)
135 plot([A]/60,[M],'-g')
136 xlabel('Time (hr)')
137 ylabel('%Rejection')
138 title('%Rejection as a Function of Time')
139 legend('Model',1)
140
141 subplot(2,3,4)
142 plot([N],[M],'-g')
143 xlabel('Jw (L/dm2/min)')
144 ylabel('%Rejection')
145 title('%Rejection as a Function of Jw')
146 legend('Model',1)
147
148 subplot(2,3,5)
149 plot([A]/60,[K5],'-g')
```

```
150 xlabel('Time (hr)')
151 ylabel('Cfa/Cf0')
152 title('Cfa/Cf0 as a Function of Time')
153 legend('Model',1)
154
155 subplot(2,3,6)
156 plot([N],[K5],'-g')
157 xlabel('Jw (L/dm2/min)')
158 ylabel('Cfa/Cf0')
159 title('Cfa/Cf0 as a Function of Jw')
160 legend('Model',1)
161
162 fprintf('\n      \n')
163 fprintf('\n      \n')
164 fprintf('\n CONCLUSION     \n')
165 fprintf('\n       R2_W            MSE_W                    R2_pH           MSE_pH\n');
166 fprintf('\n%11.3f%20.4e%19.4f%19.4e%\n',R2_W,MSE_W,R2_pH,MSE_pH)
167 fprintf('\n       \n')
168
169 fprintf('\n     Time(min)       J_w(L/dm2/min)              pH              Cf,a(mole/l) ↵
W(kg)          C_f_H\n');
170 for x=5:5:200000000;
171 z=x*4;
172 fprintf('\n%11.3f%20.4e%19.4f%19.4e%19.4e%19.4e%\n',T(z),j(z),p(z),c(z),d(z),e(z));
173
174 end
175
176
177
178
```

```matlab
1
2  function fu=Valeric_NH4Cl(z,t)
3
4  %Independent Variable
5  V_f_0=1;C_d_s_0=1;C_f_a_0=10*10^(-3);V_d_0=0.5;
6  %Fixed Variable
7  T=301;f=1.58;
8  % T=300..normal f=1.58...normal
9  L=0.9207;D=0.023;W=0.4572;A_m=42*10^(-2);d_h=0.0438;
10 %Constant Variable
11 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
12 B_a=0.492/60/100;% B of Valeric Acid
13 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
14 A=0.413/60/100;B_s=0.46/60/100;
15 S=491*10^(-5);
16 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
17 o_s=0.897;%amonium chloride coefficient
18 D_NH4_298=117.42e-7;D_Cl_298=121.92e-7;%amonium & chloride diffusion coefficient
19 D_A_298=5.226*10^(-6);D_HA_298=4.8022*10^(-6);%Valeric Physicocmical Property
20 D_H_298=5.587*10^(-5);
21 den_w=999.65+2.0438/10*(T-273)-6.174*10^-2*(T-273)^1.5;
22 visco_298=((298-273)+246)/((0.05594*(298-273)+5.2842)*(298-273)+137.37);
23 visco_T=((T-273)+246)/((0.05594*(T-273)+5.2842)*(T-273)+137.37);
24 mul_f=visco_T*6;
25 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
26 n_s=2;o_a=1;R=0.08314;
27 %Temperature Correction
28 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
29 K_a_T=10^-(921.38/T-1.8574+0.012105*T);%Valeric Acid
30 K_CO2_T=10^-(3404.71/T-14.8435+0.032786*T);
31 C_f_co2=1e-5;
32 K_HCO3_T=10^-(2902.39/T-6.4980+0.02379*T);
33 K_w_T=10^(-4470.99/T+6.0875-0.017060*T);
34 K_NH4CL=10^-(2835.76/T-0.6322+0.001225*T);%Amonium Chloride
35 E=2727.586+0.6224107*T-466.9151*log(T)-52000.87/T;
36 d=1-(((T-273)-3.9863)^2*((T-273)+288.9414))/(508929.2*((T-273)+68.12963))...
37 +0.011445*exp((-374.3)/(T-273));
38 A_I=(1.82483*10^6*d^0.5)/(E*T)^1.5;
39 K=1/K_a_T;
40 D_H_T=D_H_298*T/298*visco_298/visco_T ;
41 D_A_T=D_A_298*T/298*visco_298/visco_T;
42 D_HA_T=D_HA_298*T/298*visco_298/visco_T;
43 D_NH4_T=D_NH4_298*T/298*visco_298/visco_T;
44 D_Cl_T=D_Cl_298*T/298*visco_298/visco_T;
45 D_1_T=2/(1/D_H_T +1/D_A_T);
46 D_S_T=2/(1/D_NH4_T +1/D_Cl_T);
47 D_d_s=D_S_T;
48 D_f_s=D_S_T;
49 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
50 %Unknown Variable
51 J_w=z(1);P_i=z(2);P_m=z(3);J_s=z(4);C_i_s=z(5);
52 C_mf_s=z(6);J_a=z(7);C_mf_a=z(8);C_i_a=z(9);C_d_a=z(10);
53 C_d_H=z(11);gamma_d_H=z(12);gamma_d_A=z(13);I_i=z(14);C_d_s=z(15);
54 C_mf_H=z(16);gamma_m_H=z(17);gamma_m_A=z(18);I_m=z(19);C_md_s=z(20);
55 k_d=z(21);Sh_d=z(22);Sc_d=z(23);Re_d=z(24);
56 v_d=z(25);den_d=z(26);mul_d=z(27);f_2=z(28);D_f_a=z(29);
```

```
57  k_f_a=z(30);C_f_a=z(31);Sh_f_a=z(32);Re_f=z(33);v_f=z(34);
58  f_1=z(35);Sc_f_a=z(36);k_f_s=z(37);C_f_s=z(38);
59  C_f_ai=z(39);C_f_si=z(40);C_d_ai=z(41);C_d_si=z(42);gamma_f_Ac=z(43);
60  gamma_f_H=z(44);C_f_Ac=z(45);gamma_f_A=z(46);I_f=z(47);C_f_NH3=z(48);
61  Sh_f_s=z(49);Sc_f_s=z(50);w=z(51);C_f_H=z(52);gamma_f_HCO3=z(53);
62  gamma_f_OH=z(54);C_f_HCO3=z(55);C_f_CO3=z(56);
63  gamma_f_CO3=z(57);C_f_NH4=z(58);C_f_OH=z(59);C_f_HA=z(60);den_f=z(61);
64  pH_f=z(62);
65
66  format short e
67  %Math Model
68  fu(1)=A*(P_i-P_m )-J_w;
69  fu(2)= B_s *(C_i_s-C_mf_s)-J_s;
70  fu(3)= B_a*(C_mf_a-C_i_a)-J_a;
71  fu(4)=C_d_a-C_i_a;
72  fu(5)=J_a/J_w-C_d_a;
73  fu(6)= o_s*n_s*C_i_s*R*T+o_a*(C_d_a+C_d_H)*R*T-P_i;
74  fu(7)=(-(gamma_d_H*K_a_T)+((gamma_d_H*K_a_T)^2+4*K_a_T*gamma_d_H*gamma_d_A*C_d_a)^↙
    (1/2))...
75      /(2*gamma_d_H*gamma_d_A)-C_d_H;
76  fu(8)=A_I*(I_i^(1/2)/(1+I_i^(1/2))- 0.3*I_i)+log10(gamma_d_H);%"Davies"
77  fu(9)=A_I*(I_i^(1/2)/(1+I_i^(1/2))- 0.3*I_i)+log10(gamma_d_A);%"Davies"
78  fu(10)=0.5*(2*C_i_s+2*C_d_H)- I_i;
79  fu(11)=o_s*n_s*C_mf_s*R*T+o_a*(C_mf_a+C_mf_H)*R*T-P_m;
80  fu(12)=(-(gamma_m_H*K_a_T)+((gamma_m_H*K_a_T)^2+4*K_a_T*gamma_m_H*gamma_m_A*C_mf_a)^↙
    (1/2))...
81      /(2*gamma_m_H*gamma_m_A)-C_mf_H;
82  fu(13)=A_I*(I_m^(1/2)/(1+I_m^(1/2))- 0.3*I_m)+log10(gamma_m_H);%"Davies"
83  fu(14)=A_I*(I_m^(1/2)/(1+I_m^(1/2))- 0.3*I_m)+log10(gamma_m_A);%"Davies"
84  fu(15)=0.5*(2*C_mf_s+2*C_mf_H)-I_m;
85  fu(16)=(C_md_s-C_i_s)/2*D_S_T*(1/(J_w*(0.789*C_i_s+0.211*C_md_s)+J_s)...
86  +1/(J_w*(0.211*C_i_s+0.789*C_md_s)+ J_s))-S;
87  fu(17)=(C_d_s-C_md_s)/2*D_S_T*(1/(J_w*(0.789*C_md_s+0.211*C_d_s)+J_s)...
88  +1/(J_w*(0.211*C_md_s+0.789*C_d_s)+ J_s))-D_d_s/k_d;
89  fu(18)=Sh_d*D_d_s/d_h-k_d;
90  fu(19)=L*v_d*den_d/mul_d-Re_d;
91  fu(20)=(f_2+f)/2/W/D-v_d;
92  fu(21)=f+J_w*A_m-f_2;
93  % amonium chloride density and viscosity
94  fu(22)=6*exp((12.396*(C_d_s*53.491/den_d)^1.5039-1.7756)/(0.23471*(T-273)+1)/...
95  (-2.7591*(C_d_s*53.491/den_d)^2.8408+1))-mul_d;
96  fu(23)=den_w+0.2061e2*C_d_s-0.1577*C_d_s*(T-273)+1.553e-3*C_d_s*(T-273)^2-...
97  2.556*C_d_s^1.5+5.67e-2*C_d_s^1.5*(T-273)-5.082e-4*C_d_s^1.5*(T-273)^2-den_d;
98  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
99  fu(24)= 0.04*Re_d^(3/4)*Sc_d^(1/3)-Sh_d;
100 fu(25)= mul_d/den_d/D_d_s-Sc_d;
101 fu(26)=(0.5*(C_f_a-C_mf_a)*((D_1_T-D_HA_T)/...
102     (sqrt(1+4*K*(0.789*C_mf_a+0.211*C_f_a))*(J_w*(0.789*C_mf_a+0.211*C_f_a)-J_a))...
103     +(D_1_T-D_HA_T)/(sqrt(1+4*K*(0.211*C_mf_a+0.789*C_f_a))*(J_w*(0.211*C_mf_a...
104     +0.789*C_f_a)-J_a)+ D_HA_T/(J_w*(0.789*C_mf_a+0.211*C_f_a)-J_a)...
105     +D_HA_T/((J_w*(0.211*C_mf_a+0.789*C_f_a)-J_a))))+D_f_a/k_f_a;
106 fu(27)=Sh_f_a*D_f_a/d_h-k_f_a;
107 fu(28)=(D_1_T/(2*K*C_f_a)*(-1+sqrt(1+4*K*C_f_a))+D_HA_T/...
108     (4*K*C_f_a)*(-1+sqrt(1+4*K*C_f_a))^2)-D_f_a;
109 fu(29)=L*v_f*den_f/mul_f -Re_f;
110 fu(30)=(f+f_1)/W/D/2-v_f;
```

```
111 fu(31)=f-J_w*A_m-f_1;
112 fu(32)= 0.04*Re_f^(3/4)*Sc_f_a^(1/3)-Sh_f_a;
113 fu(33)= mul_f/den_f/D_f_a-Sc_f_a;
114 fu(34)=(C_f_s-C_mf_s)/2*D_S_T*(1/(J_w*(0.789*C_mf_s+0.211*C_f_s)+J_s)...
115 +1/(J_w*(0.211*C_mf_s+0.789*C_f_s)+ J_s))+D_f_s/k_f_s;
116 fu(35)=Sh_f_s*D_f_s/d_h-k_f_s;
117 %Mole balance Equation
118 fu(36)=C_f_a*(V_f_0+f_1*t-f*t)-(V_f_0*C_f_a_0+C_f_ai*f_1*t-C_f_a*f*t);
119 fu(37)=C_f_si*(V_f_0+f_1*t-f*t)-C_f_s*f_1*t+C_f_si*f*t;
120 fu(38)=C_f_a*f-J_a*A_m-C_f_ai*f_1;
121 fu(39)=C_f_si*f+J_s*A_m-C_f_s*f_1;
122
123 fu(40)=C_d_s_0*V_d_0+C_d_si*f_2*t-C_d_s*f*t-C_d_s*(V_d_0+f_2*t-f*t);
124 fu(41)=C_d_a*f_2*t-C_d_ai*f*t-C_d_ai*(V_d_0+f_2*t-f*t);
125 fu(42)=C_d_s*f-J_s*A_m-C_d_si*f_2;
126 fu(43)=C_d_ai*f+J_a*A_m-C_d_a*f_2;
127 %additional Equation
128 fu(44)= 0.04*Re_f^(3/4)*Sc_f_s^(1/3)-Sh_f_s;
129 fu(45)= mul_f/den_f/D_f_s-Sc_f_s;
130 fu(46)=J_w*A_m*t*den_d/1000-w;
131 %ionic strenght in feed solution%%%%%%%%%%%%%%%%%%%
132 fu(47)=0.5*(C_f_si+C_f_NH4+C_f_H+C_f_OH+C_f_Ac+C_f_HCO3+2*C_f_CO3)- I_f;
133 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
134 fu(48)=A_I*(I_f^(1/2)/(1+I_f^(1/2))- 0.3*I_f)+log10(gamma_f_H);%Davis
135 fu(49)=A_I*(I_f^(1/2)/(1+I_f^(1/2))- 0.3*I_f)+log10(gamma_f_A);%Davis
136 fu(50)=A_I*(I_f^(1/2)/(1+I_f^(1/2))- 0.3*I_f)+log10(gamma_f_HCO3);%Davis
137 fu(51)=A_I*(I_f^(1/2)/(1+I_f^(1/2))- 0.3*I_f)+log10(gamma_f_OH);%Davis
138 %Charge Balance Equation %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
139 fu(52)=C_f_HCO3+2*C_f_CO3+C_f_Ac+C_f_OH+C_f_si-C_f_NH4-C_f_H;
140 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
141 %New Equation for charge balance
142 fu(53)=K_a_T*C_f_HA/gamma_f_H/gamma_f_Ac/C_f_H-C_f_Ac;
143 fu(54)=K_CO2_T/gamma_f_H/gamma_f_HCO3/C_f_H*C_f_co2-C_f_HCO3;
144 fu(55)=K_HCO3_T*gamma_f_HCO3/gamma_f_H/gamma_f_CO3/C_f_H*C_f_HCO3-C_f_CO3;
145 fu(56)=A_I*4*(I_f^(1/2)/(1+I_f^(1/2))- 0.3*I_f)+log10(gamma_f_CO3);%Davis
146 fu(57)=A_I*(I_f^(1/2)/(1+I_f^(1/2))- 0.3*I_f)+log10(gamma_f_Ac);%Davis
147 fu(58)=K_w_T/C_f_H/gamma_f_OH/gamma_f_H-C_f_OH;
148 fu(59)=C_f_NH4+C_f_NH3-C_f_si;
149 fu(60)=C_f_HA+C_f_Ac-C_f_a;
150 fu(61)=C_f_H*C_f_NH3/K_NH4CL-C_f_NH4;
151 fu(62)=pH_f+log10(gamma_f_H*C_f_H);
152
153
154
155
```

G.4 Lactic acid as feed solution and NH$_4$Cl as draw solution

```matlab
1 clc
2 clear
3 [g]=[1.607e-03    2.3637e+01 2.943e-01    4.035e-05 5.265e-01    2.342e-04
4.120e-12    1.017e-02    2.153e-09...
4 2.147e-09    2.9253e-09 7.339e-01    7.339e-01 5.265e-01    9.997e-01    1.165e-03
9.589e-01    9.589e-01...
5 1.399e-03    9.9100e-01 1.880e-01    6.372e+02 3.700e+02    2.894e+04    1.503e+02
1.012e+03    4.839e+00...
6 1.581e+00    7.0595e-06 1.231e-01    1.000e-02 7.640e+02    2.759e+04    1.502e+02
1.579e+00    7.101e+02...
7 1.843e-01    1.4121e-05 1.001e-02    3.390e-06 1.870e-09    9.993e-01    9.625e-01
9.625e-01    1.147e-03...
8 9.625e-01    1.1508e-03 2.055e-12    6.245e+02 3.879e+02    1.366e-04    1.147e-03
9.625e-01    9.625e-01...
9 4.336e-09    2.1805e-16 8.581e-01    3.390e-06 1.181e-11    8.8539e-03    9.9624
e+02    2.9569e+00];
10 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
11 ts=0.2:0.05:1800;
12 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
13 %ts=0.2:0.05:1800;
14 for ti=1:numel(ts);
15 t=ts(ti);
16 options = optimset('Algorithm','Levenberg-Marquardt','TolFun',1*10^-15,'TolX',1*10^-
15 ...
17      ,'Maxiter',75,'Maxfun',20000);
18
19 [r,fval,exitflag,output] = fsolve(@(z)Lactic_NH4Cl(z,t),[g],options)
20
21 j(ti)=r(1,1);
22 c(ti)=r(1,31);
23 d(ti)=r(1,51);
24 p(ti)=r(1,62);
25 T(ti)=t;
26 e(ti)=r(1,52);
27 % For Rejection
28 O(ti)=(1-r(1,10)/r(1,31))*100;
29 K4(ti)=r(1,31)/0.01;
30 K1(ti)=r(1,1);
31 [g]=[r];
32 %for Plotting
33 [A(ti)]=t;
34 [B(ti)]=d(ti);
35 [C(ti)]=p(ti);
36 % For Rejection
37 [M(ti)]=O(ti);
38 [N(ti)]=K1(ti);
39 [K5(ti)]=K4(ti); %Cfa/Cf0
40 disp(ti)
41 fprintf('\n    Time(min)      J_w(L/dm2/min)         pH         Cf,a(mole/l)
W(kg)\n');
42
43 fprintf('\n%11.4f%20.4e%19.4f%19.4e%19.4e%\n' ,T(ti),j(ti),p(ti),c(ti),d(ti));
44
45 format short e
46
47 %Result of Best guess
```

```matlab
48 fprintf('\n%8.3e%15.4e%10.3e%14.3e%11.3e%13.3e%13.3e%13.3e\n'...
49     ,r(1,1),r(1,2),r(1,3),r(1,4),r(1,5),r(1,6),r(1,7),r(1,8),r(1,9));
50
51  fprintf('\n%8.3e%15.4e%10.3e%14.3e%11.3e%13.3e%13.3e%13.3e\n'...
52     ,r(1,10),r(1,11),r(1,12),r(1,13),r(1,14),r(1,15),r(1,16),r(1,17),r(1,18));
53
54 fprintf('\n%8.3e%15.4e%10.3e%14.3e%11.3e%13.3e%13.3e%13.3e\n'...
55     ,r(1,19),r(1,20),r(1,21),r(1,22),r(1,23),r(1,24),r(1,25),r(1,26),r(1,27));
56
57 fprintf('\n%8.3e%15.4e%10.3e%14.3e%11.3e%13.3e%13.3e%13.3e\n'...
58     ,r(1,28),r(1,29),r(1,30),r(1,31),r(1,32),r(1,33),r(1,34),r(1,35),r(1,36));
59
60 fprintf('\n%8.3e%15.4e%10.3e%14.3e%11.3e%13.3e%13.3e%13.3e%13.3e%13.3e\n'...
61     ,r(1,37),r(1,38),r(1,39),r(1,40),r(1,41),r(1,42),r(1,43),r(1,44),r(1,45));
62
63 fprintf('\n%8.3e%15.4e%10.3e%14.3e%11.3e%13.3e%13.3e%13.3e\n'...
64     ,r(1,46),r(1,47),r(1,48),r(1,49),r(1,50),r(1,51),r(1,52),r(1,53),r(1,54));
65
66 fprintf('\n%8.3e%15.4e%10.3e%14.3e%13.3e%13.4e%13.4e%13.4e\n'...
67     ,r(1,55),r(1,56),r(1,57),r(1,58),r(1,59),r(1,60),r(1,61),r(1,62));
68 end
69
70 %Time Interval%%%%%%%%%%%%%%%%%%%%%%%%%%
71 Ai=0:30:1800;
72 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
73 format short
74 Ci=spline(A,C,Ai);
75 format short e
76 Bi=spline(A,B,Ai);
77
78 F=[ 0    0.0190    0.0360    0.0540    0.0720    0.0900    0.1060    0.1220...
0.1390    0.1560    0.1710...
79 0.1840    0.1960    0.2090    0.2200    0.2300    0.2400    0.2520    0.2620...
0.2790    0.2880    0.2980...
80 0.3080    0.3170    0.3270    0.3360    0.3440    0.3530    0.3620    0.3700...
0.3780    0.3870    0.3940...
81 0.4010    0.4090    0.4170    0.4240    0.4320    0.4410    0.4470    0.4560...
0.4640    0.4740    0.4800...
82 0.4870    0.4940    0.5010    0.5070    0.5130    0.5200    0.5260    0.5330...
0.5380    0.5440    0.5500...
83 0.5580    0.5660    0.5750    0.5840    0.5920    0.600];
84
85 G=[2.941    2.9360    2.9330    2.9290    2.9260    2.9220    2.9180    2.9160...
2.9120    2.9080    2.9030    2.9010...
86 2.8970    2.8930    2.8900    2.8860    2.8820    2.8790    2.8750    2.8720...
2.8680    2.8640    2.8610...
87 2.8570    2.8540    2.8490    2.8470    2.8420    2.8400    2.8350    2.8310...
2.8260    2.8230    2.8190...
88 2.8140    2.8100    2.8070    2.8020    2.7980    2.7930    2.7890    2.7840...
2.7810    2.7770    2.7720...
89 2.7670    2.7630    2.7580    2.7530    2.7490    2.7440    2.7390    2.7350...
2.7300    2.7250    2.7200...
90 2.7140    2.7090    2.7040    2.6990    2.6930];
91 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
92 F_m=mean(F);
93 G_m=mean(G);
```

```matlab
94
95 sum1=0;sum2=0;
96 sum3=0;sum4=0;
97 %%% Number of Data %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
98 for k=1:1:61
99 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
100 sum1=(Bi(1,k)-F(1,k))^2+sum1;
101 sum2=(Bi(1,k)-F_m)^2+sum2;
102 sum3=(Ci(1,k)-G(1,k))^2+sum3;
103 sum4=(Ci(1,k)-G_m)^2+sum4;
104 end
105
106 format short
107 R2_W=1-sum1/sum2;
108 R2_pH=1-sum3/sum4;
109 MSE_W=sum1/k;
110 MSE_pH=sum3/k;
111 RMSE_W=sqrt(MSE_W)
112 RMSE_pH=sqrt(MSE_pH)
113 SEP_W=RMSE_W/F_m*100
114 SEP_pH=RMSE_pH/G_m*100
115
116 % Plotting
117 subplot(2,3,1)
118 plot([A]/60,[B],'-g',[Ai]/60,[F],'xr')
119 xlabel('Time (hr)')
120 ylabel('Weight (Kg)')
121 title('Weight Change as a Function of Time')
122 legend('Model','Experiment',2)
123
124 subplot(2,3,2)
125 plot([A]/60,[C],'-g',[Ai]/60,[G],'xr')
126 xlabel('Time (hr)')
127 ylabel('pH')
128 title('pH as a Function of Time')
129 legend('Model','Experiment',1)
130
131 subplot(2,3,3)
132 plot([A]/60,[M],'-g')
133 xlabel('Time (hr)')
134 ylabel('%Rejection')
135 title('%Rejection as a Function of Time')
136 legend('Model',1)
137
138 subplot(2,3,4)
139 plot([N],[M],'-g')
140 xlabel('Jw (L/dm2/min)')
141 ylabel('%Rejection')
142 title('%Rejection as a Function of Jw')
143 legend('Model',1)
144
145 subplot(2,3,5)
146 plot([A]/60,[K5],'-g')
147 xlabel('Time (hr)')
148 ylabel('Cfa/Cf0')
149 title('Cfa/Cf0 as a Function of Time')
```

```
150 legend('Model',1)
151
152 subplot(2,3,6)
153 plot([N],[K5],'-g')
154 xlabel('Jw (L/dm2/min)')
155 ylabel('Cfa/Cf0')
156 title('Cfa/Cf0 as a Function of Jw')
157 legend('Model',1)
158
159 fprintf('\n      \n')
160 fprintf('\n     \n')
161 fprintf('\n CONCLUSION    \n')
162 fprintf('\n      R2_W           MSE_W                  R2_pH          MSE_pH\n');
163 fprintf('\n%11.3f%20.4e%19.4f%19.4e%\n',R2_W,MSE_W,R2_pH,MSE_pH)
164 fprintf('\n     \n')
165
166 fprintf('\n    Time(min)       J_w(L/dm2/min)              pH              Cf,a(mole/l) ↵
W(kg)        C_f_H\n');
167 for x=5:5:200000000;
168 z=x*4;
169 fprintf('\n%11.3f%20.4e%19.4f%19.4e%19.4e%19.4e%\n',T(z),j(z),p(z),c(z),d(z),e(z));
170
171 end
172
173
174
```

```matlab
1
2 function fu=Lactic_NH4Cl(z,t)
3 %Independent Variable
4 V_f_0=1;C_d_s_0=1;C_f_a_0=10*10^(-3);V_d_0=0.5;
5 %Fixed Variable
6 T=301;f=1.58;
7 % T=300..normal f=1.58...normal
8 L=0.9207;D=0.023;W=0.4572;A_m=42*10^(-2);d_h=0.0438;
9 %Constant Variable
10 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
11 B_a=0.153/60/100; %B of Lactic Acid or(0.153 L/m2/h)
12 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
13 A=0.413/60/100;B_s=0.46/60/100;
14 S=491*10^(-5);
15 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
16 o_s=0.897;%amonium chloride coefficient
17 D_NH4_298=117.42e-7;D_Cl_298=121.92e-7;%amonium & chloride diffusion coefficient
18 D_H_298=5.587*10^(-5);
19 D_A_298=6.198*10^(-6);D_HA_298=4.5834*10^(-6);%Lactic Physicocmical Property
20 den_w=999.65+2.0438/10*(T-273)-6.174*10^-2*(T-273)^1.5;
21 visco_298=((298-273)+246)/((0.05594*(298-273)+5.2842)*(298-273)+137.37);
22 visco_T=((T-273)+246)/((0.05594*(T-273)+5.2842)*(T-273)+137.37);
23 mul_f=visco_T*6;
24 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
25 n_s=2;o_a=1;R=0.08314;
26 %Temperature Correction
27 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
28 K_a_T=10^-(1286.49/T-4.8607+0.014776*T);%Lactic Acid
29 K_CO2_T=10^-(3404.71/T-14.8435+0.032786*T);
30 C_f_co2=1e-5;
31 K_HCO3_T=10^-(2902.39/T-6.4980+0.02379*T);
32 K_w_T=10^(-4470.99/T+6.0875-0.017060*T);
33 K_NH4CL=10^-(2835.76/T-0.6322+0.001225*T);%Amonium Chloride
34 E=2727.586+0.6224107*T-466.9151*log(T)-52000.87/T;
35 d=1-(((T-273)-3.9863)^2*((T-273)+288.9414))/(508929.2*((T-273)+68.12963))...
36 +0.011445*exp((-374.3)/(T-273));
37 A_I=(1.82483*10^6*d^0.5)/(E*T)^1.5;
38 K=1/K_a_T;
39 D_H_T=D_H_298*T/298*visco_298/visco_T ;
40 D_A_T=D_A_298*T/298*visco_298/visco_T;
41 D_HA_T=D_HA_298*T/298*visco_298/visco_T;
42 D_NH4_T=D_NH4_298*T/298*visco_298/visco_T;
43 D_Cl_T=D_Cl_298*T/298*visco_298/visco_T;
44 D_1_T=2/(1/D_H_T +1/D_A_T);
45 D_S_T=2/(1/D_NH4_T +1/D_Cl_T);
46 D_d_s=D_S_T;
47 D_f_s=D_S_T;
48 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
49 %Unknown Variable
50 J_w=z(1);P_i=z(2);P_m=z(3);J_s=z(4);C_i_s=z(5);
51 C_mf_s=z(6);J_a=z(7);C_mf_a=z(8);C_i_a=z(9);C_d_a=z(10);
52 C_d_H=z(11);gamma_d_H=z(12);gamma_d_A=z(13);I_i=z(14);C_d_s=z(15);
53 C_mf_H=z(16);gamma_m_H=z(17);gamma_m_A=z(18);I_m=z(19);C_md_s=z(20);
54 k_d=z(21);Sh_d=z(22);Sc_d=z(23);Re_d=z(24);
55 v_d=z(25);den_d=z(26);mul_d=z(27);f_2=z(28);D_f_a=z(29);
56 k_f_a=z(30);C_f_a=z(31);Sh_f_a=z(32);Re_f=z(33);v_f=z(34);
```

```matlab
57 f_1=z(35);Sc_f_a=z(36);k_f_s=z(37);C_f_s=z(38);
58 C_f_ai=z(39);C_f_si=z(40);C_d_ai=z(41);C_d_si=z(42);gamma_f_Ac=z(43);
59 gamma_f_H=z(44);C_f_Ac=z(45);gamma_f_A=z(46);I_f=z(47);C_f_NH3=z(48);
60 Sh_f_s=z(49);Sc_f_s=z(50);w=z(51);C_f_H=z(52);gamma_f_HCO3=z(53);
61 gamma_f_OH=z(54);C_f_HCO3=z(55);C_f_CO3=z(56);
62 gamma_f_CO3=z(57);C_f_NH4=z(58);C_f_OH=z(59);C_f_HA=z(60);den_f=z(61);
63 pH_f=z(62);
64
65 format short e
66 %Math Model
67 fu(1)=A*(P_i-P_m )-J_w;
68 fu(2)= B_s *(C_i_s-C_mf_s)-J_s;
69 fu(3)= B_a*(C_mf_a-C_i_a)-J_a;
70 fu(4)=C_d_a-C_i_a;
71 fu(5)=J_a/J_w-C_d_a;
72 fu(6)= o_s*n_s*C_i_s*R*T+o_a*(C_d_a+C_d_H)*R*T-P_i;
73 fu(7)=(-(gamma_d_H*K_a_T)+((gamma_d_H*K_a_T)^2+4*K_a_T*gamma_d_H*gamma_d_A*C_d_a)^↙
(1/2))...
74     /(2*gamma_d_H*gamma_d_A)-C_d_H;
75 fu(8)=A_I*(I_i^(1/2)/(1+I_i^(1/2))- 0.3*I_i)+log10(gamma_d_H);%"Davies"
76 fu(9)=A_I*(I_i^(1/2)/(1+I_i^(1/2))- 0.3*I_i)+log10(gamma_d_A);%"Davies"
77 fu(10)=0.5*(2*C_i_s+2*C_d_H)- I_i;
78 fu(11)=o_s*n_s*C_mf_s*R*T+o_a*(C_mf_a+C_mf_H)*R*T-P_m;
79 fu(12)=(-(gamma_m_H*K_a_T)+((gamma_m_H*K_a_T)^2+4*K_a_T*gamma_m_H*gamma_m_A*C_mf_a)^↙
(1/2))...
80     /(2*gamma_m_H*gamma_m_A)-C_mf_H;
81 fu(13)=A_I*(I_m^(1/2)/(1+I_m^(1/2))- 0.3*I_m)+log10(gamma_m_H);%"Davies"
82 fu(14)=A_I*(I_m^(1/2)/(1+I_m^(1/2))- 0.3*I_m)+log10(gamma_m_A);%"Davies"
83 fu(15)=0.5*(2*C_mf_s+2*C_mf_H)-I_m;
84 fu(16)=(C_md_s-C_i_s)/2*D_S_T*(1/(J_w*(0.789*C_i_s+0.211*C_md_s)+J_s)...
85 +1/(J_w*(0.211*C_i_s+0.789*C_md_s)+ J_s))-S;
86 fu(17)=(C_d_s-C_md_s)/2*D_S_T*(1/(J_w*(0.789*C_md_s+0.211*C_d_s)+J_s)...
87 +1/(J_w*(0.211*C_md_s+0.789*C_d_s)+ J_s))-D_d_s/k_d;
88 fu(18)=Sh_d*D_d_s/d_h-k_d;
89 fu(19)=L*v_d*den_d/mul_d-Re_d;
90 fu(20)=(f_2+f)/2/W/D-v_d;
91 fu(21)=f+J_w*A_m-f_2;
92 % amonium chloride density and viscosity
93 fu(22)=6*exp((12.396*(C_d_s*53.491/den_d)^1.5039-1.7756)/(0.23471*(T-273)+1)/...
94 (-2.7591*(C_d_s*53.491/den_d)^2.8408+1))-mul_d;
95 fu(23)=den_w+0.2061e2*C_d_s-0.1577*C_d_s*(T-273)+1.553e-3*C_d_s*(T-273)^2-...
96 2.556*C_d_s^1.5+5.67e-2*C_d_s^1.5*(T-273)-5.082e-4*C_d_s^1.5*(T-273)^2-den_d;
97 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
98 fu(24)= 0.04*Re_d^(3/4)*Sc_d^(1/3)-Sh_d;
99 fu(25)= mul_d/den_d/D_d_s-Sc_d;
100 fu(26)=(0.5*(C_f_a-C_mf_a)*((D_1_T-D_HA_T)/...
101     (sqrt(1+4*K*(0.789*C_mf_a+0.211*C_f_a))*(J_w*(0.789*C_mf_a+0.211*C_f_a)-J_a))...
102     +(D_1_T-D_HA_T)/(sqrt(1+4*K*(0.211*C_mf_a+0.789*C_f_a))*(J_w*(0.211*C_mf_a...
103     +0.789*C_f_a)-J_a))+ D_HA_T/(J_w*(0.789*C_mf_a+0.211*C_f_a)-J_a)...
104     +D_HA_T/((J_w*(0.211*C_mf_a+0.789*C_f_a)-J_a))))+D_f_a/k_f_a;
105 fu(27)=Sh_f_a*D_f_a/d_h-k_f_a;
106 fu(28)=(D_1_T/(2*K*C_f_a)*(-1+sqrt(1+4*K*C_f_a))+D_HA_T/...
107     (4*K*C_f_a)*(-1+sqrt(1+4*K*C_f_a))^2)-D_f_a;
108 fu(29)=L*v_f*den_f/mul_f -Re_f;
109 fu(30)=(f+f_1)/W/D/2-v_f;
110 fu(31)=f-J_w*A_m-f_1;
```

```matlab
111 fu(32)= 0.04*Re_f^(3/4)*Sc_f_a^(1/3)-Sh_f_a;
112 fu(33)= mul_f/den_f/D_f_a-Sc_f_a;
113 fu(34)=(C_f_s-C_mf_s)/2*D_S_T*(1/(J_w*(0.789*C_mf_s+0.211*C_f_s)+J_s)...
114 +1/(J_w*(0.211*C_mf_s+0.789*C_f_s)+ J_s))+D_f_s/k_f_s;
115 fu(35)=Sh_f_s*D_f_s/d_h-k_f_s;
116 %Mole balance Equation
117 fu(36)=C_f_a*(V_f_0+f_1*t-f*t)-(V_f_0*C_f_a_0+C_f_ai*f_1*t-C_f_a*f*t);
118 fu(37)=C_f_si*(V_f_0+f_1*t-f*t)-C_f_s*f_1*t+C_f_si*f*t;
119 fu(38)=C_f_a*f-J_a*A_m-C_f_ai*f_1;
120 fu(39)=C_f_si*f+J_s*A_m-C_f_s*f_1;
121
122 fu(40)=C_d_s_0*V_d_0+C_d_si*f_2*t-C_d_s*f*t-C_d_s*(V_d_0+f_2*t-f*t);
123 fu(41)=C_d_a*f_2*t-C_d_ai*f*t-C_d_ai*(V_d_0+f_2*t-f*t);
124 fu(42)=C_d_s*f-J_s*A_m-C_d_si*f_2;
125 fu(43)=C_d_ai*f+J_a*A_m-C_d_a*f_2;
126 %additional Equation
127 fu(44)= 0.04*Re_f^(3/4)*Sc_f_s^(1/3)-Sh_f_s;
128 fu(45)= mul_f/den_f/D_f_s-Sc_f_s;
129 fu(46)=J_w*A_m*t*den_d/1000-w;
130 %ionic strenght in feed solution%%%%%%%%%%%%%%%%%%%
131 fu(47)=0.5*(C_f_si+C_f_NH4+C_f_H+C_f_OH+C_f_Ac+C_f_HCO3+2*C_f_CO3)- I_f;
132 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
133 fu(48)=A_I*(I_f^(1/2)/(1+I_f^(1/2))- 0.3*I_f)+log10(gamma_f_H);%Davis
134 fu(49)=A_I*(I_f^(1/2)/(1+I_f^(1/2))- 0.3*I_f)+log10(gamma_f_A);%Davis
135 fu(50)=A_I*(I_f^(1/2)/(1+I_f^(1/2))- 0.3*I_f)+log10(gamma_f_HCO3);%Davis
136 fu(51)=A_I*(I_f^(1/2)/(1+I_f^(1/2))- 0.3*I_f)+log10(gamma_f_OH);%Davis
137 %Charge Balance Equation %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
138 fu(52)=C_f_HCO3+2*C_f_CO3+C_f_Ac+C_f_OH+C_f_si-C_f_NH4-C_f_H;
139 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
140 %New Equation for charge balance
141 fu(53)=K_a_T*C_f_HA/gamma_f_H/gamma_f_Ac/C_f_H-C_f_Ac;
142 fu(54)=K_CO2_T/gamma_f_H/gamma_f_HCO3/C_f_H*C_f_co2-C_f_HCO3;
143 fu(55)=K_HCO3_T*gamma_f_HCO3/gamma_f_H/gamma_f_CO3/C_f_H*C_f_HCO3-C_f_CO3;
144 fu(56)=A_I*4*(I_f^(1/2)/(1+I_f^(1/2))- 0.3*I_f)+log10(gamma_f_CO3);%Davis
145 fu(57)=A_I*(I_f^(1/2)/(1+I_f^(1/2))- 0.3*I_f)+log10(gamma_f_Ac);%Davis
146 fu(58)=K_w_T/C_f_H/gamma_f_OH/gamma_f_H-C_f_OH;
147 fu(59)=C_f_NH4+C_f_NH3-C_f_si;
148 fu(60)=C_f_HA+C_f_Ac-C_f_a;
149 fu(61)=C_f_H*C_f_NH3/K_NH4CL-C_f_NH4;
150 fu(62)=pH_f+log10(gamma_f_H*C_f_H);
151
152
153
154
```

G.5 A mixture of 10 mM lactic acid and 10 mM butyric acid as feed solution and NH$_4$Cl as draw solution

```
 1 clc
 2 clear
 3 [g]=[1.594e-03    2.3745e+01    5.822e-01    4.053e-05  5.289e-01    2.376e-04✔
4.326e-12   1.021e-02    2.304e-09...
 4 2.298e-09    1.9193e-05    7.340e-01    5.289e-01  9.995e-01    1.248e-03    9.577✔
e-01    1.486e-03    9.909e-01...
 5 1.880e-01    6.3719e+02    3.700e+02    2.894e+04  1.503e+02    1.012e+03    4.839✔
e+00    1.581e+00    1.168e-05...
 6 1.723e-01    1.0002e-02    6.459e+02    2.759e+04  1.502e+02    1.579e+00    4.291✔
e+02    1.843e-01    1.657e-05...
 7 1.001e-02    5.7890e-06    1.424e-09    9.991e-01  9.612e-01    1.080e-03    1.237✔
e-03    6.245e+02    3.879e+02...
 8 2.305e-04    1.2317e-03    5.830e-09    1.660e-15  8.535e-01    5.789e-06    1.103✔
e-11    9.962e+02    2.927e+00...
 9 2.137e-09    1.2131e-09    5.289e-01    1.096e-03  1.153e-11    2.376e-04    4.994✔
e+00    1.013e-02    3.904e-08...
10 6.932e-11    3.7792e-08    2.376e-08    1.526e-04  1.000e-02    1.169e-05    1.724✔
e-01    6.457e+02    4.287e+02...
11 1.001e-02    2.3431e-08    1.515e-04];
12 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
13 ts=0.31:0.000025:1200;
14 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
15 for ti=1:numel(ts);
16 t=ts(ti);
17 options = optimset('Algorithm','Levenberg-Marquardt','TolFun',1*10^-18,'TolX',1*10^-✔
18 ...
18      ,'Maxiter',30,'Maxfun',15000);
19 [r,fval,exitflag,output] = fsolve(@(z)Lactic10_Butyric10(z,t),[g],options)
20 j(ti)=r(1,1);%Jw
21 c(ti)=r(1,29);%Cfa
22 d(ti)=r(1,46);%W
23 p(ti)=r(1,54);%pH
24 T(ti)=t;
25 e(ti)=r(1,68);%Cfa1
26 % For Rejection
27 O(ti)=(1-r(1,10)/r(1,31))*100;%Rej
28 O1(ti)=(1-r(1,65)/r(1,68))*100;%Rej1
29 %For Concentration Performance
30 K4(ti)=r(1,29)/0.01;%Cfa/Cf0
31 K6(ti)=r(1,68)/0.01;%Cfa1/Cf0
32 K1(ti)=r(1,1);
33 [g]=[r];
34 %for Plotting
35 [A(ti)]=t;
36 [B(ti)]=d(ti);
37 [C(ti)]=p(ti);
38 % For Rejection
39 [M(ti)]=O(ti);
40 [N(ti)]=K1(ti);
41 [K5(ti)]=K4(ti); %Cfa/Cf0
42 disp(ti)
43 fprintf('\n     Time(min)     J_w(L/dm2/min)          pH          Cf,a(mole/l)✔
W(kg)          Cf,a1(mole/l)\n');
44
45 fprintf('\n%11.4f%20.4e%19.4f%19.4e%19.4e%19.4e%\n' ,T(ti),j(ti),p(ti),c(ti),d(ti),e✔
(ti));
```

```
46 format short e
47
48 %Result of Best guess
49 fprintf('\n%8.3e%15.4e%13.3e%14.3e%11.3e%13.3e%13.3e%13.3e%13.3e\n'...
50     ,r(1,1),r(1,2),r(1,3),r(1,4),r(1,5),r(1,6),r(1,7),r(1,8),r(1,9));
51
52  fprintf('\n%8.3e%15.4e%13.3e%14.3e%11.3e%13.3e%13.3e%13.3e%13.3e\n'...
53     ,r(1,10),r(1,11),r(1,12),r(1,13),r(1,14),r(1,15),r(1,16),r(1,17),r(1,18));
54
55 fprintf('\n%8.3e%15.4e%13.3e%14.3e%11.3e%13.3e%13.3e%13.3e%13.3e\n'...
56     ,r(1,19),r(1,20),r(1,21),r(1,22),r(1,23),r(1,24),r(1,25),r(1,26),r(1,27));
57
58 fprintf('\n%8.3e%15.4e%13.3e%14.3e%11.3e%13.3e%13.3e%13.3e%13.3e\n'...
59    ,r(1,28),r(1,29),r(1,30),r(1,31),r(1,32),r(1,33),r(1,34),r(1,35),r(1,36));
60
61 fprintf('\n%8.3e%15.4e%13.3e%14.3e%11.3e%13.3e%13.3e%13.3e%13.3e%13.3e%13.3e\n'...
62     ,r(1,37),r(1,38),r(1,39),r(1,40),r(1,41),r(1,42),r(1,43),r(1,44),r(1,45));
63
64 fprintf('\n%8.3e%15.4e%13.3e%14.3e%11.3e%13.3e%13.3e%13.3e%13.3e\n'...
65     ,r(1,46),r(1,47),r(1,48),r(1,49),r(1,50),r(1,51),r(1,52),r(1,53),r(1,54));
66
67 fprintf('\n%8.3e%15.4e%13.3e%14.3e%11.3e%13.3e%13.3e%13.3e%13.3e\n'...
68     ,r(1,55),r(1,56),r(1,57),r(1,58),r(1,59),r(1,60),r(1,61),r(1,62),r(1,63));
69
70 fprintf('\n%8.3e%15.4e%13.3e%14.3e%11.3e%13.3e%13.3e%13.3e%13.3e\n'...
71     ,r(1,64),r(1,65),r(1,66),r(1,67),r(1,68),r(1,69),r(1,70),r(1,71),r(1,72));
72
73 fprintf('\n%8.3e%15.4e%13.3e%\n'...
74     ,r(1,73),r(1,74),r(1,75));
75
76 end
77 %Time Interval%%%%%%%%%%%%%%%%%%%%%%%%%
78 Ai=0:30:1200;
79 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
80 format short
81 Ci=spline(A,C,Ai);
82 format short e
83 Bi=spline(A,B,Ai);
84 %Experiment Value W and pH%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
85  F=[    0    0.0160    0.0310    0.0430    0.0580    0.0720    0.0860    0.1000↙
0.1140    0.1270    0.1400    0.1530    0.1650...
86 0.1780    0.1900    0.2010    0.2130    0.2250    0.2360    0.2480    0.2600↙
0.2710    0.2830    0.2940    0.3050    0.3160...
87 0.3260    0.3370    0.3480    0.3590    0.3690    0.3790    0.3900    0.4000↙
0.4100    0.4200    0.4300    0.4400    0.4500...
88 0.4600    0.4690];
89
90 G=[    2.9210    2.9140    2.9100    2.9060    2.9000    2.8960    2.8920    2.8890↙
2.8870    2.8840    2.8800    2.8760    2.8730...
91 2.8700    2.8660    2.8630    2.8600    2.8550    2.8520    2.8500    2.8470↙
2.8440    2.8400    2.8360    2.8330    2.8290...
92 2.8260    2.8230    2.8190    2.8150    2.8110    2.8080    2.8040    2.8010↙
2.7980    2.7950    2.7930    2.7890    2.7880...
93 2.7850    2.7820];
94
95 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```matlab
 96
 97 F_m=mean(F);
 98 G_m=mean(G);
 99 sum1=0;sum2=0;
100 sum3=0;sum4=0;
101
102 %%% Number of Data %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
103 for k=1:1:41
104 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
105 sum1=(Bi(1,k)-F(1,k))^2+sum1;
106 sum2=(Bi(1,k)-F_m)^2+sum2;
107 sum3=(Ci(1,k)-G(1,k))^2+sum3;
108 sum4=(Ci(1,k)-G_m)^2+sum4;
109 end
110
111 format short
112 R2_W=1-sum1/sum2;
113 R2_pH=1-sum3/sum4;
114 MSE_W=sum1/k;
115 MSE_pH=sum3/k;
116 RMSE_W=sqrt(MSE_W)
117 RMSE_pH=sqrt(MSE_pH)
118 SEP_W=RMSE_W/F_m*100
119 SEP_pH=RMSE_pH/G_m*100
120
121 % Plotting
122 subplot(2,3,1)
123 plot([A]/60,[B],'-g',[Ai]/60,[F],'xr')
124 xlabel('Time (hr)')
125 ylabel('Weight (Kg)')
126 title('Weight Change as a Function of Time')
127 legend('Model','Experiment',2)
128
129 subplot(2,3,2)
130 plot([A]/60,[C],'-g',[Ai]/60,[G],'xr')
131 xlabel('Time (hr)')
132 ylabel('pH')
133 title('pH as a Function of Time')
134 legend('Model','Experiment',1)
135
136 subplot(2,3,3)
137 plot([A]/60,[M],'-g')
138 xlabel('Time (hr)')
139 ylabel('%Rejection1')
140 title('%Rejection1 as a Function of Time')
141 legend('Model',1)
142
143 subplot(2,3,4)
144 plot([A]/60,[O1],'-g')
145 xlabel('Time (hr)')
146 ylabel('%Rejection2')
147 title('%Rejection2 as a Function of Time')
148 legend('Model',1)
149
150 subplot(2,3,5)
151 plot([A]/60,[K5],'-g')
```

```
152 xlabel('Time (hr)')
153 ylabel('Cfa/Cf0')
154 title('Cfa/Cf0 as a Function of Time')
155 legend('Model',1)
156
157 subplot(2,3,6)
158 plot([A]/60,[K6],'-g')
159 xlabel('Time (hr)')
160 ylabel('Cfa1/Cf0')
161 title('Cfa/Cf01 as a Function of Time')
162 legend('Model',1)
163
164 fprintf('\n      \n')
165 fprintf('\n      \n')
166 fprintf('\n CONCLUSION    \n')
167 fprintf('\n      R2_W              MSE_W                   R2_pH          MSE_pH\n');
168 fprintf('\n%11.3f%20.4e%19.4f%19.4e%\n',R2_W,MSE_W,R2_pH,MSE_pH)
169 fprintf('\n      \n')
170
171 fprintf('\n     Time(min)      J_w(L/dm2/min)           pH             Cf,a(mole/l)↵
W(kg)        Cf,a1(mole/l)\n');
172 for x=1:1:200000000;
173 z=x*40000;
174 fprintf('\n%11.3f%20.4e%19.4f%19.4e%19.4e%19.4e%\n',T(z),j(z),p(z),c(z),d(z),e(z));
175 end
176
177
178
```

```matlab
1
2  function fu=Lactic10_Butyric10(z,t)
3  %Independent Variables
4  V_f_0=1;C_d_s_0=1;V_d_0=0.5;
5  %Fixed Variable
6  T=301;f=1.58;
7  L=0.9207;D=0.023;W=0.4572;A_m=42*10^(-2);d_h=0.0438;
8  % Acid Variables
9  %
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
10 C_f_a_0=10*10^(-3);
11 C_f_a1_0=10*10^(-3);
12 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
13 B_a=0.153/60/100;
14 B_a1=0.639/60/100;
15 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
16 D_A_298=6.198*10^(-6);D_HA_298=4.5834*10^(-6);%Lactic Physicocmical Property
17 D_A1_298=5.208*10^(-6);D_HA1_298=5.429*10^(-6);%Butyric Physicocmical Property
18 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
19 K_a_T=10^-(1286.49/T-4.8607+0.014776*T);%Lactic Acid
20 K_a1_T=10^-(1033.39/T-2.6215+0.013334*T);%Butyric Acid
21 %
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
^^^^
22 %Constant Variable
23 B_s=0.460/60/100;
24 A=0.413/60/100;
25 o_s=0.897;%amonium chloride coefficient
26 S=491*10^-5;
27 D_NH4_298=117.42e-7;D_Cl_298=121.92e-7;%amonium & chloride diffusion coefficient
28 den_w=999.65+2.0438/10*(T-273)-6.174*10^-2*(T-273)^1.5;
29 D_H_298=5.587*10^(-5);
30 visco_298=((298-273)+246)/((0.05594*(298-273)+5.2842)*(298-273)+137.37);
31 visco_T=((T-273)+246)/((0.05594*(T-273)+5.2842)*(T-273)+137.37);
32 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
33 o_a=1;n_s=2;R=0.08314;o_a1=1;
34 %Temperature Correction
35 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
36 K_CO2_T=10^-(3404.71/T-14.8435+0.032786*T);
37 K_HCO3_T=10^-(2902.39/T-6.4980+0.02379*T);
38 K_w_T=10^(-4470.99/T+6.0875-0.017060*T);
39 K_NH4CL=10^-(2835.76/T-0.6322+0.001225*T);%Amonium Chloride
40 E=2727.586+0.6224107*T-466.9151*log(T)-52000.87/T;
41 d=1-(((T-273)-3.9863)^2*((T-273)+288.9414))/(508929.2*((T-273)+68.12963))...
42 +0.011445*exp((-374.3)/(T-273));
43 A_I=(1.82483*10^6*d^0.5)/(E*T)^1.5;
44 K=1/K_a_T;
45 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
46 K1=1/K_a1_T;
47 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
48 D_H_T=D_H_298*T/298*visco_298/visco_T ;
49 D_A_T=D_A_298*T/298*visco_298/visco_T;
50 D_HA_T=D_HA_298*T/298*visco_298/visco_T;
51 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
52 D_A1_T=D_A1_298*T/298*visco_298/visco_T;
53 D_HA1_T=D_HA1_298*T/298*visco_298/visco_T;
```

```matlab
54 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
55 D_NH4_T=D_NH4_298*T/298*visco_298/visco_T;
56 D_Cl_T=D_Cl_298*T/298*visco_298/visco_T;
57 D_1_T=2/(1/D_H_T +1/D_A_T);
58 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
59 D_11_T=2/(1/D_H_T +1/D_A1_T);
60 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
61 D_S_T=2/(1/D_NH4_T +1/D_Cl_T);
62 D_d_s=D_S_T;
63 D_f_s=D_S_T;
64 K_H=0.034*exp(-2400*(1/T-1/298));%*****************
65 C_f_co2=10^-3.408*K_H;%*************************
66 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
67 %Unknown Variable
68 J_w=z(1);P_i=z(2);P_m=z(3);J_s=z(4);C_i_s=z(5);
69 C_mf_s=z(6);J_a=z(7);C_mf_a=z(8);C_i_a=z(9);C_d_a=z(10);
70 C_d_H=z(11);gamma_i_H=z(12);I_i=z(13);C_d_s=z(14);
71 C_mf_H=z(15);gamma_mf_H=z(16);I_mf=z(17);C_md_s=z(18);
72 k_d=z(19);Sh_d=z(20);Sc_d=z(21);Re_d=z(22);
73 v_d=z(23);den_d=z(24);mul_d=z(25);f_2=z(26);D_f_a=z(27);
74 k_f_a=z(28);C_f_a=z(29);Sh_f_a=z(30);Re_f=z(31);v_f=z(32);
75 f_1=z(33);Sc_f_a=z(34);k_f_s=z(35);C_f_s=z(36);
76 C_f_ai=z(37);C_f_si=z(38);C_d_ai=z(39);C_d_si=z(40);
77 gamma_f_H=z(41);C_f_A=z(42);I_f=z(43);Sh_f_s=z(44);Sc_f_s=z(45);
78 w=z(46);C_f_H=z(47);C_f_HCO3=z(48);C_f_CO3=z(49);gamma_f_CO3=z(50);
79 C_f_NH4=z(51);C_f_OH=z(52);den_f=z(53);pH_f=z(54);
80 % for charge balance of NH4
81 C_d_A=z(55);C_d_OH=z(56);C_i_NH4=z(57);C_mf_A=z(58);C_mf_OH=z(59);
82 C_mf_NH4=z(60);mul_f=z(61);
83 % for two acid
84 C_mf_a1=z(62);C_i_a1=z(63);J_a1=z(64);C_d_a1=z(65);C_d_A1=z(66);
85 C_mf_A1=z(67);C_f_a1=z(68);D_f_a1=z(69);k_f_a1=z(70);Sh_f_a1=z(71);
86 Sc_f_a1=z(72);C_f_ai1=z(73);C_d_ai1=z(74);C_f_A1=z(75);
87 format short e
88 %Math Model
89 fu(1)=A*(P_i-P_m )-J_w;
90 fu(2)= B_s *(C_i_s-C_mf_s)-J_s;
91 fu(3)= B_a*(C_mf_a-C_i_a)-J_a;
92 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
93 fu(4)= B_a1*(C_mf_a1-C_i_a1)-J_a1;
94 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
95 fu(5)=C_d_a-C_i_a;
96 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
97 fu(6)=C_d_a1-C_i_a1;
98 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
99 fu(7)=J_a/J_w-C_d_a;
100 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
101 fu(8)=J_a1/J_w-C_d_a1;
102 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
103 fu(9)= o_s*n_s*C_i_s*R*T+o_a*(C_d_a+C_d_H)*R*T+o_a1*(C_d_a1+C_d_H)*R*T-P_i;
104 %Charge Balance Equation at interface %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
105 fu(10)=C_d_A+C_d_A1+C_d_OH+C_i_s-C_i_NH4-C_d_H;
106 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
107 fu(11)=10^(-(A_I*(I_i^(1/2)/(1+I_i^(1/2))- 0.3*I_i)))-gamma_i_H;
108 fu(12)=0.5*(C_i_s+C_i_NH4+C_d_H+C_d_A+C_d_A1+C_d_OH)- I_i;
109 fu(13)=K_a_T*(C_d_a-C_d_A)/gamma_i_H/gamma_i_H/C_d_H-C_d_A;
```

```
110 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
111 fu(14)=K_a1_T*(C_d_a1-C_d_A1)/gamma_i_H/gamma_i_H/C_d_H-C_d_A1;
112 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
113 fu(15)=K_w_T/C_d_H/gamma_i_H/gamma_i_H-C_d_OH;
114 fu(16)=gamma_i_H*C_d_H*(C_i_s-C_i_NH4)/gamma_i_H/K_NH4CL-C_i_NH4;
115 fu(17)=o_s*n_s*C_mf_s*R*T+o_a*(C_mf_a+C_mf_H)*R*T+o_a1*(C_mf_a1+C_mf_H)*R*T-P_m;
116 %Charge Balance Equation at membrane active layer %%%%%%%%%%%%%%%%%%%%%%%%%
117 fu(18)=C_mf_A+C_mf_A1+C_mf_OH+C_mf_s-C_mf_NH4-C_mf_H;
118 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
119 fu(19)=10^(-(A_I*(I_mf^(1/2)/(1+I_mf^(1/2))- 0.3*I_mf)))-gamma_mf_H;
120 fu(20)=0.5*(C_mf_s+C_mf_NH4+C_mf_H+C_mf_A+C_mf_OH+C_mf_A1)-I_mf;
121 fu(21)=K_a_T*(C_mf_a-C_mf_A)/gamma_mf_H/gamma_mf_H/C_mf_H-C_mf_A;
122 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
123 fu(22)=K_a1_T*(C_mf_a1-C_mf_A1)/gamma_mf_H/gamma_mf_H/C_mf_H-C_mf_A1;
124 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
125 fu(23)=K_w_T/C_mf_H/gamma_mf_H/gamma_mf_H-C_mf_OH;
126 fu(24)=gamma_mf_H*C_mf_H*(C_mf_s-C_mf_NH4)/gamma_mf_H/K_NH4CL-C_mf_NH4;
127 fu(25)=(C_md_s-C_i_s)/2*D_S_T*(1/(J_w*(0.789*C_i_s+0.211*C_md_s)+J_s))...
128 +1/(J_w*(0.211*C_i_s+0.789*C_md_s)+ J_s))-S;
129 fu(26)=(C_d_s-C_md_s)/2*D_S_T*(1/(J_w*(0.789*C_md_s+0.211*C_d_s)+J_s))...
130 +1/(J_w*(0.211*C_md_s+0.789*C_d_s)+ J_s))-D_d_s/k_d;
131 fu(27)=Sh_d*D_d_s/d_h-k_d;
132 fu(28)=L*v_d*den_d/mul_d-Re_d;
133 fu(29)=(f_2+f)/2/W/D-v_d;
134 fu(30)=f+J_w*A_m-f_2;
135 % amonium chloride density and viscosity
136 fu(31)=6*exp((12.396*(C_d_s*53.491/den_d)^1.5039-1.7756)/(0.23471*(T-273)+1)/...
137 (-2.7591*(C_d_s*53.491/den_d)^2.8408+1))-mul_d;
138 fu(32)=den_w+0.2061e2*C_d_s-0.1577*C_d_s*(T-273)+1.553e-3*C_d_s*(T-273)^2-...
139 2.556*C_d_s^1.5+5.67e-2*C_d_s^1.5*(T-273)-5.082e-4*C_d_s^1.5*(T-273)^2-den_d;
140 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
141 fu(33)= 0.04*Re_d^(3/4)*Sc_d^(1/3)-Sh_d;
142 fu(34)= mul_d/den_d/D_d_s-Sc_d;
143 fu(35)=(0.5*(C_f_a-C_mf_a)*((D_1_T-D_HA_T)/...
144    (sqrt(1+4*K*(0.789*C_mf_a+0.211*C_f_a))*(J_w*(0.789*C_mf_a+0.211*C_f_a)-J_a))...
145    +(D_1_T-D_HA_T)/(sqrt(1+4*K*(0.211*C_mf_a+0.789*C_f_a))*(J_w*(0.211*C_mf_a...
146    +0.789*C_f_a)-J_a))+ D_HA_T/(J_w*(0.789*C_mf_a+0.211*C_f_a)-J_a)...
147    +D_HA_T/((J_w*(0.211*C_mf_a+0.789*C_f_a)-J_a))))+D_f_a/k_f_a;
148 %↙
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
149 fu(36)=(0.5*(C_f_a1-C_mf_a1)*((D_11_T-D_HA1_T)/...
150    (sqrt(1+4*K1*(0.789*C_mf_a1+0.211*C_f_a1))*(J_w*(0.789*C_mf_a1+0.211*C_f_a1)-↙
J_a1))...
151    +(D_11_T-D_HA1_T)/(sqrt(1+4*K1*(0.211*C_mf_a1+0.789*C_f_a1))*(J_w*(0.211↙
*C_mf_a1...
152    +0.789*C_f_a1)-J_a1))+ D_HA1_T/(J_w*(0.789*C_mf_a1+0.211*C_f_a1)-J_a1)...
153    +D_HA1_T/((J_w*(0.211*C_mf_a1+0.789*C_f_a1)-J_a1))))+D_f_a/k_f_a1;
154 %↙
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
155 fu(37)=Sh_f_a*D_f_a/d_h-k_f_a;
156 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
157 fu(38)=Sh_f_a1*D_f_a1/d_h-k_f_a1;
158 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
159 fu(39)=(D_1_T/(2*K*C_f_a)*(-1+sqrt(1+4*K*C_f_a))+D_HA_T/...
160    (4*K*C_f_a)*(-1+sqrt(1+4*K*C_f_a))^2)-D_f_a;
161 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
```

```
162 fu(40)=(D_11_T/(2*K1*C_f_a1)*(-1+sqrt(1+4*K1*C_f_a1))+D_HA1_T/...
163     (4*K1*C_f_a1)*(-1+sqrt(1+4*K1*C_f_a1))^2)-D_f_a1;
164 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
165 fu(41)=L*v_f*den_f/mul_f -Re_f;
166 fu(42)=(f+f_1)/W/D/2-v_f;
167 fu(43)=f-J_w*A_m-f_1;
168 fu(44)= 0.04*Re_f^(3/4)*Sc_f_a^(1/3)-Sh_f_a;
169 fu(45)= mul_f/den_f/D_f_a-Sc_f_a;
170 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
171 fu(46)= 0.04*Re_f^(3/4)*Sc_f_a1^(1/3)-Sh_f_a1;
172 fu(47)= mul_f/den_f/D_f_a1-Sc_f_a1;
173 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
174 fu(48)=(C_f_s-C_mf_s)/2*D_S_T*(1/(J_w*(0.789*C_mf_s+0.211*C_f_s)+J_s))...
175 +1/(J_w*(0.211*C_mf_s+0.789*C_f_s)+ J_s))+D_f_s/k_f_s;
176 fu(49)=Sh_f_s*D_f_s/d_h-k_f_s;
177 %Mole balance Equation
178 fu(50)=(V_f_0*C_f_a_0+C_f_ai*f_1*t)/(V_f_0+f_1*t)-C_f_a;
179 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
180 fu(51)=(V_f_0*C_f_a1_0+C_f_ai1*f_1*t)/(V_f_0+f_1*t)-C_f_a1;
181 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
182 fu(52)=C_f_s*f_1*t/(V_f_0+f_1*t)-C_f_si;
183 fu(53)=(C_f_a*f-J_a*A_m)/f_1-C_f_ai;
184 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
185 fu(54)=(C_f_a1*f-J_a1*A_m)/f_1-C_f_ai1;
186 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
187 fu(55)=(C_f_si*f+J_s*A_m)/f_1-C_f_s;
188 fu(56)=(C_d_s_0*V_d_0+C_d_si*f_2*t)/(V_d_0+f_2*t)-C_d_s;
189 fu(57)=(C_d_ai*f*t+C_d_ai*(V_d_0+f_2*t-f*t))/(f_2*t)-C_d_a;
190 fu(58)=(C_d_ai1*f*t+C_d_ai1*(V_d_0+f_2*t-f*t))/(f_2*t)-C_d_a1;
191 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
192 fu(59)=(C_d_s*f-J_s*A_m)/f_2-C_d_si;
193 fu(60)=(C_d_a*f_2-J_a*A_m)/f-C_d_ai;
194 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
195 fu(61)=(C_d_a1*f_2-J_a1*A_m)/f-C_d_ai1;
196 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
197 %additional Equation
198 fu(62)= 0.04*Re_f^(3/4)*Sc_f_s^(1/3)-Sh_f_s;
199 fu(63)= mul_f/den_f/D_f_s-Sc_f_s;
200 fu(64)=J_w*A_m*t*den_d/1000-w;
201 %ionic strenght in feed solution%%%%%%%%%%%%%%%%%%%%%
202 fu(65)=0.5*(C_f_si+C_f_NH4+C_f_H+C_f_OH+C_f_A+C_f_A1+C_f_HCO3+4*C_f_CO3)- I_f;
203 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
204 fu(66)=10^(-(A_I*(I_f^(1/2)/(1+I_f^(1/2))- 0.3*I_f)))-gamma_f_H;
205 %Charge Balance Equation %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
206 fu(67)=C_f_HCO3+2*C_f_CO3+C_f_A+C_f_A1+C_f_OH+C_f_si-C_f_NH4-C_f_H;
207 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
208 %New Equation for charge balance
209 fu(68)=K_a_T*(C_f_a-C_f_A)/gamma_f_H/gamma_f_H/C_f_H-C_f_A;
210 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
211 fu(69)=K_a1_T*(C_f_a1-C_f_A1)/gamma_f_H/gamma_f_H/C_f_H-C_f_A1;
212 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
213 fu(70)=K_CO2_T/gamma_f_H/gamma_f_H/C_f_H*(C_f_co2-C_f_HCO3)-C_f_HCO3;%******
214 fu(71)=K_HCO3_T*gamma_f_H/gamma_f_H/gamma_f_CO3/C_f_H*(C_f_HCO3-C_f_CO3)-C_f_CO3;%↙
******
215 fu(72)=10^(-(A_I*4*(I_f^(1/2)/(1+I_f^(1/2))- 0.3*I_f)))-gamma_f_CO3;%Davis
216 fu(73)=K_w_T/C_f_H/gamma_f_H/gamma_f_H-C_f_OH;
```

```
217 fu(74)=gamma_f_H*C_f_H*(C_f_si-C_f_NH4)/gamma_f_H/K_NH4CL-C_f_NH4;
218 fu(75)=pH_f+log10(gamma_f_H*C_f_H);
219
220
```

G.6 A mixture of 10 mM lactic acid and 5 mM butyric acid as feed solution and NH$_4$Cl as draw solution

```
 1 clc
 2 clear
 3 [g]=[1.594e-03    2.3745e+01    5.822e-01     4.053e-05  5.289e-01    2.376e-04↙
4.326e-12    1.021e-02    2.304e-09...
 4 2.298e-09    1.9193e-05    7.340e-01     5.289e-01  9.995e-01  1.248e-03    9.577↙
e-01    1.486e-03    9.909e-01...
 5 1.880e-01    6.3719e+02    3.700e+02     2.894e+04  1.503e+02  1.012e+03    4.839↙
e+00    1.581e+00    1.168e-05...
 6 1.723e-01    1.0002e-02    6.459e+02     2.759e+04  1.502e+02  1.579e+00    4.291↙
e+02    1.843e-01    1.657e-05...
 7 1.001e-02    5.7890e-06    1.424e-09     9.991e-01  9.612e-01  1.080e-03    1.237↙
e-03    6.245e+02    3.879e+02...
 8 2.305e-04    1.2317e-03    5.830e-09     1.660e-15  8.535e-01  5.789e-06    1.103↙
e-11    9.962e+02    2.927e+00...
 9 2.137e-09    1.2131e-09    5.289e-01     1.096e-03  1.153e-11  2.376e-04    4.994↙
e+00    1.013e-02    3.904e-08...
10 6.932e-11    3.7792e-08    2.376e-08     1.526e-04  1.000e-02  1.169e-05    1.724↙
e-01    6.457e+02    4.287e+02...
11 1.001e-02    2.3431e-08    1.515e-04];
12 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
13 ts=0.31:0.000025:1200;
14 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
15 for ti=1:numel(ts);
16 t=ts(ti);
17 options = optimset('Algorithm','Levenberg-Marquardt','TolFun',1*10^-18,'TolX',1*10^-↙
18 ...
18     ,'Maxiter',30,'Maxfun',15000);
19 [r,fval,exitflag,output] = fsolve(@(z)Lactic10_Butyric5(z,t),[g],options)
20 j(ti)=r(1,1);%Jw
21 c(ti)=r(1,29);%Cfa
22 d(ti)=r(1,46);%W
23 p(ti)=r(1,54);%pH
24 T(ti)=t;
25 e(ti)=r(1,68);%Cfa1
26 % For Rejection
27 O(ti)=(1-r(1,10)/r(1,31))*100;%Rej
28 O1(ti)=(1-r(1,65)/r(1,68))*100;%Rej1
29 %For Concentration Performance
30 K4(ti)=r(1,29)/0.01;%Cfa/Cf0
31 K6(ti)=r(1,68)/0.01;%Cfa1/Cf0
32 K1(ti)=r(1,1);
33 [g]=[r];
34 %for Plotting
35 [A(ti)]=t;
36 [B(ti)]=d(ti);
37 [C(ti)]=p(ti);
38 % For Rejection
39 [M(ti)]=O(ti);
40 [N(ti)]=K1(ti);
41 [K5(ti)]=K4(ti); %Cfa/Cf0
42 disp(ti)
43 fprintf('\n     Time(min)         J_w(L/dm2/min)              pH              Cf,a(mole/l)↙
W(kg)            Cf,a1(mole/l)\n');
44
45 fprintf('\n%11.4f%20.4e%19.4f%19.4e%19.4e%19.4e\n' ,T(ti),j(ti),p(ti),c(ti),d(ti),e↙
(ti));
```

```matlab
46 format short e
47
48 %Result of Best guess
49 fprintf('\n%8.3e%15.4e%13.3e%14.3e%11.3e%13.3e%13.3e%13.3e%13.3e%\n'...
50     ,r(1,1),r(1,2),r(1,3),r(1,4),r(1,5),r(1,6),r(1,7),r(1,8),r(1,9));
51
52  fprintf('\n%8.3e%15.4e%13.3e%14.3e%11.3e%13.3e%13.3e%13.3e%13.3e%\n'...
53     ,r(1,10),r(1,11),r(1,12),r(1,13),r(1,14),r(1,15),r(1,16),r(1,17),r(1,18));
54
55 fprintf('\n%8.3e%15.4e%13.3e%14.3e%11.3e%13.3e%13.3e%13.3e%13.3e%\n'...
56     ,r(1,19),r(1,20),r(1,21),r(1,22),r(1,23),r(1,24),r(1,25),r(1,26),r(1,27));
57
58 fprintf('\n%8.3e%15.4e%13.3e%14.3e%11.3e%13.3e%13.3e%13.3e%13.3e%\n'...
59    ,r(1,28),r(1,29),r(1,30),r(1,31),r(1,32),r(1,33),r(1,34),r(1,35),r(1,36));
60
61 fprintf('\n%8.3e%15.4e%13.3e%14.3e%11.3e%13.3e%13.3e%13.3e%13.3e%13.3e%\n'...
62     ,r(1,37),r(1,38),r(1,39),r(1,40),r(1,41),r(1,42),r(1,43),r(1,44),r(1,45));
63
64 fprintf('\n%8.3e%15.4e%13.3e%14.3e%11.3e%13.3e%13.3e%13.3e%13.3e%\n'...
65     ,r(1,46),r(1,47),r(1,48),r(1,49),r(1,50),r(1,51),r(1,52),r(1,53),r(1,54));
66
67 fprintf('\n%8.3e%15.4e%13.3e%14.3e%11.3e%13.3e%13.3e%13.3e%13.3e%\n'...
68     ,r(1,55),r(1,56),r(1,57),r(1,58),r(1,59),r(1,60),r(1,61),r(1,62),r(1,63));
69
70 fprintf('\n%8.3e%15.4e%13.3e%14.3e%11.3e%13.3e%13.3e%13.3e%13.3e%\n'...
71    ,r(1,64),r(1,65),r(1,66),r(1,67),r(1,68),r(1,69),r(1,70),r(1,71),r(1,72));
72
73 fprintf('\n%8.3e%15.4e%13.3e%\n'...
74     ,r(1,73),r(1,74),r(1,75));
75
76 end
77 %Time Interval%%%%%%%%%%%%%%%%%%%%%%%%%%%
78 Ai=0:30:1200;
79 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
80 format short
81 Ci=spline(A,C,Ai);
82 format short e
83 Bi=spline(A,B,Ai);
84 %Experiment Value W and pH%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
85  F=[  0     0.0160    0.0310    0.0450    0.0600    0.0740    0.0880    0.1020
0.1150    0.1280    0.1410    0.1530    0.1650...
86 0.1780    0.1900    0.2010    0.2130    0.2240    0.2360    0.2500    0.2630
0.2750    0.2870    0.2970    0.3080    0.3180...
87 0.3280    0.3380    0.3480    0.3570    0.3670    0.3760    0.3860    0.3950
0.4040    0.4120    0.4220    0.4300    0.4390...
88 0.4470    0.4560];
89
90 G=[  2.9910    2.9270    2.9520    2.9420    2.9230    2.9290    2.9410    2.9190
2.9230    2.9200    2.9210    2.9060    2.9120...
91 2.9150    2.9120    2.8920    2.8920    2.8880    2.8860    2.8920    2.8920    2.8850
2.8770    2.8780    2.8680    2.8670    2.8580...
92 2.8530    2.8540    2.8450    2.8430    2.8400    2.8330    2.8300    2.8230
2.8200    2.8150    2.8090    2.8050    2.7970...
93 2.7920    2.7890];
94
95 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
 96
 97 F_m=mean(F);
 98 G_m=mean(G);
 99 sum1=0;sum2=0;
100 sum3=0;sum4=0;
101
102 %%% Number of Data %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
103 for k=1:1:41
104 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
105 sum1=(Bi(1,k)-F(1,k))^2+sum1;
106 sum2=(Bi(1,k)-F_m)^2+sum2;
107 sum3=(Ci(1,k)-G(1,k))^2+sum3;
108 sum4=(Ci(1,k)-G_m)^2+sum4;
109 end
110
111 format short
112 R2_W=1-sum1/sum2;
113 R2_pH=1-sum3/sum4;
114 MSE_W=sum1/k;
115 MSE_pH=sum3/k;
116 RMSE_W=sqrt(MSE_W)
117 RMSE_pH=sqrt(MSE_pH)
118 SEP_W=RMSE_W/F_m*100
119 SEP_pH=RMSE_pH/G_m*100
120
121 % Plotting
122 subplot(2,3,1)
123 plot([A]/60,[B],'-g',[Ai]/60,[F],'xr')
124 xlabel('Time (hr)')
125 ylabel('Weight (Kg)')
126 title('Weight Change as a Function of Time')
127 legend('Model','Experiment',2)
128
129 subplot(2,3,2)
130 plot([A]/60,[C],'-g',[Ai]/60,[G],'xr')
131 xlabel('Time (hr)')
132 ylabel('pH')
133 title('pH as a Function of Time')
134 legend('Model','Experiment',1)
135
136 subplot(2,3,3)
137 plot([A]/60,[M],'-g')
138 xlabel('Time (hr)')
139 ylabel('%Rejection1')
140 title('%Rejection1 as a Function of Time')
141 legend('Model',1)
142
143 subplot(2,3,4)
144 plot([A]/60,[O1],'-g')
145 xlabel('Time (hr)')
146 ylabel('%Rejection2')
147 title('%Rejection2 as a Function of Time')
148 legend('Model',1)
149
150 subplot(2,3,5)
151 plot([A]/60,[K5],'-g')
```

```
152 xlabel('Time (hr)')
153 ylabel('Cfa/Cf0')
154 title('Cfa/Cf0 as a Function of Time')
155 legend('Model',1)
156
157 subplot(2,3,6)
158 plot([A]/60,[K6],'-g')
159 xlabel('Time (hr)')
160 ylabel('Cfa1/Cf0')
161 title('Cfa/Cf01 as a Function of Time')
162 legend('Model',1)
163
164 fprintf('\n       \n')
165 fprintf('\n       \n')
166 fprintf('\n CONCLUSION     \n')
167 fprintf('\n      R2_W              MSE_W                    R2_pH             MSE_pH\n');
168 fprintf('\n%11.3f%20.4e%19.4f%19.4e%\n',R2_W,MSE_W,R2_pH,MSE_pH)
169 fprintf('\n       \n')
170
171 fprintf('\n     Time(min)        J_w(L/dm2/min)               pH              Cf,a(mole/l)↙
W(kg)        Cf,a1(mole/l)\n');
172 for x=1:1:200000000;
173 z=x*40000;
174 fprintf('\n%11.3f%20.4e%19.4f%19.4e%19.4e%19.4e%\n',T(z),j(z),p(z),c(z),d(z),e(z));
175 end
176
177
178
```

```matlab
1
2  function fu=Lactic10_Butyric5(z,t)
3  %Independent Variables
4  V_f_0=1;C_d_s_0=1;V_d_0=0.5;
5  %Fixed Variable
6  T=301;f=1.58;
7  L=0.9207;D=0.023;W=0.4572;A_m=42*10^(-2);d_h=0.0438;
8  % Acid Variables
9  %↙
10 C_f_a_0=10*10^(-3);
11 C_f_a1_0=5*10^(-3);
12 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
13 B_a=0.153/60/100;
14 B_a1=0.639/60/100;
15 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
16 D_A_298=6.198*10^(-6);D_HA_298=4.5834*10^(-6);%Lactic Physicocmical Property
17 D_A1_298=5.208*10^(-6);D_HA1_298=5.429*10^(-6);%Butyric Physicocmical Property
18 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
19 K_a_T=10^-(1286.49/T-4.8607+0.014776*T);%Lactic Acid
20 K_a1_T=10^-(1033.39/T-2.6215+0.013334*T);%Butyric Acid
21 %↙
22 %Constant Variable
23 B_s=0.460/60/100;
24 A=0.413/60/100;
25 o_s=0.897;%amonium chloride coefficient
26 S=491*10^-5;
27 D_NH4_298=117.42e-7;D_Cl_298=121.92e-7;%amonium & chloride diffusion coefficient
28 den_w=999.65+2.0438/10*(T-273)-6.174*10^-2*(T-273)^1.5;
29 D_H_298=5.587*10^(-5);
30 visco_298=((298-273)+246)/((0.05594*(298-273)+5.2842)*(298-273)+137.37);
31 visco_T=((T-273)+246)/((0.05594*(T-273)+5.2842)*(T-273)+137.37);
32 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
33 o_a=1;n_s=2;R=0.08314;o_a1=1;
34 %Temperature Correction
35 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
36 K_CO2_T=10^-(3404.71/T-14.8435+0.032786*T);
37 K_HCO3_T=10^-(2902.39/T-6.4980+0.02379*T);
38 K_w_T=10^(-4470.99/T+6.0875-0.017060*T);
39 K_NH4CL=10^-(2835.76/T-0.6322+0.001225*T);%Amonium Chloride
40 E=2727.586+0.6224107*T-466.9151*log(T)-52000.87/T;
41 d=1-(((T-273)-3.9863)^2*((T-273)+288.9414))/(508929.2*((T-273)+68.12963))...
42 +0.011445*exp((-374.3)/(T-273));
43 A_I=(1.82483*10^6*d^0.5)/(E*T)^1.5;
44 K=1/K_a_T;
45 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
46 K1=1/K_a1_T;
47 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
48 D_H_T=D_H_298*T/298*visco_298/visco_T ;
49 D_A_T=D_A_298*T/298*visco_298/visco_T;
50 D_HA_T=D_HA_298*T/298*visco_298/visco_T;
51 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
52 D_A1_T=D_A1_298*T/298*visco_298/visco_T;
53 D_HA1_T=D_HA1_298*T/298*visco_298/visco_T;
```

```
54 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
55 D_NH4_T=D_NH4_298*T/298*visco_298/visco_T;
56 D_Cl_T=D_Cl_298*T/298*visco_298/visco_T;
57 D_1_T=2/(1/D_H_T +1/D_A_T);
58 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
59 D_11_T=2/(1/D_H_T +1/D_A1_T);
60 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
61 D_S_T=2/(1/D_NH4_T +1/D_Cl_T);
62 D_d_s=D_S_T;
63 D_f_s=D_S_T;
64 K_H=0.034*exp(-2400*(1/T-1/298));%******************
65 C_f_co2=10^-3.408*K_H;%***************************
66 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
67 %Unknown Variable
68 J_w=z(1);P_i=z(2);P_m=z(3);J_s=z(4);C_i_s=z(5);
69 C_mf_s=z(6);J_a=z(7);C_mf_a=z(8);C_i_a=z(9);C_d_a=z(10);
70 C_d_H=z(11);gamma_i_H=z(12);I_i=z(13);C_d_s=z(14);
71 C_mf_H=z(15);gamma_mf_H=z(16);I_mf=z(17);C_md_s=z(18);
72 k_d=z(19);Sh_d=z(20);Sc_d=z(21);Re_d=z(22);
73 v_d=z(23);den_d=z(24);mul_d=z(25);f_2=z(26);D_f_a=z(27);
74 k_f_a=z(28);C_f_a=z(29);Sh_f_a=z(30);Re_f=z(31);v_f=z(32);
75 f_1=z(33);Sc_f_a=z(34);k_f_s=z(35);C_f_s=z(36);
76 C_f_ai=z(37);C_f_si=z(38);C_d_ai=z(39);C_d_si=z(40);
77 gamma_f_H=z(41);C_f_A=z(42);I_f=z(43);Sh_f_s=z(44);Sc_f_s=z(45);
78 w=z(46);C_f_H=z(47);C_f_HCO3=z(48);C_f_CO3=z(49);gamma_f_CO3=z(50);
79 C_f_NH4=z(51);C_f_OH=z(52);den_f=z(53);pH_f=z(54);
80 % for charge balance of NH4
81 C_d_A=z(55);C_d_OH=z(56);C_i_NH4=z(57);C_mf_A=z(58);C_mf_OH=z(59);
82 C_mf_NH4=z(60);mul_f=z(61);
83 % for two acid
84 C_mf_a1=z(62);C_i_a1=z(63);J_a1=z(64);C_d_a1=z(65);C_d_A1=z(66);
85 C_mf_A1=z(67);C_f_a1=z(68);D_f_a1=z(69);k_f_a1=z(70);Sh_f_a1=z(71);
86 Sc_f_a1=z(72);C_f_ai1=z(73);C_d_ai1=z(74);C_f_A1=z(75);
87 format short e
88 %Math Model
89 fu(1)=A*(P_i-P_m )-J_w;
90 fu(2)= B_s *(C_i_s-C_mf_s)-J_s;
91 fu(3)= B_a*(C_mf_a-C_i_a)-J_a;
92 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
93 fu(4)= B_a1*(C_mf_a1-C_i_a1)-J_a1;
94 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
95 fu(5)=C_d_a-C_i_a;
96 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
97 fu(6)=C_d_a1-C_i_a1;
98 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
99 fu(7)=J_a/J_w-C_d_a;
100 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
101 fu(8)=J_a1/J_w-C_d_a1;
102 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
103 fu(9)= o_s*n_s*C_i_s*R*T+o_a*(C_d_a+C_d_H)*R*T+o_a1*(C_d_a1+C_d_H)*R*T-P_i;
104 %Charge Balance Equation at interface %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
105 fu(10)=C_d_A+C_d_A1+C_d_OH+C_i_s-C_i_NH4-C_d_H;
106 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
107 fu(11)=10^(-(A_I*(I_i^(1/2)/(1+I_i^(1/2))- 0.3*I_i)))-gamma_i_H;
108 fu(12)=0.5*(C_i_s+C_i_NH4+C_d_H+C_d_A+C_d_A1+C_d_OH)- I_i;
109 fu(13)=K_a_T*(C_d_a-C_d_A)/gamma_i_H/gamma_i_H/C_d_H-C_d_A;
```

```
110 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
111 fu(14)=K_a1_T*(C_d_a1-C_d_A1)/gamma_i_H/gamma_i_H/C_d_H-C_d_A1;
112 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
113 fu(15)=K_w_T/C_d_H/gamma_i_H/gamma_i_H-C_d_OH;
114 fu(16)=gamma_i_H*C_d_H*(C_i_s-C_i_NH4)/gamma_i_H/K_NH4CL-C_i_NH4;
115 fu(17)=o_s*n_s*C_mf_s*R*T+o_a*(C_mf_a+C_mf_H)*R*T+o_a1*(C_mf_a1+C_mf_H)*R*T-P_m;
116 %Charge Balance Equation at membrane active layer %%%%%%%%%%%%%%%%%%%%%%%%%%
117 fu(18)=C_mf_A+C_mf_A1+C_mf_OH+C_mf_s-C_mf_NH4-C_mf_H;
118 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
119 fu(19)=10^(-(A_I*(I_mf^(1/2)/(1+I_mf^(1/2))- 0.3*I_mf)))-gamma_mf_H;
120 fu(20)=0.5*(C_mf_s+C_mf_NH4+C_mf_H+C_mf_A+C_mf_OH+C_mf_A1)-I_mf;
121 fu(21)=K_a_T*(C_mf_a-C_mf_A)/gamma_mf_H/gamma_mf_H/C_mf_H-C_mf_A;
122 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
123 fu(22)=K_a1_T*(C_mf_a1-C_mf_A1)/gamma_mf_H/gamma_mf_H/C_mf_H-C_mf_A1;
124 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
125 fu(23)=K_w_T/C_mf_H/gamma_mf_H/gamma_mf_H-C_mf_OH;
126 fu(24)=gamma_mf_H*C_mf_H*(C_mf_s-C_mf_NH4)/gamma_mf_H/K_NH4CL-C_mf_NH4;
127 fu(25)=(C_md_s-C_i_s)/2*D_S_T*(1/(J_w*(0.789*C_i_s+0.211*C_md_s)+J_s))...
128 +1/(J_w*(0.211*C_i_s+0.789*C_md_s)+ J_s))-S;
129 fu(26)=(C_d_s-C_md_s)/2*D_S_T*(1/(J_w*(0.789*C_md_s+0.211*C_d_s)+J_s))...
130 +1/(J_w*(0.211*C_md_s+0.789*C_d_s)+ J_s))-D_d_s/k_d;
131 fu(27)=Sh_d*D_d_s/d_h-k_d;
132 fu(28)=L*v_d*den_d/mul_d-Re_d;
133 fu(29)=(f_2+f)/2/W/D-v_d;
134 fu(30)=f+J_w*A_m-f_2;
135 % amonium chloride density and viscosity
136 fu(31)=6*exp((12.396*(C_d_s*53.491/den_d)^1.5039-1.7756)/(0.23471*(T-273)+1)/...
137 (-2.7591*(C_d_s*53.491/den_d)^2.8408+1))-mul_d;
138 fu(32)=den_w+0.2061e2*C_d_s-0.1577*C_d_s*(T-273)+1.553e-3*C_d_s*(T-273)^2-...
139 2.556*C_d_s^1.5+5.67e-2*C_d_s^1.5*(T-273)-5.082e-4*C_d_s^1.5*(T-273)^2-den_d;
140 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
141 fu(33)= 0.04*Re_d^(3/4)*Sc_d^(1/3)-Sh_d;
142 fu(34)= mul_d/den_d/D_d_s-Sc_d;
143 fu(35)=(0.5*(C_f_a-C_mf_a)*((D_1_T-D_HA_T)/...
144     (sqrt(1+4*K*(0.789*C_mf_a+0.211*C_f_a))*(J_w*(0.789*C_mf_a+0.211*C_f_a)-J_a))...
145     +(D_1_T-D_HA_T)/(sqrt(1+4*K*(0.211*C_mf_a+0.789*C_f_a))*(J_w*(0.211*C_mf_a...
146     +0.789*C_f_a)-J_a))+ D_HA_T/(J_w*(0.789*C_mf_a+0.211*C_f_a)-J_a)...
147     +D_HA_T/((J_w*(0.211*C_mf_a+0.789*C_f_a)-J_a))))+D_f_a/k_f_a;
148 %↙
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
149 fu(36)=(0.5*(C_f_a1-C_mf_a1)*((D_11_T-D_HA1_T)/...
150     (sqrt(1+4*K1*(0.789*C_mf_a1+0.211*C_f_a1))*(J_w*(0.789*C_mf_a1+0.211*C_f_a1)-↙
J_a1))...
151     +(D_11_T-D_HA1_T)/(sqrt(1+4*K1*(0.211*C_mf_a1+0.789*C_f_a1))*(J_w*(0.211↙
*C_mf_a1...
152     +0.789*C_f_a1)-J_a1))+ D_HA1_T/(J_w*(0.789*C_mf_a1+0.211*C_f_a1)-J_a1)...
153     +D_HA1_T/((J_w*(0.211*C_mf_a1+0.789*C_f_a1)-J_a1))))+D_f_a1/k_f_a1;
154 %↙
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
155 fu(37)=Sh_f_a*D_f_a/d_h-k_f_a;
156 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
157 fu(38)=Sh_f_a1*D_f_a1/d_h-k_f_a1;
158 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
159 fu(39)=(D_1_T/(2*K*C_f_a)*(-1+sqrt(1+4*K*C_f_a))+D_HA_T/...
160     (4*K*C_f_a)*(-1+sqrt(1+4*K*C_f_a))^2)-D_f_a;
161 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
```

```
162 fu(40)=(D_11_T/(2*K1*C_f_a1)*(-1+sqrt(1+4*K1*C_f_a1))+D_HA1_T/...
163    (4*K1*C_f_a1)*(-1+sqrt(1+4*K1*C_f_a1))^2)-D_f_a1;
164 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
165 fu(41)=L*v_f*den_f/mul_f -Re_f;
166 fu(42)=(f+f_1)/W/D/2-v_f;
167 fu(43)=f-J_w*A_m-f_1;
168 fu(44)= 0.04*Re_f^(3/4)*Sc_f_a^(1/3)-Sh_f_a;
169 fu(45)= mul_f/den_f/D_f_a-Sc_f_a;
170 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
171 fu(46)= 0.04*Re_f^(3/4)*Sc_f_a1^(1/3)-Sh_f_a1;
172 fu(47)= mul_f/den_f/D_f_a1-Sc_f_a1;
173 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
174 fu(48)=(C_f_s-C_mf_s)/2*D_S_T*(1/(J_w*(0.789*C_mf_s+0.211*C_f_s)+J_s)...
175 +1/(J_w*(0.211*C_mf_s+0.789*C_f_s)+ J_s))+D_f_s/k_f_s;
176 fu(49)=Sh_f_s*D_f_s/d_h-k_f_s;
177 %Mole balance Equation
178 fu(50)=(V_f_0*C_f_a_0+C_f_ai*f_1*t)/(V_f_0+f_1*t)-C_f_a;
179 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
180 fu(51)=(V_f_0*C_f_a1_0+C_f_ai1*f_1*t)/(V_f_0+f_1*t)-C_f_a1;
181 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
182 fu(52)=C_f_s*f_1*t/(V_f_0+f_1*t)-C_f_si;
183 fu(53)=(C_f_a*f-J_a*A_m)/f_1-C_f_ai;
184 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
185 fu(54)=(C_f_a1*f-J_a1*A_m)/f_1-C_f_ai1;
186 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
187 fu(55)=(C_f_si*f+J_s*A_m)/f_1-C_f_s;
188 fu(56)=(C_d_s_0*V_d_0+C_d_si*f_2*t)/(V_d_0+f_2*t)-C_d_s;
189 fu(57)=(C_d_ai*f*t+C_d_ai*(V_d_0+f_2*t-f*t))/(f_2*t)-C_d_a;
190 fu(58)=(C_d_ai1*f*t+C_d_ai1*(V_d_0+f_2*t-f*t))/(f_2*t)-C_d_a1;
191 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
192 fu(59)=(C_d_s*f-J_s*A_m)/f_2-C_d_si;
193 fu(60)=(C_d_a*f_2-J_a*A_m)/f-C_d_ai;
194 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
195 fu(61)=(C_d_a1*f_2-J_a1*A_m)/f-C_d_ai1;
196 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
197 %additional Equation
198 fu(62)= 0.04*Re_f^(3/4)*Sc_f_s^(1/3)-Sh_f_s;
199 fu(63)= mul_f/den_f/D_f_s-Sc_f_s;
200 fu(64)=J_w*A_m*t*den_d/1000-w;
201 %ionic strenght in feed solution%%%%%%%%%%%%%%%%%%%
202 fu(65)=0.5*(C_f_si+C_f_NH4+C_f_H+C_f_OH+C_f_A+C_f_A1+C_f_HCO3+4*C_f_CO3)- I_f;
203 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
204 fu(66)=10^(-(A_I*(I_f^(1/2)/(1+I_f^(1/2))- 0.3*I_f)))-gamma_f_H;
205 %Charge Balance Equation %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
206 fu(67)=C_f_HCO3+2*C_f_CO3+C_f_A+C_f_A1+C_f_OH+C_f_si-C_f_NH4-C_f_H;
207 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
208 %New Equation for charge balance
209 fu(68)=K_a_T*(C_f_a-C_f_A)/gamma_f_H/gamma_f_H/C_f_H-C_f_A;
210 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
211 fu(69)=K_a1_T*(C_f_a1-C_f_A1)/gamma_f_H/gamma_f_H/C_f_H-C_f_A1;
212 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
213 fu(70)=K_CO2_T/gamma_f_H/gamma_f_H/C_f_H*(C_f_co2-C_f_HCO3)-C_f_HCO3;%******
214 fu(71)=K_HCO3_T*gamma_f_H/gamma_f_H/gamma_f_CO3/C_f_H*(C_f_HCO3-C_f_CO3)-C_f_CO3;%↙
****** 
215 fu(72)=10^(-(A_I*4*(I_f^(1/2)/(1+I_f^(1/2))- 0.3*I_f)))-gamma_f_CO3;%Davis
216 fu(73)=K_w_T/C_f_H/gamma_f_H/gamma_f_H-C_f_OH;
```

```
217 fu(74)=gamma_f_H*C_f_H*(C_f_si-C_f_NH4)/gamma_f_H/K_NH4CL-C_f_NH4;
218 fu(75)=pH_f+log10(gamma_f_H*C_f_H);
219
220
```

G.7 A mixture of 10 mM acetic acid and 10 mM butyric acid as feed solution and NH$_4$Cl as draw solution

```
 1 clc
 2 clear
 3 [g]=[1.596e-03    2.3736e+01    5.483e-01    4.051e-05  5.287e-01    2.346e-04↙
3.840e-11   1.013e-02   1.962e-08...
 4 1.837e-08    1.9184e-05    7.340e-01    7.340e-01  5.287e-01    9.998e-01    5.845↙
e-04    9.680e-01    9.680e-01...
 5 8.191e-04    9.9110e-01    1.880e-01    6.372e+02  3.700e+02    2.894e+04    1.503↙
e+02    1.012e+03    4.839e+00...
 6 1.581e+00    1.1682e-05    1.723e-01    1.000e-02  6.459e+02    2.759e+04    1.502↙
e+02    1.579e+00    4.291e+02...
 7 1.843e-01    1.3668e-05    1.001e-02    2.893e-06  7.723e-09    9.993e-01    9.729↙
e-01    3.106e-04    9.729e-01...
 8 5.800e-04    3.4869e-12    6.245e+02    3.879e+02  1.154e-04    5.771e-04    9.729↙
e-01    9.729e-01    1.214e-08...
 9 5.354e-15    8.9579e-01    2.893e-06    2.298e-11  9.691e-03    9.962e+02    3.251↙
e+00    7.340e-01    7.340e-01...
10 1.155e-08    1.2134e-09    5.287e-01    1.917e-05  6.820e-09    3.138e-04    2.407↙
e-11    2.346e-04    9.680e-01...
11 9.680e-01    2.7921e-10    9.820e-03    9.729e-01  4.994e+00    1.018e-02    5.017↙
e-09    1.059e-11    4.901e-09...
12 2.898e-09    7.3399e-01    2.003e-09    2.707e-04  9.680e-01    9.914e-03    1.000↙
e-02    1.169e-05    1.724e-01...
13 6.457e+02    4.2873e+02    1.001e-02    2.060e-09  2.665e-04    9.729e-01    9.735↙
e-03    1.439e-05    1.214e-08];
14 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
15 ts=0.17:0.0006:1800;
16 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
17 for ti=1:numel(ts);
18 t=ts(ti);
19 options = optimset('Algorithm','Levenberg-Marquardt','TolFun',1*10^-15,'TolX',1*10^-↙
15 ...
20     ,'Maxiter',25,'Maxfun',15000);
21 [r] = fsolve(@(z)Acetic10_Butyric10(z,t),[g],options)
22 j(ti)=r(1,1);%Jw
23 c(ti)=r(1,31);%Cfa
24 d(ti)=r(1,50);%W
25 p(ti)=r(1,61);%pH
26 T(ti)=t;
27 e(ti)=r(1,88);%Cfa1
28 % For Rejection
29 O(ti)=(1-r(1,10)/r(1,31))*100;%Rej
30 O1(ti)=(1-r(1,81)/r(1,88))*100;%Rej1
31 %For Concentration Performance
32 K4(ti)=r(1,31)/0.01;%Cfa/Cf0
33 K6(ti)=r(1,88)/0.01;%Cfa1/Cf0
34 K1(ti)=r(1,1);
35 [g]=[r];
36 %for Plotting
37 [A(ti)]=t;
38 [B(ti)]=d(ti);
39 [C(ti)]=p(ti);
40 % For Rejection
41 [M(ti)]=O(ti);
42 [N(ti)]=K1(ti);
43 [K5(ti)]=K4(ti); %Cfa/Cf0
44 disp(ti)
```

```
45 fprintf('\n     Time(min)       J_w(L/dm2/min)           pH          Cf,a(mole/l)
W(kg)              Cf,a1(mole/l)\n');
46
47 fprintf('\n%11.4f%20.4e%19.4f%19.4e%19.4e%19.4e%\n' ,T(ti),j(ti),p(ti),c(ti),d(ti),e
(ti));
48
49 format short e
50
51 %Result of Best guess
52 fprintf('\n%8.3e%15.4e%13.3e%14.3e%11.3e%13.3e%13.3e%13.3e%13.3e%\n'...
53      ,r(1,1),r(1,2),r(1,3),r(1,4),r(1,5),r(1,6),r(1,7),r(1,8),r(1,9));
54
55  fprintf('\n%8.3e%15.4e%13.3e%14.3e%11.3e%13.3e%13.3e%13.3e%13.3e%\n'...
56      ,r(1,10),r(1,11),r(1,12),r(1,13),r(1,14),r(1,15),r(1,16),r(1,17),r(1,18));
57
58 fprintf('\n%8.3e%15.4e%13.3e%14.3e%11.3e%13.3e%13.3e%13.3e%13.3e%\n'...
59      ,r(1,19),r(1,20),r(1,21),r(1,22),r(1,23),r(1,24),r(1,25),r(1,26),r(1,27));
60
61 fprintf('\n%8.3e%15.4e%13.3e%14.3e%11.3e%13.3e%13.3e%13.3e%13.3e%\n'...
62     ,r(1,28),r(1,29),r(1,30),r(1,31),r(1,32),r(1,33),r(1,34),r(1,35),r(1,36));
63
64 fprintf('\n%8.3e%15.4e%13.3e%14.3e%11.3e%13.3e%13.3e%13.3e%13.3e%13.3e%13.3e%\n'...
65      ,r(1,37),r(1,38),r(1,39),r(1,40),r(1,41),r(1,42),r(1,43),r(1,44),r(1,45));
66
67 fprintf('\n%8.3e%15.4e%13.3e%14.3e%11.3e%13.3e%13.3e%13.3e%13.3e%\n'...
68     ,r(1,46),r(1,47),r(1,48),r(1,49),r(1,50),r(1,51),r(1,52),r(1,53),r(1,54));
69
70 fprintf('\n%8.3e%15.4e%13.3e%14.3e%11.3e%13.3e%13.3e%13.3e%13.3e%\n'...
71      ,r(1,55),r(1,56),r(1,57),r(1,58),r(1,59),r(1,60),r(1,61),r(1,62),r(1,63));
72
73 fprintf('\n%8.3e%15.4e%13.3e%14.3e%11.3e%13.3e%13.3e%13.3e%13.3e%\n'...
74     ,r(1,64),r(1,65),r(1,66),r(1,67),r(1,68),r(1,69),r(1,70),r(1,71),r(1,72));
75
76 fprintf('\n%8.3e%15.4e%13.3e%14.3e%11.3e%13.3e%13.3e%13.3e%13.3e%\n'...
77     ,r(1,73),r(1,74),r(1,75),r(1,76),r(1,77),r(1,78),r(1,79),r(1,80),r(1,81));
78
79 fprintf('\n%8.3e%15.4e%13.3e%14.3e%11.3e%13.3e%13.3e%13.3e%13.3e%\n'...
80     ,r(1,82),r(1,83),r(1,84),r(1,85),r(1,86),r(1,87),r(1,88),r(1,89),r(1,90));
81
82 fprintf('\n%8.3e%15.4e%13.3e%14.3e%11.3e%13.3e%13.3e%13.3e%13.3e%\n'...
83     ,r(1,91),r(1,92),r(1,93),r(1,94),r(1,95),r(1,96),r(1,97),r(1,98),r(1,99));
84 end
85
86 %Time Interval%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
87 Ai=0:30:1800;
88 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
89 format short
90 Ci=spline(A,C,Ai);
91 format short e
92 Bi=spline(A,B,Ai);
93
94 %Experiment Value W and pH%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
95 F=[   0     0.0120    0.0220    0.0340    0.0460    0.0580    0.0680    0.0800
0.0910    0.1020    0.1130    0.1240    0.1340...
96 0.1450    0.1560    0.1660    0.1760    0.1870    0.1970    0.2080    0.2180
0.2280    0.2370    0.2470    0.2570    0.2670...
```

```
 97 0.2760    0.2860    0.2950    0.3040    0.3140    0.3230    0.3330    0.3420↙
0.3510    0.3610    0.3690    0.3780    0.3880...
 98 0.3970    0.4050    0.4130    0.4220    0.4310    0.4390    0.4480    0.4570↙
0.4640    0.4720    0.4800    0.4890    0.4960...
 99 0.5040    0.5120    0.5200    0.5270    0.5340    0.5410    0.5480    0.5550↙
0.5620];
100
101 G=[      3.3010    3.2990    3.2960    3.2970  3.2950    3.2930    3.2890    3.2880
3.2820    3.2780    3.2730    3.2680    3.2640...
102 3.2610    3.2570    3.2530    3.2490    3.2450    3.2430    3.2390    3.2360↙
3.2330    3.2290    3.2260    3.2220    3.2180...
103 3.2150    3.2110    3.2060    3.2020    3.1960    3.1910    3.1860    3.1810↙
3.1760    3.1710    3.1650    3.1600    3.1550...
104 3.1500    3.1460    3.1420    3.1380    3.1350    3.1330    3.1290    3.1240↙
3.1230    3.1220    3.1180    3.1140    3.1110...
105 3.1060    3.1060    3.1050    3.1020    3.1010    3.0980    3.0960    3.0940↙
3.0930];
106 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
107 F_m=mean(F);
108 G_m=mean(G);
109 sum1=0;sum2=0;
110 sum3=0;sum4=0;
111 %%% Number of Data %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
112 for k=1:1:61
113 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
114 sum1=(Bi(1,k)-F(1,k))^2+sum1;
115 sum2=(Bi(1,k)-F_m)^2+sum2;
116 sum3=(Ci(1,k)-G(1,k))^2+sum3;
117 sum4=(Ci(1,k)-G_m)^2+sum4;
118 end
119
120 format short
121 R2_W=1-sum1/sum2;
122 R2_pH=1-sum3/sum4;
123 MSE_W=sum1/k;
124 MSE_pH=sum3/k;
125 RMSE_W=sqrt(MSE_W)
126 RMSE_pH=sqrt(MSE_pH)
127 SEP_W=RMSE_W/F_m*100
128 SEP_pH=RMSE_pH/G_m*100
129
130 % Plotting
131 subplot(2,3,1)
132 plot([A]/60,[B],'-g',[Ai]/60,[F],'xr')
133 xlabel('Time (hr)')
134 ylabel('Weight (Kg)')
135 title('Weight Change as a Function of Time')
136 legend('Model','Experiment',2)
137
138 subplot(2,3,2)
139 plot([A]/60,[C],'-g',[Ai]/60,[G],'xr')
140 xlabel('Time (hr)')
141 ylabel('pH')
142 title('pH as a Function of Time')
143 legend('Model','Experiment',1)
144
```

```matlab
145 subplot(2,3,3)
146 plot([A]/60,[M],'-g')
147 xlabel('Time (hr)')
148 ylabel('%Rejection1')
149 title('%Rejection1 as a Function of Time')
150 legend('Model',1)
151
152 subplot(2,3,4)
153 plot([A]/60,[O1],'-g')
154 xlabel('Time (hr)')
155 ylabel('%Rejection2')
156 title('%Rejection2 as a Function of Time')
157 legend('Model',1)
158
159 subplot(2,3,5)
160 plot([A]/60,[K5],'-g')
161 xlabel('Time (hr)')
162 ylabel('Cfa/Cf0')
163 title('Cfa/Cf0 as a Function of Time')
164 legend('Model',1)
165
166 subplot(2,3,6)
167 plot([A]/60,[K6],'-g')
168 xlabel('Time (hr)')
169 ylabel('Cfa1/Cf0')
170 title('Cfa/Cf01 as a Function of Time')
171 legend('Model',1)
172
173 fprintf('\n      \n')
174 fprintf('\n      \n')
175 fprintf('\n CONCLUSION    \n')
176 fprintf('\n       R2_W           MSE_W                 R2_pH           MSE_pH\n');
177 fprintf('\n%11.3f%20.4e%19.4f%19.4e%\n',R2_W,MSE_W,R2_pH,MSE_pH)
178 fprintf('\n      \n')
179
180 fprintf('\n    Time(min)       J_w(L/dm2/min)          pH            Cf,a(mole/l) ↙
W(kg)        Cf,a1(mole/l)\n');
181 for x=1:1:200000000;
182 z=x*5000;
183 fprintf('\n%11.3f%20.4e%19.4f%19.4e%19.4e%19.4e%\n',T(z),j(z),p(z),c(z),d(z),e(z));
184
185 end
186
187
```

```matlab
1
2 function fu=Acetic10_Butyric10(z,t)
3 %Independent Variables
4 V_f_0=1;C_d_s_0=1;V_d_0=0.5;
5 %Fixed Variable
6 T=301;f=1.58;
7 % T=300..normal f=1.58...normal
8 L=0.9207;D=0.023;W=0.4572;A_m=42*10^(-2);d_h=0.0438;
9 % Acid Variables
10 %
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
11 C_f_a_0=10*10^(-3);
12 C_f_a1_0=10*10^(-3);
13 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
14 B_a=2.102/60/100;%B of Acetic Acid
15 B_a1=0.639/60/100;% B of Butyric Acid
16 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
17 D_A_298=6.534*10^(-6);D_HA_298=7.5429*10^(-6);%Acetic Physicocmical Property
18 D_A1_298=5.208*10^(-6);D_HA1_298=5.429*10^(-6);%Butyric Physicocmical Property
19 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
20 K_a_T=10^-(1170.48/T-3.1649+0.013399*T);%Acetic acid
21 K_a1_T=10^-(1033.39/T-2.6215+0.013334*T);%Butyric Acid
22 %
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
^^^^
23 %Constant Variable
24 B_s=0.460/60/100;
25 A=0.413/60/100;
26 o_s=0.897;%amonium chloride coefficient
27 S=491*10^-5;
28 D_NH4_298=117.42e-7;D_Cl_298=121.92e-7;%amonium & chloride diffusion coefficient
29 den_w=999.65+2.0438/10*(T-273)-6.174*10^-2*(T-273)^1.5;
30 D_H_298=5.587*10^(-5);
31 visco_298=((298-273)+246)/((0.05594*(298-273)+5.2842)*(298-273)+137.37);
32 visco_T=((T-273)+246)/((0.05594*(T-273)+5.2842)*(T-273)+137.37);
33 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
34 o_a=1;n_s=2;R=0.08314;o_a1=1;
35 %Temperature Correction
36 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
37 K_CO2_T=10^-(3404.71/T-14.8435+0.032786*T);
38 K_HCO3_T=10^-(2902.39/T-6.4980+0.02379*T);
39 K_w_T=10^(-4470.99/T+6.0875-0.017060*T);
40 K_NH4CL=10^-(2835.76/T-0.6322+0.001225*T);%Amonium Chloride
41 E=2727.586+0.6224107*T-466.9151*log(T)-52000.87/T;
42 d=1-(((T-273)-3.9863)^2*((T-273)+288.9414))/(508929.2*((T-273)+68.12963))...
43 +0.011445*exp((-374.3)/(T-273));
44 A_I=(1.82483*10^6*d^0.5)/(E*T)^1.5;
45 K=1/K_a_T;
46 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
47 K1=1/K_a1_T;
48 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
49 D_H_T=D_H_298*T/298*visco_298/visco_T ;
50 D_A_T=D_A_298*T/298*visco_298/visco_T;
51 D_HA_T=D_HA_298*T/298*visco_298/visco_T;
52 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
53 D_A1_T=D_A1_298*T/298*visco_298/visco_T;
```

```matlab
54 D_HA1_T=D_HA1_298*T/298*visco_298/visco_T;
55 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
56 D_NH4_T=D_NH4_298*T/298*visco_298/visco_T;
57 D_Cl_T=D_Cl_298*T/298*visco_298/visco_T;
58 D_1_T=2/(1/D_H_T +1/D_A_T);
59 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
60 D_11_T=2/(1/D_H_T +1/D_A1_T);
61 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
62 D_S_T=2/(1/D_NH4_T +1/D_Cl_T);
63 D_d_s=D_S_T;
64 D_f_s=D_S_T;
65 K_H=0.034*exp(-2400*(1/T-1/298));%******************
66 C_f_co2=10^-3.408*K_H;%****************************
67 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
68 %Unknown Variable
69 J_w=z(1);P_i=z(2);P_m=z(3);J_s=z(4);C_i_s=z(5);
70 C_mf_s=z(6);J_a=z(7);C_mf_a=z(8);C_i_a=z(9);C_d_a=z(10);
71 C_d_H=z(11);gamma_d_H=z(12);gamma_d_A=z(13);I_i=z(14);C_d_s=z(15);
72 C_mf_H=z(16);gamma_mf_H=z(17);gamma_mf_A=z(18);I_mf=z(19);C_md_s=z(20);
73 k_d=z(21);Sh_d=z(22);Sc_d=z(23);Re_d=z(24);
74 v_d=z(25);den_d=z(26);mul_d=z(27);f_2=z(28);D_f_a=z(29);
75 k_f_a=z(30);C_f_a=z(31);Sh_f_a=z(32);Re_f=z(33);v_f=z(34);
76 f_1=z(35);Sc_f_a=z(36);k_f_s=z(37);C_f_s=z(38);
77 C_f_ai=z(39);C_f_si=z(40);C_d_ai=z(41);C_d_si=z(42);
78 gamma_f_H=z(43);C_f_A=z(44);gamma_f_A=z(45);I_f=z(46);C_f_NH3=z(47);
79 Sh_f_s=z(48);Sc_f_s=z(49);w=z(50);C_f_H=z(51);gamma_f_HCO3=z(52);
80 gamma_f_OH=z(53);C_f_HCO3=z(54);C_f_CO3=z(55);
81 gamma_f_CO3=z(56);C_f_NH4=z(57);C_f_OH=z(58);C_f_HA=z(59);den_f=z(60);
82 pH_f=z(61);
83 % for charge balance of NH4
84 gamma_d_OH=z(62);gamma_i_NH4=z(63);C_d_A=z(64);C_d_OH=z(65);C_i_NH4=z(66);
85 C_i_NH3=z(67);C_d_HA=z(68);C_mf_A=z(69);C_mf_OH=z(70);C_mf_NH4=z(71);
86 gamma_mf_OH=z(72);gamma_mf_NH4=z(73);C_mf_NH3=z(74);C_mf_HA=z(75);gamma_f_NH4=z(76);
87 mul_f=z(77);
88 % for two acid
89 C_mf_a1=z(78);C_i_a1=z(79);J_a1=z(80);C_d_a1=z(81);C_d_A1=z(82);gamma_d_A1=z(83);
90 C_d_HA1=z(84);C_mf_A1=z(85);gamma_mf_A1=z(86);C_mf_HA1=z(87);C_f_a1=z(88);D_f_a1=z↙
(89);
91 k_f_a1=z(90);Sh_f_a1=z(91);Sc_f_a1=z(92);C_f_ai1=z(93);C_d_ai1=z(94);C_f_A1=z(95);
92 gamma_f_A1=z(96);C_f_HA1=z(97);C_f_HCO3e=z(98);C_f_CO3e=z(99);
93 format short e
94 %Math Model
95 fu(1)=A*(P_i-P_m )-J_w;
96 fu(2)= B_s *(C_i_s-C_mf_s)-J_s;
97 fu(3)= B_a*(C_mf_a-C_i_a)-J_a;
98 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
99 fu(4)= B_a1*(C_mf_a1-C_i_a1)-J_a1;         %Ba1,C_mf_a1,C_i_a1,J_a1
100 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
101 fu(5)=C_d_a-C_i_a;
102 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
103 fu(6)=C_d_a1-C_i_a1;                       %C_d_a1
104 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
105 fu(7)=J_a/J_w-C_d_a;
106 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
107 fu(8)=J_a1/J_w-C_d_a1;
108 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
```

```
109 fu(9)= o_s*n_s*C_i_s*R*T+o_a*(C_d_a+C_d_H)*R*T+o_a1*(C_d_a1+C_d_H)*R*T-P_i;
110 %Charge Balance Equation at interface %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
111 fu(10)=C_d_A+C_d_A1+C_d_OH+C_i_s-C_i_NH4-C_d_H;              %C_d_A1
112 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
113 fu(11)=10^(-(A_I*(I_i^(1/2)/(1+I_i^(1/2))- 0.3*I_i)))-gamma_d_H;
114 fu(12)=10^(-(A_I*(I_i^(1/2)/(1+I_i^(1/2))- 0.3*I_i)))-gamma_d_A;
115 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
116 fu(13)=10^(-(A_I*(I_i^(1/2)/(1+I_i^(1/2))- 0.3*I_i)))-gamma_d_A1;       %gamma_d_A1
117 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
118 fu(14)=10^(-(A_I*(I_i^(1/2)/(1+I_i^(1/2))- 0.3*I_i)))-gamma_d_OH;
119 fu(15)=10^(-(A_I*(I_i^(1/2)/(1+I_i^(1/2))- 0.3*I_i)))-gamma_i_NH4;
120 fu(16)=0.5*(C_i_s+C_i_NH4+C_d_H+C_d_A+C_d_A1+C_d_OH)- I_i;
121 fu(17)=K_a_T*C_d_HA/gamma_d_H/gamma_d_A/C_d_H-C_d_A;
122 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
123 fu(18)=K_a1_T*C_d_HA1/gamma_d_H/gamma_d_A1/C_d_H-C_d_A1;   %C_d_HA1
124 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
125 fu(19)=K_w_T/C_d_H/gamma_d_OH/gamma_d_H-C_d_OH;
126 fu(20)=C_i_NH4+C_i_NH3-C_i_s;
127 fu(21)=C_d_HA+C_d_A-C_d_a;
128 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
129 fu(22)=C_d_HA1+C_d_A1-C_d_a1;
130 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
131 fu(23)=gamma_d_H*C_d_H*C_i_NH3/gamma_i_NH4/K_NH4CL-C_i_NH4;
132 fu(24)=o_s*n_s*C_mf_s*R*T+o_a*(C_mf_a+C_mf_H)*R*T+o_a1*(C_mf_a1+C_mf_H)*R*T-P_m;
133 %Charge Balance Equation at membrane active layer %%%%%%%%%%%%%%%%%%%%%%%%%%%
134 fu(25)=C_mf_A+C_mf_A1+C_mf_OH+C_mf_s-C_mf_NH4-C_mf_H;              %C_mf_A1
135 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
136 fu(26)=10^(-(A_I*(I_mf^(1/2)/(1+I_mf^(1/2))- 0.3*I_mf)))-gamma_mf_H;
137 fu(27)=10^(-(A_I*(I_mf^(1/2)/(1+I_mf^(1/2))- 0.3*I_mf)))-gamma_mf_A;
138 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
139 fu(28)=10^(-(A_I*(I_mf^(1/2)/(1+I_mf^(1/2))- 0.3*I_mf)))-gamma_mf_A1;        %↙
gamma_mf_A1
140 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
141 fu(29)=10^(-(A_I*(I_mf^(1/2)/(1+I_mf^(1/2))- 0.3*I_mf)))-gamma_mf_OH;
142 fu(30)=10^(-(A_I*(I_mf^(1/2)/(1+I_mf^(1/2))- 0.3*I_mf)))-gamma_mf_NH4;
143 fu(31)=0.5*(C_mf_s+C_mf_NH4+C_mf_H+C_mf_A+C_mf_OH+C_mf_A1)-I_mf;
144 fu(32)=K_a_T*C_mf_HA/gamma_mf_H/gamma_mf_A/C_mf_H-C_mf_A;
145 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
146 fu(33)=K_a1_T*C_mf_HA1/gamma_mf_H/gamma_mf_A1/C_mf_H-C_mf_A1;
147 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
148 fu(34)=K_w_T/C_mf_H/gamma_mf_OH/gamma_mf_H-C_mf_OH;
149 fu(35)=C_mf_NH4+C_mf_NH3-C_mf_s;
150 fu(36)=C_mf_HA+C_mf_A-C_mf_a;
151 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
152 fu(37)=C_mf_HA1+C_mf_A1-C_mf_a1;     %C_mf_HA1
153 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
154 fu(38)=gamma_mf_H*C_mf_H*C_mf_NH3/gamma_mf_NH4/K_NH4CL-C_mf_NH4;
155 fu(39)=(C_md_s-C_i_s)/2*D_S_T*(1/(J_w*(0.789*C_i_s+0.211*C_md_s)+J_s)...
156 +1/(J_w*(0.211*C_i_s+0.789*C_md_s)+ J_s))-S;
157 fu(40)=(C_d_s-C_md_s)/2*D_S_T*(1/(J_w*(0.789*C_md_s+0.211*C_d_s)+J_s)...
158 +1/(J_w*(0.211*C_md_s+0.789*C_d_s)+ J_s))-D_d_s/k_d;
159 fu(41)=Sh_d*D_d_s/d_h-k_d;
160 fu(42)=L*v_d*den_d/mul_d-Re_d;
161 fu(43)=(f_2+f)/2/W/D-v_d;
162 fu(44)=f+J_w*A_m-f_2;
163 % amonium chloride density and viscosity
```

```
164 fu(45)=6*exp((12.396*(C_d_s*53.491/den_d)^1.5039-1.7756)/(0.23471*(T-273)+1)/...
165 (-2.7591*(C_d_s*53.491/den_d)^2.8408+1))-mul_d;
166 fu(46)=den_w+0.2061e2*C_d_s-0.1577*C_d_s*(T-273)+1.553e-3*C_d_s*(T-273)^2-...
167 2.556*C_d_s^1.5+5.67e-2*C_d_s^1.5*(T-273)-5.082e-4*C_d_s^1.5*(T-273)^2-den_d;
168 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
169 fu(47)= 0.04*Re_d^(3/4)*Sc_d^(1/3)-Sh_d;
170 fu(48)= mul_d/den_d/D_d_s-Sc_d;
171 fu(49)=(0.5*(C_f_a-C_mf_a)*((D_1_T-D_HA_T)/...
172    (sqrt(1+4*K*(0.789*C_mf_a+0.211*C_f_a))*(J_w*(0.789*C_mf_a+0.211*C_f_a)-J_a))...
173    +(D_1_T-D_HA_T)/(sqrt(1+4*K*(0.211*C_mf_a+0.789*C_f_a))*(J_w*(0.211*C_mf_a...
174    +0.789*C_f_a)-J_a))+ D_HA_T/(J_w*(0.789*C_mf_a+0.211*C_f_a)-J_a)...
175    +D_HA_T/((J_w*(0.211*C_mf_a+0.789*C_f_a)-J_a))))+D_f_a/k_f_a;
176 %
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
177 fu(50)=(0.5*(C_f_a1-C_mf_a1)*((D_11_T-D_HA1_T)/...                          %
C_f_a1,D_f_a1,k_f_a1
178    (sqrt(1+4*K1*(0.789*C_mf_a1+0.211*C_f_a1))*(J_w*(0.789*C_mf_a1+0.211*C_f_a1)-
J_a1))...
179    +(D_11_T-D_HA1_T)/(sqrt(1+4*K1*(0.211*C_mf_a1+0.789*C_f_a1))*(J_w*(0.211
*C_mf_a1...
180    +0.789*C_f_a1)-J_a1))+ D_HA1_T/(J_w*(0.789*C_mf_a1+0.211*C_f_a1)-J_a1)...
181    +D_HA1_T/((J_w*(0.211*C_mf_a1+0.789*C_f_a1)-J_a1))))+D_f_a1/k_f_a1;
182 %
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
183 fu(51)=Sh_f_a*D_f_a/d_h-k_f_a;
184 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
185 fu(52)=Sh_f_a1*D_f_a1/d_h-k_f_a1;                  %Sh_f_a1
186 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
187 fu(53)=(D_1_T/(2*K*C_f_a)*(-1+sqrt(1+4*K*C_f_a))+D_HA_T/...
188    (4*K*C_f_a)*(-1+sqrt(1+4*K*C_f_a))^2)-D_f_a;
189 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
190 fu(54)=(D_11_T/(2*K1*C_f_a1)*(-1+sqrt(1+4*K1*C_f_a1))+D_HA1_T/...
191    (4*K1*C_f_a1)*(-1+sqrt(1+4*K1*C_f_a1))^2)-D_f_a1;
192 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
193 fu(55)=L*v_f*den_f/mul_f -Re_f;
194 fu(56)=(f+f_1)/W/D/2-v_f;
195 fu(57)=f-J_w*A_m-f_1;
196 fu(58)= 0.04*Re_f^(3/4)*Sc_f_a^(1/3)-Sh_f_a;
197 fu(59)= mul_f/den_f/D_f_a-Sc_f_a;
198 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
199 fu(60)= 0.04*Re_f^(3/4)*Sc_f_a1^(1/3)-Sh_f_a1;     %Sc_f_a1
200 fu(61)= mul_f/den_f/D_f_a1-Sc_f_a1;
201 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
202 fu(62)=(C_f_s-C_mf_s)/2*D_S_T*(1/(J_w*(0.789*C_mf_s+0.211*C_f_s)+J_s)...
203 +1/(J_w*(0.211*C_mf_s+0.789*C_f_s)+ J_s))+D_f_s/k_f_s;
204 fu(63)=Sh_f_s*D_f_s/d_h-k_f_s;
205 %Mole balance Equation
206 fu(64)=(V_f_0*C_f_a_0+C_f_ai*f_1*t)/(V_f_0+f_1*t)-C_f_a;
207 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
208 fu(65)=(V_f_0*C_f_a1_0+C_f_ai1*f_1*t)/(V_f_0+f_1*t)-C_f_a1;
209 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
210 fu(66)=C_f_s*f_1*t/(V_f_0+f_1*t)-C_f_si;
211 fu(67)=(C_f_a*f-J_a*A_m)/f_1-C_f_ai;
212 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
213 fu(68)=(C_f_a1*f-J_a1*A_m)/f_1-C_f_ai1;
214 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
```

```
215 fu(69)=(C_f_si*f+J_s*A_m)/f_1-C_f_s;
216 fu(70)=(C_d_s_0*V_d_0+C_d_si*f_2*t)/(V_d_0+f_2*t)-C_d_s;
217 fu(71)=(C_d_ai*f*t+C_d_ai*(V_d_0+f_2*t-f*t))/(f_2*t)-C_d_a;
218 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
219 fu(72)=(C_d_ai1*f*t+C_d_ai1*(V_d_0+f_2*t-f*t))/(f_2*t)-C_d_a1;
220 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
221 fu(73)=(C_d_s*f-J_s*A_m)/f_2-C_d_si;
222 fu(74)=(C_d_a*f_2-J_a*A_m)/f-C_d_ai;
223 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
224 fu(75)=(C_d_a1*f_2-J_a1*A_m)/f-C_d_ai1;
225 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
226 %additional Equation
227 fu(76)= 0.04*Re_f^(3/4)*Sc_f_s^(1/3)-Sh_f_s;
228 fu(77)= mul_f/den_f/D_f_s-Sc_f_s;
229 fu(78)=J_w*A_m*t*den_d/1000-w;
230 %ionic strenght in feed solution%%%%%%%%%%%%%%%%%%%
231 fu(79)=0.5*(C_f_si+C_f_NH4+C_f_H+C_f_OH+C_f_A+C_f_A1+C_f_HCO3+4*C_f_CO3)- I_f;%C_f_A1
232 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
233 fu(80)=10^(-(A_I*(I_f^(1/2)/(1+I_f^(1/2))- 0.3*I_f)))-gamma_f_H;%Davis
234 fu(81)=10^(-(A_I*(I_f^(1/2)/(1+I_f^(1/2))- 0.3*I_f)))-gamma_f_A;%Davis
235 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
236 fu(82)=10^(-(A_I*(I_f^(1/2)/(1+I_f^(1/2))- 0.3*I_f)))-gamma_f_A1;%Davis        %↙
gamma_f_A1
237 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
238 fu(83)=10^(-(A_I*(I_f^(1/2)/(1+I_f^(1/2))- 0.3*I_f)))-gamma_f_HCO3;%Davis
239 fu(84)=10^(-(A_I*(I_f^(1/2)/(1+I_f^(1/2))- 0.3*I_f)))-gamma_f_OH;%Davis
240 %Charge Balance Equation %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
241 fu(85)=C_f_HCO3+2*C_f_CO3+C_f_A+C_f_A1+C_f_OH+C_f_si-C_f_NH4-C_f_H;
242 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
243 %New Equation for charge balance
244 fu(86)=K_a_T*C_f_HA/gamma_f_H/gamma_f_A/C_f_H-C_f_A;
245 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
246 fu(87)=K_a1_T*C_f_HA1/gamma_f_H/gamma_f_A1/C_f_H-C_f_A1;       %C_f_HA1
247 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
248 fu(88)=K_CO2_T/gamma_f_H/gamma_f_HCO3/C_f_H*C_f_HCO3e-C_f_HCO3;%******
249 fu(89)=K_HCO3_T*gamma_f_HCO3/gamma_f_H/gamma_f_CO3/C_f_H*C_f_CO3e-C_f_CO3;%******
250 fu(90)=10^(-(A_I*4*(I_f^(1/2)/(1+I_f^(1/2))- 0.3*I_f)))-gamma_f_CO3;%Davis
251 fu(91)=K_w_T/C_f_H/gamma_f_OH/gamma_f_H-C_f_OH;
252 fu(92)=C_f_NH4+C_f_NH3-C_f_si;
253 fu(93)=C_f_HA+C_f_A-C_f_a;
254 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
255 fu(94)=C_f_HA1+C_f_A1-C_f_a1; %C_f_HA1
256 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
257 fu(95)=gamma_f_H*C_f_H*C_f_NH3/gamma_f_NH4/K_NH4CL-C_f_NH4;
258 fu(96)=10^(-(A_I*(I_f^(1/2)/(1+I_f^(1/2))- 0.3*I_f)))-gamma_f_NH4;
259 fu(97)=pH_f+log10(gamma_f_H*C_f_H);
260 fu(98)=C_f_co2-C_f_HCO3-C_f_HCO3e;
261 fu(99)=C_f_HCO3-C_f_CO3-C_f_CO3e;
262 %protron condition
263 %fu(51)=C_f_HCO3+2*C_f_CO3+C_f_Ac+C_f_OH+C_f_NH3-C_f_H;
264
265
266
```

G.8 A mixture of 10 mM acetic acid and 10 mM valeric acid as feed solution and NH$_4$Cl as draw solution

```
 1 clc
 2 clear
 3 [g]=[1.596e-03    2.3736e+01    5.483e-01    4.051e-05  5.287e-01    2.346e-04↙
3.840e-11   1.013e-02   1.962e-08...
 4 1.837e-08      1.9184e-05    7.340e-01    7.340e-01 5.287e-01    9.998e-01    5.845↙
e-04    9.680e-01    9.680e-01...
 5 8.191e-04      9.9110e-01    1.880e-01    6.372e+02 3.700e+02    2.894e+04    1.503↙
e+02    1.012e+03    4.839e+00...
 6 1.581e+00      1.1682e-05    1.723e-01    1.000e-02 6.459e+02    2.759e+04    1.502↙
e+02    1.579e+00    4.291e+02...
 7 1.843e-01      1.3668e-05    1.001e-02    2.893e-06 7.723e-09    9.993e-01    9.729↙
e-01    3.106e-04    9.729e-01...
 8 5.800e-04      3.4869e-12    6.245e+02    3.879e+02 1.154e-04    5.771e-04    9.729↙
e-01    9.729e-01    1.214e-08...
 9 5.354e-15      8.9579e-01    2.893e-06    2.298e-11 9.691e-03    9.962e+02    3.251↙
e+00    7.340e-01    7.340e-01...
10 1.155e-08      1.2134e-09    5.287e-01    1.917e-05 6.820e-09    3.138e-04    2.407↙
e-11    2.346e-04    9.680e-01...
11 9.680e-01      2.7921e-10    9.820e-03    9.729e-01 4.994e+00    1.018e-02    5.017↙
e-09    1.059e-11    4.901e-09...
12 2.898e-09      7.3399e-01    2.003e-09    2.707e-04 9.680e-01    9.914e-03    1.000↙
e-02    1.169e-05    1.724e-01...
13 6.457e+02      4.2873e+02    1.001e-02    2.060e-09 2.665e-04    9.729e-01    9.735↙
e-03    1.439e-05    1.214e-08];
14 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
15 ts=0.17:0.0006:1800;
16 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
17 for ti=1:numel(ts);
18 t=ts(ti);
19 options = optimset('Algorithm','Levenberg-Marquardt','TolFun',1*10^-15,'TolX',1*10^-↙
15 ...
20     ,'Maxiter',25,'Maxfun',15000);
21 [r] = fsolve(@(z)Acetic10_Valeric10(z,t),[g],options)
22 j(ti)=r(1,1);%Jw
23 c(ti)=r(1,31);%Cfa
24 d(ti)=r(1,50);%W
25 p(ti)=r(1,61);%pH
26 T(ti)=t;
27 e(ti)=r(1,88);%Cfa1
28 % For Rejection
29 O(ti)=(1-r(1,10)/r(1,31))*100;%Rej
30 O1(ti)=(1-r(1,81)/r(1,88))*100;%Rej1
31 %For Concentration Performance
32 K4(ti)=r(1,31)/0.01;%Cfa/Cf0
33 K6(ti)=r(1,88)/0.01;%Cfa1/Cf0
34 K1(ti)=r(1,1);
35 [g]=[r];
36 %for Plotting
37 [A(ti)]=t;
38 [B(ti)]=d(ti);
39 [C(ti)]=p(ti);
40 % For Rejection
41 [M(ti)]=O(ti);
42 [N(ti)]=K1(ti);
43 [K5(ti)]=K4(ti); %Cfa/Cf0
44 disp(ti)
```

```
45 fprintf('\n      Time(min)        J_w(L/dm2/min)            pH            Cf,a(mole/l)↙
W(kg)             Cf,a1(mole/l)\n');
46
47 fprintf('\n%11.4f%20.4e%19.4f%19.4e%19.4e%19.4e\n' ,T(ti),j(ti),p(ti),c(ti),d(ti),e(
(ti));
48
49 format short e
50
51 %Result of Best guess
52 fprintf('\n%8.3e%15.4e%13.3e%14.3e%11.3e%13.3e%13.3e%13.3e%13.3e%\n'...
53      ,r(1,1),r(1,2),r(1,3),r(1,4),r(1,5),r(1,6),r(1,7),r(1,8),r(1,9));
54
55  fprintf('\n%8.3e%15.4e%13.3e%14.3e%11.3e%13.3e%13.3e%13.3e%13.3e%\n'...
56     ,r(1,10),r(1,11),r(1,12),r(1,13),r(1,14),r(1,15),r(1,16),r(1,17),r(1,18));
57
58 fprintf('\n%8.3e%15.4e%13.3e%14.3e%11.3e%13.3e%13.3e%13.3e%13.3e%\n'...
59      ,r(1,19),r(1,20),r(1,21),r(1,22),r(1,23),r(1,24),r(1,25),r(1,26),r(1,27));
60
61 fprintf('\n%8.3e%15.4e%13.3e%14.3e%11.3e%13.3e%13.3e%13.3e%13.3e%\n'...
62    ,r(1,28),r(1,29),r(1,30),r(1,31),r(1,32),r(1,33),r(1,34),r(1,35),r(1,36));
63
64 fprintf('\n%8.3e%15.4e%13.3e%14.3e%11.3e%13.3e%13.3e%13.3e%13.3e%13.3e%13.3e\n'...
65      ,r(1,37),r(1,38),r(1,39),r(1,40),r(1,41),r(1,42),r(1,43),r(1,44),r(1,45));
66
67 fprintf('\n%8.3e%15.4e%13.3e%14.3e%11.3e%13.3e%13.3e%13.3e%13.3e%\n'...
68     ,r(1,46),r(1,47),r(1,48),r(1,49),r(1,50),r(1,51),r(1,52),r(1,53),r(1,54));
69
70 fprintf('\n%8.3e%15.4e%13.3e%14.3e%11.3e%13.3e%13.3e%13.3e%13.3e%\n'...
71      ,r(1,55),r(1,56),r(1,57),r(1,58),r(1,59),r(1,60),r(1,61),r(1,62),r(1,63));
72
73 fprintf('\n%8.3e%15.4e%13.3e%14.3e%11.3e%13.3e%13.3e%13.3e%13.3e%\n'...
74     ,r(1,64),r(1,65),r(1,66),r(1,67),r(1,68),r(1,69),r(1,70),r(1,71),r(1,72));
75
76 fprintf('\n%8.3e%15.4e%13.3e%14.3e%11.3e%13.3e%13.3e%13.3e%13.3e%\n'...
77     ,r(1,73),r(1,74),r(1,75),r(1,76),r(1,77),r(1,78),r(1,79),r(1,80),r(1,81));
78
79 fprintf('\n%8.3e%15.4e%13.3e%14.3e%11.3e%13.3e%13.3e%13.3e%13.3e%\n'...
80     ,r(1,82),r(1,83),r(1,84),r(1,85),r(1,86),r(1,87),r(1,88),r(1,89),r(1,90));
81
82 fprintf('\n%8.3e%15.4e%13.3e%14.3e%11.3e%13.3e%13.3e%13.3e%13.3e%\n'...
83     ,r(1,91),r(1,92),r(1,93),r(1,94),r(1,95),r(1,96),r(1,97),r(1,98),r(1,99));
84 end
85
86 %Time Interval%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
87 Ai=0:30:1800;
88 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
89 format short
90 Ci=spline(A,C,Ai);
91 format short e
92 Bi=spline(A,B,Ai);
93
94 %Experiment Value W and pH%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
95  F=[ 0    0.0160    0.0320    0.0470    0.0620    0.0770    0.0900    0.1040↙
0.1170    0.1310    0.1430    0.1560    0.1670...
96 0.1790    0.1910    0.2020    0.2140    0.2250    0.2360    0.2490    0.2600↙
0.2710    0.2810    0.2920    0.3030    0.3130...
```

```
 97 0.3240    0.3340    0.3440    0.3540    0.3630    0.3730    0.3820    0.3910↵
0.4000    0.4090    0.4180    0.4260    0.4340...
 98 0.4430    0.4510    0.4590    0.4730    0.4800    0.4880    0.4960    0.5030↵
0.5100    0.5170    0.5240    0.5310    0.5380...
 99 0.5440    0.5500    0.5570    0.5640    0.5700    0.5760    0.5830    0.5890↵
0.5980];
100
101  G=[ 3.2380    3.2300    3.2240    3.2210    3.2170    3.2140    3.2110    3.2080↵
3.2050    3.2020    3.2000    3.1980    3.1940...
102 3.1910    3.1890    3.1850    3.1830    3.1790    3.1780    3.1750    3.1720↵
3.1690    3.1670    3.1640    3.1620    3.1590...
103 3.1570    3.1550    3.1520    3.1500    3.1480    3.1450    3.1430    3.1410↵
3.1390    3.1360    3.1340    3.1320    3.1300...
104 3.1280    3.1260    3.1240    3.1220    3.1200    3.1190    3.1170    3.1150↵
3.1120    3.1110    3.1090    3.1080    3.1060...
105 3.1050    3.1030    3.1020    3.1010    3.0990    3.0980    3.0970    3.0960↵
3.0950];
106
107 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
108 F_m=mean(F);
109 G_m=mean(G);
110 sum1=0;sum2=0;
111 sum3=0;sum4=0;
112 %%% Number of Data %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
113 for k=1:1:61
114 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
115 sum1=(Bi(1,k)-F(1,k))^2+sum1;
116 sum2=(Bi(1,k)-F_m)^2+sum2;
117 sum3=(Ci(1,k)-G(1,k))^2+sum3;
118 sum4=(Ci(1,k)-G_m)^2+sum4;
119 end
120 format short
121 R2_W=1-sum1/sum2;
122 R2_pH=1-sum3/sum4;
123 MSE_W=sum1/k;
124 MSE_pH=sum3/k;
125 RMSE_W=sqrt(MSE_W)
126 RMSE_pH=sqrt(MSE_pH)
127 SEP_W=RMSE_W/F_m*100
128 SEP_pH=RMSE_pH/G_m*100
129
130 % Plotting
131 subplot(2,3,1)
132 plot([A]/60,[B],'-g',[Ai]/60,[F],'xr')
133 xlabel('Time (hr)')
134 ylabel('Weight (Kg)')
135 title('Weight Change as a Function of Time')
136 legend('Model','Experiment',2)
137
138 subplot(2,3,2)
139 plot([A]/60,[C],'-g',[Ai]/60,[G],'xr')
140 xlabel('Time (hr)')
141 ylabel('pH')
142 title('pH as a Function of Time')
143 legend('Model','Experiment',1)
144
```

```
145 subplot(2,3,3)
146 plot([A]/60,[M],'-g')
147 xlabel('Time (hr)')
148 ylabel('%Rejection1')
149 title('%Rejection1 as a Function of Time')
150 legend('Model',1)
151
152 subplot(2,3,4)
153 plot([A]/60,[O1],'-g')
154 xlabel('Time (hr)')
155 ylabel('%Rejection2')
156 title('%Rejection2 as a Function of Time')
157 legend('Model',1)
158
159 subplot(2,3,5)
160 plot([A]/60,[K5],'-g')
161 xlabel('Time (hr)')
162 ylabel('Cfa/Cf0')
163 title('Cfa/Cf0 as a Function of Time')
164 legend('Model',1)
165
166 subplot(2,3,6)
167 plot([A]/60,[K6],'-g')
168 xlabel('Time (hr)')
169 ylabel('Cfa1/Cf0')
170 title('Cfa/Cf01 as a Function of Time')
171 legend('Model',1)
172
173 fprintf('\n      \n')
174 fprintf('\n     \n')
175 fprintf('\n CONCLUSION     \n')
176 fprintf('\n       R2_W           MSE_W                   R2_pH          MSE_pH\n');
177 fprintf('\n%11.3f%20.4e%19.4f%19.4e%\n',R2_W,MSE_W,R2_pH,MSE_pH)
178 fprintf('\n      \n')
179
180 fprintf('\n    Time(min)       J_w(L/dm2/min)          pH            Cf,a(mole/l)↵
W(kg)          Cf,a1(mole/l)\n');
181 for x=1:1:200000000;
182 z=x*5000;
183 fprintf('\n%11.3f%20.4e%19.4f%19.4e%19.4e%19.4e%\n',T(z),j(z),p(z),c(z),d(z),e(z));
184
185 end
186
187
```

```matlab
1
2  function fu=Acetic10_Valeric10(z,t)
3  %Independent Variables
4  V_f_0=1;C_d_s_0=1;V_d_0=0.5;
5  %Fixed Variable
6  T=301;f=1.58;
7  % T=300..normal f=1.58...normal
8  L=0.9207;D=0.023;W=0.4572;A_m=42*10^(-2);d_h=0.0438;
9  % Acid Variables
10 %↙
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
11 C_f_a_0=10*10^(-3);
12 C_f_a1_0=10*10^(-3);
13 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
14 B_a=2.102/60/100;%New Value 2.102 L/m2/h/60/100 = L/dm2/min....
15 B_a1=0.492/60/100;% B of Valeric Acid, L/dm2/min Unit(0.57 L/m2/h)
16 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
17 D_A_298=6.534*10^(-6);D_HA_298=7.5429*10^(-6);%Acetic Physicocmical Property
18 D_Al_298=5.226*10^(-6);D_HA1_298=4.8022*10^(-6);%Valeric Physicocmical Property
19 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
20 K_a_T=10^-(1170.48/T-3.1649+0.013399*T);%Acetic acid
21 K_a1_T=10^-(921.38/T-1.8574+0.012105*T);%Valeric Acid
22 %↙
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^↙
^^^^
23 %Constant Variable
24 B_s=0.460/60/100;
25 A=0.413/60/100;
26 o_s=0.897;%amonium chloride coefficient
27 S=491*10^-5;
28 D_NH4_298=117.42e-7;D_Cl_298=121.92e-7;%amonium & chloride diffusion coefficient
29 den_w=999.65+2.0438/10*(T-273)-6.174*10^-2*(T-273)^1.5;
30 D_H_298=5.587*10^(-5);
31 visco_298=((298-273)+246)/((0.05594*(298-273)+5.2842)*(298-273)+137.37);
32 visco_T=((T-273)+246)/((0.05594*(T-273)+5.2842)*(T-273)+137.37);
33 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
34 o_a=1;n_s=2;R=0.08314;o_a1=1;
35 %Temperature Correction
36 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
37 K_CO2_T=10^-(3404.71/T-14.8435+0.032786*T);
38 K_HCO3_T=10^-(2902.39/T-6.4980+0.02379*T);
39 K_w_T=10^(-4470.99/T+6.0875-0.017060*T);
40 K_NH4CL=10^-(2835.76/T-0.6322+0.001225*T);%Amonium Chloride
41 E=2727.586+0.6224107*T-466.9151*log(T)-52000.87/T;
42 d=1-(((T-273)-3.9863)^2*((T-273)+288.9414))/(508929.2*((T-273)+68.12963))...
43 +0.011445*exp((-374.3)/(T-273));
44 A_I=(1.82483*10^6*d^0.5)/(E*T)^1.5;
45 K=1/K_a_T;
46 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
47 K1=1/K_a1_T;
48 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
49 D_H_T=D_H_298*T/298*visco_298/visco_T ;
50 D_A_T=D_A_298*T/298*visco_298/visco_T;
51 D_HA_T=D_HA_298*T/298*visco_298/visco_T;
52 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
53 D_A1_T=D_A1_298*T/298*visco_298/visco_T;
```

```
 54 D_HA1_T=D_HA1_298*T/298*visco_298/visco_T;
 55 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
 56 D_NH4_T=D_NH4_298*T/298*visco_298/visco_T;
 57 D_Cl_T=D_Cl_298*T/298*visco_298/visco_T;
 58 D_1_T=2/(1/D_H_T +1/D_A_T);
 59 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
 60 D_11_T=2/(1/D_H_T +1/D_A1_T);
 61 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
 62 D_S_T=2/(1/D_NH4_T +1/D_Cl_T);
 63 D_d_s=D_S_T;
 64 D_f_s=D_S_T;
 65 K_H=0.034*exp(-2400*(1/T-1/298));%*****************
 66 C_f_co2=10^-3.408*K_H;%***************************
 67 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 68 %Unknown Variable
 69 J_w=z(1);P_i=z(2);P_m=z(3);J_s=z(4);C_i_s=z(5);
 70 C_mf_s=z(6);J_a=z(7);C_mf_a=z(8);C_i_a=z(9);C_d_a=z(10);
 71 C_d_H=z(11);gamma_d_H=z(12);gamma_d_A=z(13);I_i=z(14);C_d_s=z(15);
 72 C_mf_H=z(16);gamma_mf_H=z(17);gamma_mf_A=z(18);I_mf=z(19);C_md_s=z(20);
 73 k_d=z(21);Sh_d=z(22);Sc_d=z(23);Re_d=z(24);
 74 v_d=z(25);den_d=z(26);mul_d=z(27);f_2=z(28);D_f_a=z(29);
 75 k_f_a=z(30);C_f_a=z(31);Sh_f_a=z(32);Re_f=z(33);v_f=z(34);
 76 f_1=z(35);Sc_f_a=z(36);k_f_s=z(37);C_f_s=z(38);
 77 C_f_ai=z(39);C_f_si=z(40);C_d_ai=z(41);C_d_si=z(42);
 78 gamma_f_H=z(43);C_f_A=z(44);gamma_f_A=z(45);I_f=z(46);C_f_NH3=z(47);
 79 Sh_f_s=z(48);Sc_f_s=z(49);w=z(50);C_f_H=z(51);gamma_f_HCO3=z(52);
 80 gamma_f_OH=z(53);C_f_HCO3=z(54);C_f_CO3=z(55);
 81 gamma_f_CO3=z(56);C_f_NH4=z(57);C_f_OH=z(58);C_f_HA=z(59);den_f=z(60);
 82 pH_f=z(61);
 83 % for charge balance of NH4
 84 gamma_d_OH=z(62);gamma_i_NH4=z(63);C_d_A=z(64);C_d_OH=z(65);C_i_NH4=z(66);
 85 C_i_NH3=z(67);C_d_HA=z(68);C_mf_A=z(69);C_mf_OH=z(70);C_mf_NH4=z(71);
 86 gamma_mf_OH=z(72);gamma_mf_NH4=z(73);C_mf_NH3=z(74);C_mf_HA=z(75);gamma_f_NH4=z(76);
 87 mul_f=z(77);
 88 % for two acid
 89 C_mf_a1=z(78);C_i_a1=z(79);J_a1=z(80);C_d_a1=z(81);C_d_A1=z(82);gamma_d_A1=z(83);
 90 C_d_HA1=z(84);C_mf_A1=z(85);gamma_mf_A1=z(86);C_mf_HA1=z(87);C_f_a1=z(88);D_f_a1=z↙
(89);
 91 k_f_a1=z(90);Sh_f_a1=z(91);Sc_f_a1=z(92);C_f_ai1=z(93);C_d_ai1=z(94);C_f_A1=z(95);
 92 gamma_f_A1=z(96);C_f_HA1=z(97);C_f_HCO3e=z(98);C_f_CO3e=z(99);
 93 format short e
 94 %Math Model
 95 fu(1)=A*(P_i-P_m )-J_w;
 96 fu(2)= B_s *(C_i_s-C_mf_s)-J_s;
 97 fu(3)= B_a*(C_mf_a-C_i_a)-J_a;
 98 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
 99 fu(4)= B_a1*(C_mf_a1-C_i_a1)-J_a1;        %Ba1,C_mf_a1,C_i_a1,J_a1
100 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
101 fu(5)=C_d_a-C_i_a;
102 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
103 fu(6)=C_d_a1-C_i_a1;                      %C_d_a1
104 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
105 fu(7)=J_a/J_w-C_d_a;
106 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
107 fu(8)=J_a1/J_w-C_d_a1;
108 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
```

```matlab
109 fu(9)= o_s*n_s*C_i_s*R*T+o_a*(C_d_a+C_d_H)*R*T+o_a1*(C_d_a1+C_d_H)*R*T-P_i;
110 %Charge Balance Equation at interface %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
111 fu(10)=C_d_A+C_d_A1+C_d_OH+C_i_s-C_i_NH4-C_d_H;                %C_d_A1
112 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
113 fu(11)=10^(-(A_I*(I_i^(1/2)/(1+I_i^(1/2))- 0.3*I_i)))-gamma_d_H;
114 fu(12)=10^(-(A_I*(I_i^(1/2)/(1+I_i^(1/2))- 0.3*I_i)))-gamma_d_A;
115 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
116 fu(13)=10^(-(A_I*(I_i^(1/2)/(1+I_i^(1/2))- 0.3*I_i)))-gamma_d_A1;        %gamma_d_A1
117 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
118 fu(14)=10^(-(A_I*(I_i^(1/2)/(1+I_i^(1/2))- 0.3*I_i)))-gamma_d_OH;
119 fu(15)=10^(-(A_I*(I_i^(1/2)/(1+I_i^(1/2))- 0.3*I_i)))-gamma_i_NH4;
120 fu(16)=0.5*(C_i_s+C_i_NH4+C_d_H+C_d_A+C_d_A1+C_d_OH)- I_i;
121 fu(17)=K_a_T*C_d_HA/gamma_d_H/gamma_d_A/C_d_H-C_d_A;
122 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
123 fu(18)=K_a1_T*C_d_HA1/gamma_d_H/gamma_d_A1/C_d_H-C_d_A1;   %C_d_HA1
124 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
125 fu(19)=K_w_T/C_d_H/gamma_d_OH/gamma_d_H-C_d_OH;
126 fu(20)=C_i_NH4+C_i_NH3-C_i_s;
127 fu(21)=C_d_HA+C_d_A-C_d_a;
128 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
129 fu(22)=C_d_HA1+C_d_A1-C_d_a1;
130 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
131 fu(23)=gamma_d_H*C_d_H*C_i_NH3/gamma_i_NH4/K_NH4CL-C_i_NH4;
132 fu(24)=o_s*n_s*C_mf_s*R*T+o_a*(C_mf_a+C_mf_H)*R*T+o_a1*(C_mf_a1+C_mf_H)*R*T-P_m;
133 %Charge Balance Equation at membrane active layer %%%%%%%%%%%%%%%%%%%%%%%%%%%
134 fu(25)=C_mf_A+C_mf_A1+C_mf_OH+C_mf_s-C_mf_NH4-C_mf_H;               %C_mf_A1
135 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
136 fu(26)=10^(-(A_I*(I_mf^(1/2)/(1+I_mf^(1/2))- 0.3*I_mf)))-gamma_mf_H;
137 fu(27)=10^(-(A_I*(I_mf^(1/2)/(1+I_mf^(1/2))- 0.3*I_mf)))-gamma_mf_A;
138 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
139 fu(28)=10^(-(A_I*(I_mf^(1/2)/(1+I_mf^(1/2))- 0.3*I_mf)))-gamma_mf_A1;        %↙
gamma_mf_A1
140 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
141 fu(29)=10^(-(A_I*(I_mf^(1/2)/(1+I_mf^(1/2))- 0.3*I_mf)))-gamma_mf_OH;
142 fu(30)=10^(-(A_I*(I_mf^(1/2)/(1+I_mf^(1/2))- 0.3*I_mf)))-gamma_mf_NH4;
143 fu(31)=0.5*(C_mf_s+C_mf_NH4+C_mf_H+C_mf_A+C_mf_OH+C_mf_A1)-I_mf;
144 fu(32)=K_a_T*C_mf_HA/gamma_mf_H/gamma_mf_A/C_mf_H-C_mf_A;
145 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
146 fu(33)=K_a1_T*C_mf_HA1/gamma_mf_H/gamma_mf_A1/C_mf_H-C_mf_A1;
147 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
148 fu(34)=K_w_T/C_mf_H/gamma_mf_OH/gamma_mf_H-C_mf_OH;
149 fu(35)=C_mf_NH4+C_mf_NH3-C_mf_s;
150 fu(36)=C_mf_HA+C_mf_A-C_mf_a;
151 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
152 fu(37)=C_mf_HA1+C_mf_A1-C_mf_a1;     %C_mf_HA1
153 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
154 fu(38)=gamma_mf_H*C_mf_H*C_mf_NH3/gamma_mf_NH4/K_NH4CL-C_mf_NH4;
155 fu(39)=(C_md_s-C_i_s)/2*D_S_T*(1/(J_w*(0.789*C_i_s+0.211*C_md_s)+J_s)...
156 +1/(J_w*(0.211*C_i_s+0.789*C_md_s)+ J_s))-S;
157 fu(40)=(C_d_s-C_md_s)/2*D_S_T*(1/(J_w*(0.789*C_md_s+0.211*C_d_s)+J_s)...
158 +1/(J_w*(0.211*C_md_s+0.789*C_d_s)+ J_s))-D_d_s/k_d;
159 fu(41)=Sh_d*D_d_s/d_h-k_d;
160 fu(42)=L*v_d*den_d/mul_d-Re_d;
161 fu(43)=(f_2+f)/2/W/D-v_d;
162 fu(44)=f+J_w*A_m-f_2;
163 % amonium chloride density and viscosity
```

```
164 fu(45)=6*exp((12.396*(C_d_s*53.491/den_d)^1.5039-1.7756)/(0.23471*(T-273)+1)/...
165 (-2.7591*(C_d_s*53.491/den_d)^2.8408+1))-mul_d;
166 fu(46)=den_w+0.2061e2*C_d_s-0.1577*C_d_s*(T-273)+1.553e-3*C_d_s*(T-273)^2-...
167 2.556*C_d_s^1.5+5.67e-2*C_d_s^1.5*(T-273)-5.082e-4*C_d_s^1.5*(T-273)^2-den_d;
168 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
169 fu(47)= 0.04*Re_d^(3/4)*Sc_d^(1/3)-Sh_d;
170 fu(48)= mul_d/den_d/D_d_s-Sc_d;
171 fu(49)=(0.5*(C_f_a-C_mf_a)*((D_1_T-D_HA_T)/...
172     (sqrt(1+4*K*(0.789*C_mf_a+0.211*C_f_a))*(J_w*(0.789*C_mf_a+0.211*C_f_a)-J_a))...
173     +(D_1_T-D_HA_T)/(sqrt(1+4*K*(0.211*C_mf_a+0.789*C_f_a))*(J_w*(0.211*C_mf_a...
174     +0.789*C_f_a)-J_a))+ D_HA_T/(J_w*(0.789*C_mf_a+0.211*C_f_a)-J_a)...
175     +D_HA_T/((J_w*(0.211*C_mf_a+0.789*C_f_a)-J_a))))+D_f_a/k_f_a;
176 %
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
177 fu(50)=(0.5*(C_f_a1-C_mf_a1)*((D_11_T-D_HA1_T)/...                      %
C_f_a1,D_f_a1,k_f_a1
178     (sqrt(1+4*K1*(0.789*C_mf_a1+0.211*C_f_a1))*(J_w*(0.789*C_mf_a1+0.211*C_f_a1)-
J_a1))...
179     +(D_11_T-D_HA1_T)/(sqrt(1+4*K1*(0.211*C_mf_a1+0.789*C_f_a1))*(J_w*(0.211
*C_mf_a1...
180     +0.789*C_f_a1)-J_a1))+ D_HA1_T/(J_w*(0.789*C_mf_a1+0.211*C_f_a1)-J_a1)...
181     +D_HA1_T/((J_w*(0.211*C_mf_a1+0.789*C_f_a1)-J_a1))))+D_f_a1/k_f_a1;
182 %
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
183 fu(51)=Sh_f_a*D_f_a/d_h-k_f_a;
184 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
185 fu(52)=Sh_f_a1*D_f_a1/d_h-k_f_a1;                    %Sh_f_a1
186 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
187 fu(53)=(D_1_T/(2*K*C_f_a)*(-1+sqrt(1+4*K*C_f_a))+D_HA_T/...
188     (4*K*C_f_a)*(-1+sqrt(1+4*K*C_f_a))^2)-D_f_a;
189 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
190 fu(54)=(D_11_T/(2*K1*C_f_a1)*(-1+sqrt(1+4*K1*C_f_a1))+D_HA1_T/...
191     (4*K1*C_f_a1)*(-1+sqrt(1+4*K1*C_f_a1))^2)-D_f_a1;
192 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
193 fu(55)=L*v_f*den_f/mul_f -Re_f;
194 fu(56)=(f+f_1)/W/D/2-v_f;
195 fu(57)=f-J_w*A_m-f_1;
196 fu(58)= 0.04*Re_f^(3/4)*Sc_f_a^(1/3)-Sh_f_a;
197 fu(59)= mul_f/den_f/D_f_a-Sc_f_a;
198 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
199 fu(60)= 0.04*Re_f^(3/4)*Sc_f_a1^(1/3)-Sh_f_a1;      %Sc_f_a1
200 fu(61)= mul_f/den_f/D_f_a1-Sc_f_a1;
201 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
202 fu(62)=(C_f_s-C_mf_s)/2*D_S_T*(1/(J_w*(0.789*C_mf_s+0.211*C_f_s)+J_s)...
203 +1/(J_w*(0.211*C_mf_s+0.789*C_f_s)+ J_s))+D_f_s/k_f_s;
204 fu(63)=Sh_f_s*D_f_s/d_h-k_f_s;
205 %Mole balance Equation
206 fu(64)=(V_f_0*C_f_a_0+C_f_ai*f_1*t)/(V_f_0+f_1*t)-C_f_a;
207 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
208 fu(65)=(V_f_0*C_f_a1_0+C_f_ai1*f_1*t)/(V_f_0+f_1*t)-C_f_a1;
209 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
210 fu(66)=C_f_s*f_1*t/(V_f_0+f_1*t)-C_f_si;
211 fu(67)=(C_f_a*f-J_a*A_m)/f_1-C_f_ai;
212 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
213 fu(68)=(C_f_a1*f-J_a1*A_m)/f_1-C_f_ai1;
214 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
```

```
215  fu(69)=(C_f_si*f+J_s*A_m)/f_1-C_f_s;
216  fu(70)=(C_d_s_0*V_d_0+C_d_si*f_2*t)/(V_d_0+f_2*t)-C_d_s;
217  fu(71)=(C_d_ai*f*t+C_d_ai*(V_d_0+f_2*t-f*t))/(f_2*t)-C_d_a;
218  %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
219  fu(72)=(C_d_ai1*f*t+C_d_ai1*(V_d_0+f_2*t-f*t))/(f_2*t)-C_d_a1;
220  %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
221  fu(73)=(C_d_s*f-J_s*A_m)/f_2-C_d_si;
222  fu(74)=(C_d_a*f_2-J_a*A_m)/f-C_d_ai;
223  %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
224  fu(75)=(C_d_a1*f_2-J_a1*A_m)/f-C_d_ai1;
225  %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
226  %additional Equation
227  fu(76)= 0.04*Re_f^(3/4)*Sc_f_s^(1/3)-Sh_f_s;
228  fu(77)= mul_f/den_f/D_f_s-Sc_f_s;
229  fu(78)=J_w*A_m*t*den_d/1000-w;
230  %ionic strenght in feed solution%%%%%%%%%%%%%%%%%%%
231  fu(79)=0.5*(C_f_si+C_f_NH4+C_f_H+C_f_OH+C_f_A+C_f_A1+C_f_HCO3+4*C_f_CO3)- I_f;%C_f_A1
232  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
233  fu(80)=10^(-(A_I*(I_f^(1/2)/(1+I_f^(1/2))- 0.3*I_f)))-gamma_f_H;%Davis
234  fu(81)=10^(-(A_I*(I_f^(1/2)/(1+I_f^(1/2))- 0.3*I_f)))-gamma_f_A;%Davis
235  %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
236  fu(82)=10^(-(A_I*(I_f^(1/2)/(1+I_f^(1/2))- 0.3*I_f)))-gamma_f_A1;%Davis         %↙
gamma_f_A1
237  %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
238  fu(83)=10^(-(A_I*(I_f^(1/2)/(1+I_f^(1/2))- 0.3*I_f)))-gamma_f_HCO3;%Davis
239  fu(84)=10^(-(A_I*(I_f^(1/2)/(1+I_f^(1/2))- 0.3*I_f)))-gamma_f_OH;%Davis
240  %Charge Balance Equation %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
241  fu(85)=C_f_HCO3+2*C_f_CO3+C_f_A+C_f_A1+C_f_OH+C_f_si-C_f_NH4-C_f_H;
242  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
243  %New Equation for charge balance
244  fu(86)=K_a_T*C_f_HA/gamma_f_H/gamma_f_A/C_f_H-C_f_A;
245  %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
246  fu(87)=K_a1_T*C_f_HA1/gamma_f_H/gamma_f_A1/C_f_H-C_f_A1;      %C_f_HA1
247  %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
248  fu(88)=K_CO2_T/gamma_f_H/gamma_f_HCO3/C_f_H*C_f_HCO3e-C_f_HCO3;%******
249  fu(89)=K_HCO3_T*gamma_f_HCO3/gamma_f_H/gamma_f_CO3/C_f_H*C_f_CO3e-C_f_CO3;%******
250  fu(90)=10^(-(A_I*4*(I_f^(1/2)/(1+I_f^(1/2))- 0.3*I_f)))-gamma_f_CO3;%Davis
251  fu(91)=K_w_T/C_f_H/gamma_f_OH/gamma_f_H-C_f_OH;
252  fu(92)=C_f_NH4+C_f_NH3-C_f_si;
253  fu(93)=C_f_HA+C_f_A-C_f_a;
254  %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
255  fu(94)=C_f_HA1+C_f_A1-C_f_a1; %C_f_HA1
256  %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
257  fu(95)=gamma_f_H*C_f_H*C_f_NH3/gamma_f_NH4/K_NH4CL-C_f_NH4;
258  fu(96)=10^(-(A_I*(I_f^(1/2)/(1+I_f^(1/2))- 0.3*I_f)))-gamma_f_NH4;
259  fu(97)=pH_f+log10(gamma_f_H*C_f_H);
260  fu(98)=C_f_co2-C_f_HCO3-C_f_HCO3e;
261  fu(99)=C_f_HCO3-C_f_CO3-C_f_CO3e;
262  %protron condition
263  %fu(51)=C_f_HCO3+2*C_f_CO3+C_f_Ac+C_f_OH+C_f_NH3-C_f_H;
264
265
266
```

G.9 Acetic acid as feed solution and NaCl as draw solution

```
 1 clear
 2 clc
 3 %ts=0.44  for mathlab Den,Vis for den_dM1 mul_dM2 work
 4 [g]=[1.545e-03     2.1245e+01    2.700e-01     1.934e-05  4.535e-01     1.278e-04↙
7.366e-11    1.013e-02    4.342e-08...
 5 4.219e-08     2.6398e-06    7.898e-01    4.535e-01  9.994e-01    4.234e-04    9.735↙
e-01    5.512e-04    9.893e-01...
 6 9.545e-06     1.4718e-01    6.754e+02    5.550e+02  2.612e+04    1.503e+02    1.034↙
e+03    5.479e+00    1.581e+00...
 7 1.060e-05     1.5607e-01    1.000e-02    6.452e+02  2.545e+04    1.502e+02    1.579↙
e+00    5.129e+02    9.258e-06...
 8 1.426e-01     8.7212e-06    1.001e-02    3.576e-06  3.576e-08    9.990e-01    3.388↙
e+00    9.767e-01    4.194e-04...
 9 4.229e-04     5.4140e+00    6.748e+02    5.870e+02  2.953e-04    4.194e-04    1.341↙
e-08    5.497e-09    6.303e-09...
10 2.974e-08     1.1896e-05    2.601e-06    9.990e-01  3.856e-08    4.214e-08    7.306↙
e-01    1.176e-05];
11 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
12 ts=0.44:0.025:1800;
13 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
14 for ti=1:numel(ts);
15 t=ts(ti);
16 options = optimset('Algorithm','Levenberg-Marquardt','TolFun',1*10^-15,'TolX',1*10^-↙
15 ...
17     ,'Maxiter',25,'Maxfun',15000);
18
19 [r,fval,exitflag,output] = fsolve(@(z)Acetic_NaCl(z,t),[g],options)
20
21 [g]=[r];
22 j(ti)=r(1,1);
23 p(ti)=r(1,43);%pH_f
24 c(ti)=r(1,30);%C_f_a
25 d(ti)=r(1,50);%w
26 T(ti)=t;
27 d1(ti)=r(1,55);%J_co2
28 d2(ti)=r(1,52);%C_f_co2
29 d3(ti)=r(1,54);%C_f_HCO3C
30 d4(ti)=r(1,56);%C_d_co2
31 d5(ti)=r(1,57);%C_d_HCO3
32 d6(ti)=-log10(r(1,12)*r(1,11));%pH_d
33 d7(ti)=(1-r(1,43)/r(1,32))*100;% %Rejection
34 d8(ti)=r(1,32)/0.01;%Concentration Performance
35 %for Plotting
36 [A(ti)]=t;
37 [B(ti)]=d(ti);
38 [C(ti)]=p(ti);
39 [M(ti)]=d7(ti);
40 [N(ti)]=j(ti);
41 [K5(ti)]=d8(ti);
42 disp(ti)
43 fprintf('\n     Time(min)        J_w(L/dm2/min)            pH            Cf,a(mole/l)↙
W(kg)\n');
44
45 fprintf('\n%11.2f%20.4e%19.4f%19.4e%19.4e%\n' ,T(ti),j(ti),p(ti),c(ti),d(ti));
46
47 format short e
```

```matlab
48  %Result of Best guess
49  fprintf('\n%8.3e%15.4e%13.3e%14.3e%11.3e%13.3e%13.3e%13.3e%13.3e\n'...
50      ,r(1,1),r(1,2),r(1,3),r(1,4),r(1,5),r(1,6),r(1,7),r(1,8),r(1,9));
51
52   fprintf('\n%8.3e%15.4e%13.3e%14.3e%11.3e%13.3e%13.3e%13.3e%13.3e\n'...
53      ,r(1,10),r(1,11),r(1,12),r(1,13),r(1,14),r(1,15),r(1,16),r(1,17),r(1,18));
54
55  fprintf('\n%8.3e%15.4e%13.3e%14.3e%11.3e%13.3e%13.3e%13.3e%13.3e\n'...
56      ,r(1,19),r(1,20),r(1,21),r(1,22),r(1,23),r(1,24),r(1,25),r(1,26),r(1,27));
57
58  fprintf('\n%8.3e%15.4e%13.3e%14.3e%11.3e%13.3e%13.3e%13.3e%13.3e\n'...
59      ,r(1,28),r(1,29),r(1,30),r(1,31),r(1,32),r(1,33),r(1,34),r(1,35),r(1,36));
60
61  fprintf('\n%8.3e%15.4e%13.3e%14.3e%11.3e%13.3e%13.3e%13.3e%13.3e%13.3e\n'...
62      ,r(1,37),r(1,38),r(1,39),r(1,40),r(1,41),r(1,42),r(1,43),r(1,44),r(1,45));
63
64  fprintf('\n%8.3e%15.4e%13.3e%14.3e%11.3e%13.3e%13.3e%13.3e%13.3e\n'...
65      ,r(1,46),r(1,47),r(1,48),r(1,49),r(1,50),r(1,51),r(1,52),r(1,53),r(1,54));
66
67  fprintf('\n%8.3e%15.4e%13.3e%14.3e%11.3e%13.3e%13.3e%13.3e%\n'...
68      ,r(1,55),r(1,56),r(1,57),r(1,58),r(1,59),r(1,60),r(1,61),r(1,62));
69  end
70
71  %Time Interval%%%%%%%%%%%%%%%%%%%%%%%%%%
72  Ai=0:30:1800;
73  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
74  format short
75  Ci=spline(A,C,Ai);
76  format short e
77  Bi=spline(A,B,Ai);
78
79  %Experiment Value W and pH
80  F=[ 0     0.0170    0.0310    0.0450    0.0600    0.0770    0.0900    0.0990    0.1130↵
0.1300    0.1450    0.1590    0.1710...
81  0.1820    0.1960    0.2050    0.2210    0.2310    0.2440    0.2550    0.2630↵
0.2750    0.2890    0.2980    0.3080    0.3230...
82  0.3320    0.3400    0.3540    0.3670    0.3760    0.3880    0.3990    0.4130↵
0.4220    0.4290    0.4400    0.4510    0.4610...
83  0.4680    0.4770    0.4860    0.4940    0.5030    0.5120    0.5220    0.5300↵
0.5390    0.5470    0.5560    0.5640    0.5720...
84  0.5800    0.5880    0.5980    0.6040    0.6140    0.6190    0.6280    0.6340↵
0.6450];
85
86  G=[3.4270    3.3840    3.3600    3.3570    3.3710    3.3510    3.3610    3.3570↵
3.3140    3.3270    3.3000    3.2850    3.2890...
87  3.2760    3.2660    3.2520    3.2410    3.2290    3.2190    3.2070    3.1980↵
3.1880    3.1770    3.1680    3.1600    3.1400...
88  3.1260    3.1130    3.1140    3.0910    3.0870    3.0720    3.0640    3.0580↵
3.0400    3.0360    3.0250    3.0110    3.0050...
89  2.9950    2.9800    2.9680    2.9610    2.9510    2.9420    2.9280    2.9210↵
2.9110    2.9030    2.8950    2.8860    2.8770...
90  2.8700    2.8610    2.8510    2.8440    2.8360    2.8320    2.8250    2.8180↵
2.8110];
91  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
92  F_m=mean(F);
93  G_m=mean(G);
```

```matlab
 94
 95 sum1=0;sum2=0;
 96 sum3=0;sum4=0;
 97 %%% Number of Data %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 98 for k=1:1:61
 99 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
100 sum1=(Bi(1,k)-F(1,k))^2+sum1;
101 sum2=(Bi(1,k)-F_m)^2+sum2;
102 sum3=(Ci(1,k)-G(1,k))^2+sum3;
103 sum4=(Ci(1,k)-G_m)^2+sum4;
104 end
105
106 format short
107 R2_W=1-sum1/sum2;
108 R2_pH=1-sum3/sum4;
109 MSE_W=sum1/k;
110 MSE_pH=sum3/k;
111 RMSE_W=sqrt(MSE_W)
112 RMSE_pH=sqrt(MSE_pH)
113 SEP_W=RMSE_W/F_m*100
114 SEP_pH=RMSE_pH/G_m*100
115
116 % Plotting
117 subplot(2,3,1)
118 plot([A]/60,[B],'-g',[Ai]/60,[F],'xr')
119 xlabel('Time (hr)')
120 ylabel('Weight (Kg)')
121 title('Weight Change as a Function of Time')
122 legend('Model','Experiment',2)
123
124 subplot(2,3,2)
125 plot([A]/60,[C],'-g',[Ai]/60,[G],'xr')
126 xlabel('Time (hr)')
127 ylabel('pH')
128 title('pH as a Function of Time')
129 legend('Model','Experiment',1)
130
131 subplot(2,3,3)
132 plot([A]/60,[M],'-g')
133 xlabel('Time (hr)')
134 ylabel('%Rejection')
135 title('%Rejection as a Function of Time')
136 legend('Model',1)
137
138 subplot(2,3,4)
139 plot([N],[M],'-g')
140 xlabel('Jw (L/dm2/min)')
141 ylabel('%Rejection')
142 title('%Rejection as a Function of Jw')
143 legend('Model',1)
144
145 subplot(2,3,5)
146 plot([A]/60,[K5],'-g')
147 xlabel('Time (hr)')
148 ylabel('Cfa/Cf0')
149 title('Cfa/Cf0 as a Function of Time')
```

```
150 legend('Model',1)
151
152 subplot(2,3,6)
153 plot([N],[K5],'-g')
154 xlabel('Jw (L/dm2/min)')
155 ylabel('Cfa/Cf0')
156 title('Cfa/Cf0 as a Function of Jw')
157 legend('Model',1)
158
159 fprintf('\n      \n')
160 fprintf('\n      \n')
161 fprintf('\n CONCLUSION     \n')
162 fprintf('\n         R2_W             MSE_W                    R2_pH          MSE_pH\n');
163 fprintf('\n%11.3f%20.4e%19.4f%19.4e%\n',R2_W,MSE_W,R2_pH,MSE_pH)
164 fprintf('\n      \n')
165 fprintf('\n     Time(min)       J_w(L/dm2/min)             pH             Cf,a(mole/l)↙
W(kg)\n');
166
167 for x=5:5:20000;
168 z=x*4;
169 fprintf('\n%11.4f %20.4e%19.4f%19.4e%19.4e\n',T(z),j(z),p(z),c(z),d(z));
170
171 end
172
173
```

```
 1
 2 function fu=Acetic_NaCl(z,t)
 3 %Independent Variable
 4 V_f_0=1;C_d_s_0=1;C_f_a_0=10*10^(-3);V_d_0=0.5;
 5 %Fixed Variable
 6 T=301;f=1.58;
 7 % T=300..normal f=1.58...normal
 8 L=0.9207;D=0.023;W=0.4572;A_m=42*10^(-2);d_h=0.0438;
 9 %Constant Variable
10
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
11 B_a=2.093/60/100;%B of Acetic Acid,
12 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
13 B_s=0.256/60/100;
14 A=0.442/60/100;
15 S=500*10^-5;
16 K_a_T=10^-(1170.48/T-3.1649+0.013399*T);%Acetic acid
17 D_A_298=6.534*10^(-6);D_HA_298=7.74*10^(-6);%Acetic Physicocmical Property
18 D_H_298=5.587*10^(-5);
19
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%5
20 o_s=0.936;
21 den_f=999.65+2.0438/10*(T-273)-6.174*10^-2*(T-273)^1.5;
22 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
23 n_s=2;o_a=1;R=0.08314;
24 %Temperature Correction
25 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
26 K_CO2_T=10^-(3404.71/T-14.8435+0.032786*T);
27 K_w_T=10^(-4470.99/T+6.0875-0.017060*T);
28 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
29 E=2727.586+0.6224107*T-466.9151*log(T)-52000.87/T;
30 d=1-(((T-273)-3.9863)^2*((T-273)+288.9414))/(508929.2*((T-273)+68.12963))...
31 +0.011445*exp((-374.3)/(T-273));
32 A_I=(1.82483*10^6*d^0.5)/(E*T)^1.5;
33 K=1/K_a_T;
34 visco_298=((298-273)+246)/((0.05594*(298-273)+5.2842)*(298-273)+137.37);
35 visco_T=((T-273)+246)/((0.05594*(T-273)+5.2842)*(T-273)+137.37);
36 D_H_T=D_H_298*T/298*visco_298/visco_T ;
37 D_A_T=D_A_298*T/298*visco_298/visco_T;
38 D_HA_T=D_HA_298*T/298*visco_298/visco_T;
39 D_1_T=2/(1/D_H_T +1/D_A_T);
40 K_H=0.034*exp(-2400*(1/T-1/298));%************************
41 %For carbonic acid
42 P_M =0.0025;
43 %Unknown Variable
44 J_w=z(1);P_i=z(2);P_m=z(3);J_s=z(4);C_i_s=z(5);
45 C_mf_s=z(6);J_a=z(7);C_mf_a=z(8);C_i_a=z(9);C_d_a=z(10);
46 C_d_H=z(11);gamma_d_H=z(12);I_i=z(13);C_d_s=z(14);
47 C_mf_H=z(15);gamma_m_H=z(16);I_m=z(17);C_md_s=z(18);
48 D_d_s=z(19);k_d=z(20);Sh_d=z(21);Sc_d=z(22);Re_d=z(23);
49 v_d=z(24);den_d=z(25);mul_d=z(26);f_2=z(27);D_f_a=z(28);
50 k_f_a=z(29);C_f_a=z(30);Sh_f_a=z(31);Re_f=z(32);v_f=z(33);
51 f_1=z(34);Sc_f_a=z(35);D_f_s=z(36);k_f_s=z(37);C_f_s=z(38);
52 C_f_ai=z(39);C_f_si=z(40);C_d_ai=z(41);C_d_si=z(42);pH_f=z(43);
```

```
53 gamma_f_H=z(44);C_f_A=z(45);I_f=z(46);mul_f=z(47);
54 Sh_f_s=z(48);Sc_f_s=z(49);w=z(50);C_f_H=z(51);
55 C_f_co2=z(52);C_f_co2i=z(53);C_f_HCO3c=z(54);
56 J_co2=z(55);C_d_co2=z(56);C_d_HCO3=z(57);I_d=z(58);C_d_A=z(59);
57 C_i_H=z(60);gamma_i_H=z(61);C_d_co2a=z(62);
58 %Math Model
59 fu(1)=A*(P_i-P_m )-J_w;
60 fu(2)= B_s *(C_i_s-C_mf_s)-J_s;
61 fu(3)= B_a*(C_mf_a-C_i_a)-J_a;
62 fu(4)=C_d_a-C_i_a;
63 fu(5)=J_a/J_w-C_d_a;
64 fu(6)= o_s*n_s*C_i_s*R*T+o_a*(C_i_a+C_i_H)*R*T-P_i;
65 fu(7)=(-(K_a_T)+((K_a_T)^2+4*K_a_T*gamma_i_H*gamma_i_H*C_d_a)^(1/2))...
66     /(2*gamma_i_H*gamma_i_H)-C_i_H;
67 fu(8)=A_I*(I_i^(1/2)/(1+I_i^(1/2))- 0.3*I_i)+log10(gamma_i_H);%"Davies"
68 fu(9)=0.5*(2*C_i_s+2*C_i_H)- I_i;
69 fu(10)=o_s*n_s*C_mf_s*R*T+o_a*(C_mf_a+C_mf_H)*R*T-P_m;
70 fu(11)=(-(K_a_T)+((K_a_T)^2+4*K_a_T*gamma_m_H*gamma_m_H*C_mf_a)^(1/2))...
71     /(2*gamma_m_H*gamma_m_H)-C_mf_H;
72 fu(12)=A_I*(I_m^(1/2)/(1+I_m^(1/2))- 0.3*I_m)+log10(gamma_m_H);%"Davies"
73 fu(13)=0.5*(2*C_mf_s+2*C_mf_H)-I_m;
74
75 fu(14)=((C_md_s-C_i_s)*(12.625*10^(-11)*((0.789*C_i_s+0.211*C_md_s)^4 ...
76 /(J_w*(0.789*C_i_s+0.211*C_md_s)+ J_s)+(0.211*C_i_s+0.789*C_md_s)^4 ...
77 /(J_w*(0.211*C_i_s+0.789*C_md_s)+ J_s ))...
78 -3.885*10^(-9)*((0.789*C_i_s+0.211*C_md_s)^3 ...
79 /(J_w*(0.789*C_i_s+0.211*C_md_s)+ J_s)+(0.211*C_i_s+0.789*C_md_s)^3 ...
80 /(J_w*(0.211*C_i_s+0.789*C_md_s)+ J_s ))...
81 +6.975*10^(-8)*((0.789*C_i_s+0.211*C_md_s)^2 ...
82 /(J_w*(0.789*C_i_s+0.211*C_md_s)+ J_s)+(0.211*C_i_s+0.789*C_md_s)^2 ...
83 /(J_w*(0.211*C_i_s+0.789*C_md_s)+ J_s ))...
84 -4.03*10^(-7)*((0.789*C_i_s+0.211*C_md_s)...
85 /(J_w*(0.789*C_i_s+0.211*C_md_s)+ J_s)+(0.211*C_i_s+0.789*C_md_s)...
86 /(J_w*(0.211*C_i_s+0.789*C_md_s)+ J_s))...
87 +0.53*10^(-5)*(1/(J_w*(0.789*C_i_s+0.211*C_md_s)+J_s)...
88 +1/(J_w*(0.211*C_i_s+0.789*C_md_s)+ J_s))))-S;
89
90 fu(15)=((C_d_s-C_md_s)*(12.625*10^(-11)*((0.789*C_md_s+0.211*C_d_s)^4 ...
91 /(J_w*(0.789*C_md_s+0.211*C_d_s)+ J_s)+(0.211*C_md_s+0.789*C_d_s)^4 ...
92 /(J_w*(0.211*C_md_s+0.789*C_d_s)+ J_s ))...
93 -3.885*10^(-9)*((0.789*C_md_s+0.211*C_d_s)^3 ...
94 /(J_w*(0.789*C_md_s+0.211*C_d_s)+ J_s)+(0.211*C_md_s+0.789*C_d_s)^3 ...
95 /(J_w*(0.211*C_md_s+0.789*C_d_s)+ J_s ))...
96 +6.975*10^(-8)*((0.789*C_md_s+0.211*C_d_s)^2 ...
97 /(J_w*(0.789*C_md_s+0.211*C_d_s)+ J_s)+(0.211*C_md_s+0.789*C_d_s)^2 ...
98 /(J_w*(0.211*C_md_s+0.789*C_d_s)+ J_s ))...
99 -4.03*10^(-7)*((0.789*C_md_s+0.211*C_d_s)...
100 /(J_w*(0.789*C_md_s+0.211*C_d_s)+ J_s)+(0.211*C_md_s+0.789*C_d_s)...
101 /(J_w*(0.211*C_md_s+0.789*C_d_s)+ J_s))...
102 +0.53*10^(-5)*(1/(J_w*(0.789*C_md_s+0.211*C_d_s)+J_s)...
103 +1/(J_w*(0.211*C_md_s+0.789*C_d_s)+ J_s))))-D_d_s/k_d;
104
105 fu(16)=Sh_d*D_d_s/d_h-k_d;
106 fu(17)=5.05*10^(-11)*C_d_s^4-2.59*10^(-9)*C_d_s^3+4.65*10^(-8)*C_d_s^2-4.03*10^(-7)
*C_d_s+1.06*10^(-5)-D_d_s;
107 fu(18)=L*v_d*den_d/mul_d-Re_d;
```

```
108 fu(19)=(f_2+f)/2/W/D-v_d;
109 fu(20)=f+J_w*A_m-f_2;
110
111 %↙
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^↙
^^^^^^^^^^^^^^^^^^
112 fu(21)=0.0538*C_d_s+0.4159+5.0095/C_d_s-mul_d/C_d_s;%Math Lab
113 fu(22)=37.0166*C_d_s+997.2911-den_d;%Math lab
114 %↙
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^↙
^^^^^^^^^^^^^^^^^^^^^^
115 fu(23)= 0.04*Re_d^(3/4)*Sc_d^(1/3)-Sh_d;
116 fu(24)= mul_d/den_d/D_d_s-Sc_d;
117 fu(25)=(0.5*(C_f_a-C_mf_a)*((D_1_T-D_HA_T)/...
118     (sqrt(1+4*K*(0.789*C_mf_a+0.211*C_f_a))*(J_w*(0.789*C_mf_a+0.211*C_f_a)-J_a))...
119     +(D_1_T-D_HA_T)/(sqrt(1+4*K*(0.211*C_mf_a+0.789*C_f_a))*(J_w*(0.211*C_mf_a...
120     +0.789*C_f_a)-J_a))+ D_HA_T/(J_w*(0.789*C_mf_a+0.211*C_f_a)-J_a)...
121     +D_HA_T/((J_w*(0.211*C_mf_a+0.789*C_f_a)-J_a))))+D_f_a/k_f_a;
122 fu(26)=Sh_f_a*D_f_a/d_h-k_f_a;
123 fu(27)=(D_1_T/(2*K*C_f_a)*(-1+sqrt(1+4*K*C_f_a))+D_HA_T/...
124     (4*K*C_f_a)*(-1+sqrt(1+4*K*C_f_a))^2)-D_f_a;
125 fu(28)=L*v_f*den_f/mul_f -Re_f;
126 fu(29)=(f+f_1)/W/D/2-v_f;
127 fu(30)=f-J_w*A_m-f_1;
128 fu(31)= 0.04*Re_f^(3/4)*Sc_f_a^(1/3)-Sh_f_a;
129 fu(32)= mul_f/den_f/D_f_a-Sc_f_a;
130
131 fu(33)=((C_f_s-C_mf_s)*(12.625*10^(-11)*((0.789*C_mf_s+0.211*C_f_s)^4 ...
132 /(J_w*(0.789*C_mf_s+0.211*C_f_s)+ J_s)+(0.211*C_mf_s+0.789*C_f_s)^4 ...
133 /(J_w*(0.211*C_mf_s+0.789*C_f_s)+ J_s ))...
134 -3.885*10^(-9)*((0.789*C_mf_s+0.211*C_f_s)^3 ...
135 /(J_w*(0.789*C_mf_s+0.211*C_f_s)+ J_s)+(0.211*C_mf_s+0.789*C_f_s)^3 ...
136 /(J_w*(0.211*C_mf_s+0.789*C_f_s)+ J_s ))...
137 +6.975*10^(-8)*((0.789*C_mf_s+0.211*C_f_s)^2 ...
138 /(J_w*(0.789*C_mf_s+0.211*C_f_s)+ J_s)+(0.211*C_mf_s+0.789*C_f_s)^2 ...
139 /(J_w*(0.211*C_mf_s+0.789*C_f_s)+ J_s ))...
140 -4.03*10^(-7)*((0.789*C_mf_s+0.211*C_f_s)...
141 /(J_w*(0.789*C_mf_s+0.211*C_f_s)+ J_s)+(0.211*C_mf_s+0.789*C_f_s)...
142 /(J_w*(0.211*C_mf_s+0.789*C_f_s)+ J_s))...
143 +0.53*10^(-5)*(1/(J_w*(0.789*C_mf_s+0.211*C_f_s)+J_s)...
144 +1/(J_w*(0.211*C_mf_s+0.789*C_f_s)+ J_s))))+D_f_s/k_f_s;
145
146 fu(34)=Sh_f_s*D_f_s/d_h-k_f_s;
147 fu(35)=5.05*10^(-11)*C_f_s^4-2.59*10^(-9)*C_f_s^3+4.65*10^(-8)*C_f_s^2-4.03*10^(-7)↙
*C_f_s+1.06*10^(-5)-D_f_s;
148
149 %Mole balance Equation
150 fu(36)=C_f_a*(V_f_0+f_1*t-f*t)-(V_f_0*C_f_a_0+C_f_ai*f_1*t-C_f_a*f*t);%new
151 fu(37)=C_f_si*(V_f_0+f_1*t-f*t)-C_f_s*f_1*t+C_f_si*f*t;%old
152 fu(38)=C_f_a*f-J_a*A_m-C_f_ai*f_1;%new
153 fu(39)=C_f_si*f+J_s*A_m-C_f_s*f_1;%old
154 fu(40)=C_d_s_0*V_d_0+C_d_si*f_2*t-C_d_s*f*t-C_d_s*(V_d_0+f_2*t-f*t);%new
155 fu(41)=C_d_a*f_2*t-C_d_ai*f*t-C_d_ai*(V_d_0+f_2*t-f*t);%old
156 fu(42)=C_d_s*f-J_s*A_m-C_d_si*f_2;%new
157 fu(43)=C_d_ai*f+J_a*A_m-C_d_a*f_2;%old
158
```

```
159  %additional Equation
160  fu(44)= 0.04*Re_f^(3/4)*Sc_f_s^(1/3)-Sh_f_s;
161  fu(45)= mul_f/den_f/D_f_s-Sc_f_s;
162  fu(46)=J_w*A_m*t*den_d/1000-w;
163
164  %pH Equation..
165  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
166  fu(47)= pH_f+log10(gamma_f_H*C_f_H);
167  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
168  fu(48)=K_a_T*(C_f_a-C_f_A)/gamma_f_H/gamma_f_H/C_f_H-C_f_A;
169  fu(49)=A_I*(I_f^(1/2)/(1+I_f^(1/2))-0.3*I_f)+log10(gamma_f_H);%Davies;
170  %ionic streanght in feed solution%%%%%%%%%%%%%%%%%%%%
171  fu(50)=0.5*(2*C_f_si+C_f_H+K_w_T/C_f_H/gamma_f_H/gamma_f_H+C_f_A+C_f_HCO3c)- I_f;
172  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
173  %Charge Balance Equation %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
174  fu(51)=(C_f_HCO3c+C_f_A+K_w_T/C_f_H/gamma_f_H/gamma_f_H)-C_f_H;
175  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
176  fu(52)=P_M*C_d_co2-J_co2;
177  fu(53)=C_f_co2i*(V_f_0+f_1*t-f*t)-(C_f_co2*f_1*t-C_f_co2i*f*t);
178  fu(54)=C_f_co2i*f*t+J_co2*A_m*t-C_f_co2*f_1*t;
179  fu(55)=10^-3.45/gamma_f_H/gamma_f_H/C_f_H*(C_f_co2-C_f_HCO3c)-C_f_HCO3c;
180  % Draw solution
181  fu(56)=gamma_d_H*gamma_d_H*C_d_H*C_d_HCO3/K_CO2_T+C_d_HCO3-C_d_co2;%******
182  fu(57)=(-K_CO2_T+sqrt(K_CO2_T^2+4*K_CO2_T*gamma_d_H*gamma_d_H*C_d_co2a))/↵
     (2*gamma_d_H*gamma_d_H)-C_d_HCO3;
183  %ionic streanght in draw solution%%%%%%%%%%%%%%%%%%%%
184  fu(58)=0.5*(2*C_d_si+C_d_H+K_w_T/C_d_H/gamma_d_H/gamma_d_H+C_d_A+C_d_HCO3)- I_d;
185  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
186  %Charge balance at draw solution%%%%%%%%%%%%%%%%%%%%
187  fu(59)=(C_d_HCO3+C_d_A+K_w_T/C_f_H/gamma_d_H/gamma_d_H)-C_d_H;
188  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
189  fu(60)=K_a_T*(C_d_a-C_d_A)/gamma_d_H/gamma_d_H/C_d_H-C_d_A;
190  fu(61)=10^-(A_I*(I_d^(1/2)/(1+I_d^(1/2))-0.3*I_d))-gamma_d_H;%Davies;
191  fu(62)=10^-3.408*K_H/10^((((33.5-0.109*(T-273)+0.0014*(T-273)^2)*I_d-(1.5+0.015*(T-↵
     273)+0.004*(T-273)^2)*I_d^2))...
192      /T)-C_d_co2a;
193
194
```

G.10 Butyric acid as feed solution and NaCl as draw solution

```
 1 clear
 2 clc
 3 [g]=[1.545e-03    2.1244e+01    2.706e-01    1.934e-05 4.535e-01    1.283e-04↙
2.100e-11    1.018e-02    1.203e-08...
 4 1.192e-08    2.6120e-06    7.898e-01    4.535e-01 9.993e-01    3.927e-04    9.742↙
e-01    5.210e-04    9.892e-01...
 5 9.545e-06    1.4718e-01    6.754e+02    5.550e+02 2.612e+04    1.503e+02    1.034↙
e+03    5.479e+00    1.581e+00...
 6 1.060e-05    1.5607e-01    1.000e-02    6.452e+02 2.545e+04    1.502e+02    1.579↙
e+00    5.129e+02    9.258e-06...
 7 1.426e-01    9.2088e-06    1.001e-02    4.063e-06 1.007e-08    9.989e-01    3.421↙
e+00    9.775e-01    3.879e-04...
 8 3.920e-04    5.4140e+00    6.748e+02    5.870e+02 3.356e-04    3.879e-04    1.404↙
e-08    6.196e-09    6.868e-09...
 9 2.950e-08    1.1798e-05    2.601e-06    9.989e-01 1.075e-08    1.191e-08    7.306↙
e-01    1.176e-05];
10 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
11 ts=0.5:0.05:1800;
12 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
13 for ti=1:numel(ts);
14 t=ts(ti);
15 options = optimset('Algorithm','Levenberg-Marquardt','TolFun',1*10^-15,'TolX',1*10^-↙
15 ...
16      ,'Maxiter',25,'Maxfun',15000);
17
18 [r,fval,exitflag,output] = fsolve(@(z)Butyric_NaCl(z,t),[g],options)
19 [g]=[r];
20 j(ti)=r(1,1);
21 p(ti)=r(1,43);%pH_f
22 c(ti)=r(1,30);%C_f_a
23 d(ti)=r(1,50);%w
24 T(ti)=t;
25 d1(ti)=r(1,55);%J_co2
26 d2(ti)=r(1,52);%C_f_co2
27 d3(ti)=r(1,54);%C_f_HCO3C
28 d4(ti)=r(1,56);%C_d_co2
29 d5(ti)=r(1,57);%C_d_HCO3
30 d6(ti)=-log10(r(1,12)*r(1,11));%pH_d
31 d7(ti)=(1-r(1,43)/r(1,32))*100;% %Rejection
32 d8(ti)=r(1,32)/0.01;%Concentration Performance
33 %for Plotting
34 [A(ti)]=t;
35 [B(ti)]=d(ti);
36 [C(ti)]=p(ti);
37 [M(ti)]=d7(ti);
38 [N(ti)]=j(ti);
39 [K5(ti)]=d8(ti);
40 disp(ti)
41 fprintf('\n     Time(min)       J_w(L/dm2/min)           pH           Cf,a(mole/l)↙
W(kg)\n');
42 fprintf('\n%11.2f%20.4e%19.4f%19.4e%19.4e%\n' ,T(ti),j(ti),p(ti),c(ti),d(ti));
43 format short e
44 %Result of Best guess
45 fprintf('\n%8.3e%15.4e%13.3e%14.3e%11.3e%13.3e%13.3e%13.3e%13.3e%\n'...
46      ,r(1,1),r(1,2),r(1,3),r(1,4),r(1,5),r(1,6),r(1,7),r(1,8),r(1,9));
47
```

```
48  fprintf('\n%8.3e%15.4e%13.3e%14.3e%11.3e%13.3e%13.3e%13.3e%\n'...
49      ,r(1,10),r(1,11),r(1,12),r(1,13),r(1,14),r(1,15),r(1,16),r(1,17),r(1,18));
50
51  fprintf('\n%8.3e%15.4e%13.3e%14.3e%11.3e%13.3e%13.3e%13.3e%\n'...
52      ,r(1,19),r(1,20),r(1,21),r(1,22),r(1,23),r(1,24),r(1,25),r(1,26),r(1,27));
53
54  fprintf('\n%8.3e%15.4e%13.3e%14.3e%11.3e%13.3e%13.3e%13.3e%\n'...
55      ,r(1,28),r(1,29),r(1,30),r(1,31),r(1,32),r(1,33),r(1,34),r(1,35),r(1,36));
56
57  fprintf('\n%8.3e%15.4e%13.3e%14.3e%11.3e%13.3e%13.3e%13.3e%13.3e%13.3e%\n'...
58      ,r(1,37),r(1,38),r(1,39),r(1,40),r(1,41),r(1,42),r(1,43),r(1,44),r(1,45));
59
60  fprintf('\n%8.3e%15.4e%13.3e%14.3e%11.3e%13.3e%13.3e%13.3e%\n'...
61      ,r(1,46),r(1,47),r(1,48),r(1,49),r(1,50),r(1,51),r(1,52),r(1,53),r(1,54));
62
63  fprintf('\n%8.3e%15.4e%13.3e%14.3e%11.3e%13.3e%13.3e%13.3e%\n'...
64      ,r(1,55),r(1,56),r(1,57),r(1,58),r(1,59),r(1,60),r(1,61),r(1,62));
65  end
66
67  %Time Interval%%%%%%%%%%%%%%%%%%%%%%%%%%
68  Ai=0:30:1800;
69  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
70  format short
71  Ci=spline(A,C,Ai);
72  format short e
73  Bi=spline(A,B,Ai);
74
75  %Experiment Value W and pH%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
76  F=[ 0     0.0160    0.0320    0.0460    0.0630    0.0790    0.0940    0.1070    0.1210↙
0.1350    0.1490    0.1610    0.1730...
77  0.1870    0.1970    0.2100    0.2230    0.2340    0.2450    0.2610    0.2730↙
0.2850    0.2960    0.3060    0.3170    0.3280...
78  0.3390    0.3500    0.3600    0.3710    0.3800    0.3890    0.4000    0.4090↙
0.4190    0.4280    0.4370    0.4460    0.4550...
79  0.4620    0.4710    0.4800    0.4880    0.4960    0.5040    0.5120    0.5190↙
0.5270    0.5350    0.5420    0.5490    0.5560...
80  0.5630    0.5690    0.5750    0.5820    0.5880    0.5940    0.5990    0.6060↙
0.6100];
81
82  G=[3.448  3.432     3.427     3.422     3.411     3.405     3.401     3.394     3.389↙
3.383     3.376     3.365     3.353     3.357...
83  3.348     3.334     3.336     3.327     3.314     3.308     3.305     3.292     3.288↙
3.275     3.262     3.254     3.248     3.246     3.223...
84  3.211     3.204     3.199     3.197     3.182     3.174     3.165     3.153     3.145↙
3.133     3.136     3.128     3.111     3.103     3.092...
85  3.0806    3.078     3.074     3.069     3.055     3.046     3.035     3.031     3.026↙
3.018     3.013     3.005     2.997     2.993     2.989...
86  2.978     2.974];
87  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
88  F_m=mean(F);
89  G_m=mean(G);
90
91  sum1=0;sum2=0;
92  sum3=0;sum4=0;
93  %%% Number of Data %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
94  for k=1:1:61
```

```
 95 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 96 sum1=(Bi(1,k)-F(1,k))^2+sum1;
 97 sum2=(Bi(1,k)-F_m)^2+sum2;
 98 sum3=(Ci(1,k)-G(1,k))^2+sum3;
 99 sum4=(Ci(1,k)-G_m)^2+sum4;
100 end
101
102 format short
103 R2_W=1-sum1/sum2;
104 R2_pH=1-sum3/sum4;
105 MSE_W=sum1/k;
106 MSE_pH=sum3/k;
107 RMSE_W=sqrt(MSE_W)
108 RMSE_pH=sqrt(MSE_pH)
109 SEP_W=RMSE_W/F_m*100
110 SEP_pH=RMSE_pH/G_m*100
111
112 % Plotting
113 subplot(2,3,1)
114 plot([A]/60,[B],'-g',[Ai]/60,[F],'xr')
115 xlabel('Time (hr)')
116 ylabel('Weight (Kg)')
117 title('Weight Change as a Function of Time')
118 legend('Model','Experiment',2)
119
120 subplot(2,3,2)
121 plot([A]/60,[C],'-g',[Ai]/60,[G],'xr')
122 xlabel('Time (hr)')
123 ylabel('pH')
124 title('pH as a Function of Time')
125 legend('Model','Experiment',1)
126
127 subplot(2,3,3)
128 plot([A]/60,[M],'-g')
129 xlabel('Time (hr)')
130 ylabel('%Rejection')
131 title('%Rejection as a Function of Time')
132 legend('Model',1)
133
134 subplot(2,3,4)
135 plot([N],[M],'-g')
136 xlabel('Jw (L/dm2/min)')
137 ylabel('%Rejection')
138 title('%Rejection as a Function of Jw')
139 legend('Model',1)
140
141 subplot(2,3,5)
142 plot([A]/60,[K5],'-g')
143 xlabel('Time (hr)')
144 ylabel('Cfa/Cf0')
145 title('Cfa/Cf0 as a Function of Time')
146 legend('Model',1)
147
148 subplot(2,3,6)
149 plot([N],[K5],'-g')
150 xlabel('Jw (L/dm2/min)')
```

```matlab
151 ylabel('Cfa/Cf0')
152 title('Cfa/Cf0 as a Function of Jw')
153 legend('Model',1)
154
155 fprintf('\n     \n')
156 fprintf('\n     \n')
157 fprintf('\n CONCLUSION     \n')
158 fprintf('\n     R2_W          MSE_W                    R2_pH          MSE_pH\n');
159 fprintf('\n%11.3f%20.4e%19.4f%19.4e%\n',R2_W,MSE_W,R2_pH,MSE_pH)
160 fprintf('\n     \n')
161 fprintf('\n    Time(min)       J_w(L/dm2/min)           pH              Cf,a(mole/l) ↵
W(kg)\n');
162
163 for x=5:5:20000000;
164 z=x*4;
165 fprintf('\n%11.4f %20.4e%19.4f%19.4e%19.4e%\n',T(z),j(z),p(z),c(z),d(z));
166
167 end
168
169
```

```matlab
1
2 function fu=Butyric_NaCl(z,t)
3 %Independent Variable
4 V_f_0=1;C_d_s_0=1;C_f_a_0=10*10^(-3);V_d_0=0.5;
5 %Fixed Variable
6 T=301;f=1.58;
7 % T=300..normal f=1.58...normal
8 L=0.9207;D=0.023;W=0.4572;A_m=42*10^(-2);d_h=0.0438;
9 %Constant Variable
10
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
11 B_a=0.637/60/100;% B of Butyric Acid,
12 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
13 B_s=0.256/60/100;
14 A=0.442/60/100;
15 S=500*10^-5;
16 K_a_T=10^-(1033.39/T-2.6215+0.013334*T);%Butyric Acid
17 D_A_298=5.208*10^(-6);D_HA_298=5.429*10^(-6);%Butyric Physicocmical Property
18 D_H_298=5.587*10^(-5);
19
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%5
20 o_s=0.936;
21 den_f=999.65+2.0438/10*(T-273)-6.174*10^-2*(T-273)^1.5;
22 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
23 n_s=2;o_a=1;R=0.08314;
24 %Temperature Correction
25 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
26 K_CO2_T=10^-(3404.71/T-14.8435+0.032786*T);
27 K_w_T=10^(-4470.99/T+6.0875-0.017060*T);
28 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
29 E=2727.586+0.6224107*T-466.9151*log(T)-52000.87/T;
30 d=1-(((T-273)-3.9863)^2*((T-273)+288.9414))/(508929.2*((T-273)+68.12963))...
31 +0.011445*exp((-374.3)/(T-273));
32 %B=50.2916*d^0.5/(E*T)^0.5;
33 A_I=(1.82483*10^6*d^0.5)/(E*T)^1.5;
34 K=1/K_a_T;
35 visco_298=((298-273)+246)/((0.05594*(298-273)+5.2842)*(298-273)+137.37);
36 visco_T=((T-273)+246)/((0.05594*(T-273)+5.2842)*(T-273)+137.37);
37 D_H_T=D_H_298*T/298*visco_298/visco_T ;
38 D_A_T=D_A_298*T/298*visco_298/visco_T;
39 D_HA_T=D_HA_298*T/298*visco_298/visco_T;
40 D_1_T=2/(1/D_H_T +1/D_A_T);
41 K_H=0.034*exp(-2400*(1/T-1/298));%************************
42 %For carbonic acid
43 P_M =0.0025;
44 %Unknown Variable
45 J_w=z(1);P_i=z(2);P_m=z(3);J_s=z(4);C_i_s=z(5);
46 C_mf_s=z(6);J_a=z(7);C_mf_a=z(8);C_i_a=z(9);C_d_a=z(10);
47 C_d_H=z(11);gamma_d_H=z(12);I_i=z(13);C_d_s=z(14);
48 C_mf_H=z(15);gamma_m_H=z(16);I_m=z(17);C_md_s=z(18);
49 D_d_s=z(19);k_d=z(20);Sh_d=z(21);Sc_d=z(22);Re_d=z(23);
50 v_d=z(24);den_d=z(25);mul_d=z(26);f_2=z(27);D_f_a=z(28);
51 k_f_a=z(29);C_f_a=z(30);Sh_f_a=z(31);Re_f=z(32);v_f=z(33);
52 f_1=z(34);Sc_f_a=z(35);D_f_s=z(36);k_f_s=z(37);C_f_s=z(38);
```

```
53 C_f_ai=z(39);C_d_si=z(40);C_d_ai=z(41);C_d_si=z(42);pH_f=z(43);
54 gamma_f_H=z(44);C_f_A=z(45);I_f=z(46);mul_f=z(47);
55 Sh_f_s=z(48);Sc_f_s=z(49);w=z(50);C_f_H=z(51);
56 C_f_co2=z(52);C_f_co2i=z(53);C_f_HCO3c=z(54);
57 J_co2=z(55);C_d_co2=z(56);C_d_HCO3=z(57);I_d=z(58);C_d_A=z(59);
58 C_i_H=z(60);gamma_i_H=z(61);C_d_co2a=z(62);
59 %Math Model
60 fu(1)=A*(P_i-P_m )-J_w;
61 fu(2)= B_s *(C_i_s-C_mf_s)-J_s;
62 fu(3)= B_a*(C_mf_a-C_i_a)-J_a;
63 fu(4)=C_d_a-C_i_a;
64 fu(5)=J_a/J_w-C_d_a;
65 fu(6)= o_s*n_s*C_i_s*R*T+o_a*(C_i_a+C_i_H)*R*T-P_i;
66 fu(7)=(-(K_a_T)+((K_a_T)^2+4*K_a_T*gamma_i_H*gamma_i_H*C_d_a)^(1/2))...
67     /(2*gamma_i_H*gamma_i_H)-C_i_H;
68 fu(8)=A_I*(I_i^(1/2)/(1+I_i^(1/2))- 0.3*I_i)+log10(gamma_i_H);%"Davies"
69 fu(9)=0.5*(2*C_i_s+2*C_i_H)- I_i;
70 fu(10)=o_s*n_s*C_mf_s*R*T+o_a*(C_mf_a+C_mf_H)*R*T-P_m;
71 fu(11)=(-(K_a_T)+((K_a_T)^2+4*K_a_T*gamma_m_H*gamma_m_H*C_mf_a)^(1/2))...
72     /(2*gamma_m_H*gamma_m_H)-C_mf_H;
73 fu(12)=A_I*(I_m^(1/2)/(1+I_m^(1/2))- 0.3*I_m)+log10(gamma_m_H);%"Davies"
74 fu(13)=0.5*(2*C_mf_s+2*C_mf_H)-I_m;
75
76 fu(14)=((C_md_s-C_i_s)*(12.625*10^(-11)*((0.789*C_i_s+0.211*C_md_s)^4 ...
77 /(J_w*(0.789*C_i_s+0.211*C_md_s)+ J_s)+(0.211*C_i_s+0.789*C_md_s)^4 ...
78 /(J_w*(0.211*C_i_s+0.789*C_md_s)+ J_s ))...
79 -3.885*10^(-9)*((0.789*C_i_s+0.211*C_md_s)^3 ...
80 /(J_w*(0.789*C_i_s+0.211*C_md_s)+ J_s)+(0.211*C_i_s+0.789*C_md_s)^3 ...
81 /(J_w*(0.211*C_i_s+0.789*C_md_s)+ J_s ))...
82 +6.975*10^(-8)*((0.789*C_i_s+0.211*C_md_s)^2 ...
83 /(J_w*(0.789*C_i_s+0.211*C_md_s)+ J_s)+(0.211*C_i_s+0.789*C_md_s)^2 ...
84 /(J_w*(0.211*C_i_s+0.789*C_md_s)+ J_s )...
85 -4.03*10^(-7)*((0.789*C_i_s+0.211*C_md_s)...
86 /(J_w*(0.789*C_i_s+0.211*C_md_s)+ J_s)+(0.211*C_i_s+0.789*C_md_s)...
87 /(J_w*(0.211*C_i_s+0.789*C_md_s)+ J_s))...
88 +0.53*10^(-5)*(1/(J_w*(0.789*C_i_s+0.211*C_md_s)+J_s)...
89 +1/(J_w*(0.211*C_i_s+0.789*C_md_s)+ J_s))))-S;
90
91 fu(15)=((C_d_s-C_md_s)*(12.625*10^(-11)*((0.789*C_md_s+0.211*C_d_s)^4 ...
92 /(J_w*(0.789*C_md_s+0.211*C_d_s)+ J_s)+(0.211*C_md_s+0.789*C_d_s)^4 ...
93 /(J_w*(0.211*C_md_s+0.789*C_d_s)+ J_s ))...
94 -3.885*10^(-9)*((0.789*C_md_s+0.211*C_d_s)^3 ...
95 /(J_w*(0.789*C_md_s+0.211*C_d_s)+ J_s)+(0.211*C_md_s+0.789*C_d_s)^3 ...
96 /(J_w*(0.211*C_md_s+0.789*C_d_s)+ J_s ))...
97 +6.975*10^(-8)*((0.789*C_md_s+0.211*C_d_s)^2 ...
98 /(J_w*(0.789*C_md_s+0.211*C_d_s)+ J_s)+(0.211*C_md_s+0.789*C_d_s)^2 ...
99 /(J_w*(0.211*C_md_s+0.789*C_d_s)+ J_s ))...
100 -4.03*10^(-7)*((0.789*C_md_s+0.211*C_d_s)...
101 /(J_w*(0.789*C_md_s+0.211*C_d_s)+ J_s)+(0.211*C_md_s+0.789*C_d_s)...
102 /(J_w*(0.211*C_md_s+0.789*C_d_s)+ J_s))...
103 +0.53*10^(-5)*(1/(J_w*(0.789*C_md_s+0.211*C_d_s)+J_s)...
104 +1/(J_w*(0.211*C_md_s+0.789*C_d_s)+ J_s))))-D_d_s/k_d;
105
106 fu(16)=Sh_d*D_d_s/d_h-k_d;
107 fu(17)=5.05*10^(-11)*C_d_s^4-2.59*10^(-9)*C_d_s^3+4.65*10^(-8)*C_d_s^2-4.03*10^(-7
*C_d_s+1.06*10^(-5)-D_d_s;
```

```
108 fu(18)=L*v_d*den_d/mul_d-Re_d;
109 fu(19)=(f_2+f)/2/W/D-v_d;
110 fu(20)=f+J_w*A_m-f_2;
111
112 %↵
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^↵
^^^^^^^^^^^^^^^^^
113 fu(21)=0.0538*C_d_s+0.4159+5.0095/C_d_s-mul_d/C_d_s;%Math Lab
114 fu(22)=37.0166*C_d_s+997.2911-den_d;%Math lab
115 %↵
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^↵
^^^^^^^^^^^^^^^^^^^^^^
116 fu(23)= 0.04*Re_d^(3/4)*Sc_d^(1/3)-Sh_d;
117 fu(24)= mul_d/den_d/D_d_s-Sc_d;
118 fu(25)=(0.5*(C_f_a-C_mf_a)*((D_1_T-D_HA_T)/...
119     (sqrt(1+4*K*(0.789*C_mf_a+0.211*C_f_a))*(J_w*(0.789*C_mf_a+0.211*C_f_a)-J_a))...
120     +(D_1_T-D_HA_T)/(sqrt(1+4*K*(0.211*C_mf_a+0.789*C_f_a))*(J_w*(0.211*C_mf_a...
121     +0.789*C_f_a)-J_a))+ D_HA_T/(J_w*(0.789*C_mf_a+0.211*C_f_a)-J_a)...
122     +D_HA_T/((J_w*(0.211*C_mf_a+0.789*C_f_a)-J_a))))+D_f_a/k_f_a;
123 fu(26)=Sh_f_a*D_f_a/d_h-k_f_a;
124 fu(27)=(D_1_T/(2*K*C_f_a)*(-1+sqrt(1+4*K*C_f_a))+D_HA_T/...
125     (4*K*C_f_a)*(-1+sqrt(1+4*K*C_f_a))^2)-D_f_a;
126 fu(28)=L*v_f*den_f/mul_f -Re_f;
127 fu(29)=(f+f_1)/W/D/2-v_f;
128 fu(30)=f-J_w*A_m-f_1;
129 fu(31)= 0.04*Re_f^(3/4)*Sc_f_a^(1/3)-Sh_f_a;
130 fu(32)= mul_f/den_f/D_f_a-Sc_f_a;
131
132 fu(33)=((C_f_s-C_mf_s)*(12.625*10^(-11)*((0.789*C_mf_s+0.211*C_f_s)^4 ...
133 /(J_w*(0.789*C_mf_s+0.211*C_f_s)+ J_s)+(0.211*C_mf_s+0.789*C_f_s)^4 ...
134 /(J_w*(0.211*C_mf_s+0.789*C_f_s)+ J_s ))...
135 -3.885*10^(-9)*((0.789*C_mf_s+0.211*C_f_s)^3 ...
136 /(J_w*(0.789*C_mf_s+0.211*C_f_s)+ J_s)+(0.211*C_mf_s+0.789*C_f_s)^3 ...
137 /(J_w*(0.211*C_mf_s+0.789*C_f_s)+ J_s )...
138 +6.975*10^(-8)*((0.789*C_mf_s+0.211*C_f_s)^2 ...
139 /(J_w*(0.789*C_mf_s+0.211*C_f_s)+ J_s)+(0.211*C_mf_s+0.789*C_f_s)^2 ...
140 /(J_w*(0.211*C_mf_s+0.789*C_f_s)+ J_s ))...
141 -4.03*10^(-7)*((0.789*C_mf_s+0.211*C_f_s)...
142 /(J_w*(0.789*C_mf_s+0.211*C_f_s)+ J_s)+(0.211*C_mf_s+0.789*C_f_s)...
143 /(J_w*(0.211*C_mf_s+0.789*C_f_s)+ J_s))...
144 +0.53*10^(-5)*(1/(J_w*(0.789*C_mf_s+0.211*C_f_s)+J_s)...
145 +1/(J_w*(0.211*C_mf_s+0.789*C_f_s)+ J_s))))+D_f_s/k_f_s;
146
147 fu(34)=Sh_f_s*D_f_s/d_h-k_f_s;
148 fu(35)=5.05*10^(-11)*C_f_s^4-2.59*10^(-9)*C_f_s^3+4.65*10^(-8)*C_f_s^2-4.03*10^(-7)↵
*C_f_s+1.06*10^(-5)-D_f_s;
149
150 %Mole balance Equation
151 fu(36)=C_f_a*(V_f_0+f_1*t-f*t)-(V_f_0*C_f_a_0+C_f_ai*f_1*t-C_f_a*f*t);%new
152 fu(37)=C_f_si*(V_f_0+f_1*t-f*t)-C_f_s*f_1*t+C_f_si*f*t;%old
153 fu(38)=C_f_a*f-J_a*A_m-C_f_ai*f_1;%new
154 fu(39)=C_f_si*f+J_s*A_m-C_f_s*f_1;%old
155 fu(40)=C_d_s_0*V_d_0+C_d_si*f_2*t-C_d_s*f*t-C_d_s*(V_d_0+f_2*t-f*t);%new
156 fu(41)=C_d_a*f_2*t-C_d_ai*f*t-C_d_ai*(V_d_0+f_2*t-f*t);%old
157 fu(42)=C_d_s*f-J_s*A_m-C_d_si*f_2;%new
158 fu(43)=C_d_ai*f+J_a*A_m-C_d_a*f_2;%old
```

```matlab
159
160 %additional Equation
161 fu(44)= 0.04*Re_f^(3/4)*Sc_f_s^(1/3)-Sh_f_s;
162 fu(45)= mul_f/den_f/D_f_s-Sc_f_s;
163 fu(46)=J_w*A_m*t*den_d/1000-w;
164
165 %pH Equation..
166 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
167 fu(47)= pH_f+log10(gamma_f_H*C_f_H);
168 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
169 fu(48)=K_a_T*(C_f_a-C_f_A)/gamma_f_H/gamma_f_H/C_f_H-C_f_A;
170 fu(49)=A_I*(I_f^(1/2)/(1+I_f^(1/2))-0.3*I_f)+log10(gamma_f_H);%Davies;
171 %ionic streanght in feed solution%%%%%%%%%%%%%%%%%%%%
172 fu(50)=0.5*(2*C_f_si+C_f_H+K_w_T/C_f_H/gamma_f_H/gamma_f_H+C_f_A+C_f_HCO3c)- I_f;
173 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
174 %Charge Balance Equation %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
175 fu(51)=(C_f_HCO3c+C_f_A+K_w_T/C_f_H/gamma_f_H/gamma_f_H)-C_f_H;
176 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
177 fu(52)=P_M*C_d_co2-J_co2;
178 fu(53)=C_f_co2i*(V_f_0+f_1*t-f*t)-(C_f_co2*f_1*t-C_f_co2i*f*t);
179 fu(54)=C_f_co2i*f*t+J_co2*A_m*t-C_f_co2*f_1*t;
180 fu(55)=10^-3.45/gamma_f_H/gamma_f_H/C_f_H*(C_f_co2-C_f_HCO3c)-C_f_HCO3c;
181 % Draw solution
182 fu(56)=gamma_d_H*gamma_d_H*C_d_H*C_d_HCO3/K_CO2_T+C_d_HCO3-C_d_co2;%******
183 fu(57)=(-K_CO2_T+sqrt(K_CO2_T^2+4*K_CO2_T*gamma_d_H*gamma_d_H*C_d_co2a))/↙
(2*gamma_d_H*gamma_d_H)-C_d_HCO3;
184 %ionic streanght in draw solution%%%%%%%%%%%%%%%%%%%%
185 fu(58)=0.5*(2*C_d_si+C_d_H+K_w_T/C_d_H/gamma_d_H/gamma_d_H+C_d_A+C_d_HCO3)- I_d;
186 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
187 %Charge balance at draw solution%%%%%%%%%%%%%%%%%%%%%%
188 fu(59)=(C_d_HCO3+C_d_A+K_w_T/C_f_H/gamma_d_H/gamma_d_H)-C_d_H;
189 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
190 fu(60)=K_a_T*(C_d_a-C_d_A)/gamma_d_H/gamma_d_H/C_d_H-C_d_A;
191 fu(61)=10^-(A_I*(I_d^(1/2)/(1+I_d^(1/2))-0.3*I_d))-gamma_d_H;%Davies;
192 fu(62)=10^-3.408*K_H/10^(((33.5-0.109*(T-273)+0.0014*(T-273)^2)*I_d-(1.5+0.015*(T-↙
273)+0.004*(T-273)^2)*I_d^2))...
193     /T)-C_d_co2a;
194
195
```

G.11 Valeric acid as feed solution and NaCl as draw solution

```
 1 clear
 2 clc
 3 [g]=[1.545e-03    2.1246e+01    2.708e-01    1.934e-05 4.535e-01    1.275e-04↙
1.471e-11    1.020e-02    8.299e-09...
 4 8.230e-09    2.6086e-06    7.898e-01    4.535e-01 9.995e-01    3.833e-04    9.745↙
e-01    5.108e-04    9.893e-01...
 5 9.545e-06    1.4718e-01    6.754e+02    5.550e+02 2.612e+04    1.503e+02    1.034↙
e+03    5.479e+00    1.581e+00...
 6 1.060e-05    1.5607e-01    1.000e-02    6.452e+02 2.545e+04    1.502e+02    1.579↙
e+00    5.129e+02    9.258e-06...
 7 1.426e-01    8.3963e-06    1.001e-02    3.251e-06 6.997e-09    9.991e-01    3.432↙
e+00    9.778e-01    3.783e-04...
 8 3.815e-04    5.4140e+00    6.748e+02    5.870e+02 2.685e-04    3.783e-04    1.279↙
e-08    4.951e-09    6.333e-09...
 9 2.947e-08    1.1786e-05    2.601e-06    9.991e-01 7.385e-09    8.228e-09    7.306↙
e-01    1.176e-05];
10 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
11 ts=0.4:0.006:1800;
12 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
13
14 for ti=1:numel(ts);
15 t=ts(ti);
16 options = optimset('Algorithm','Levenberg-Marquardt','TolFun',1*10^-15,'TolX',1*10^-↙
15 ...
17     ,'Maxiter',25,'Maxfun',15000);
18
19 [r,fval,exitflag,output] = fsolve(@(z)Valeric_NaCl(z,t),[g],options)
20 [g]=[r];
21 j(ti)=r(1,1);
22 p(ti)=r(1,43);%pH_f
23 c(ti)=r(1,30);%C_f_a
24 d(ti)=r(1,50);%w
25 T(ti)=t;
26 d1(ti)=r(1,55);%J_co2
27 d2(ti)=r(1,52);%C_f_co2
28 d3(ti)=r(1,54);%C_f_HCO3C
29 d4(ti)=r(1,56);%C_d_co2
30 d5(ti)=r(1,57);%C_d_HCO3
31 d6(ti)=-log10(r(1,12)*r(1,11));%pH_d
32 d7(ti)=(1-r(1,43)/r(1,32))*100;% %Rejection
33 d8(ti)=r(1,32)/0.01;%Concentration Performance
34 %for Plotting
35 [A(ti)]=t;
36 [B(ti)]=d(ti);
37 [C(ti)]=p(ti);
38 [M(ti)]=d7(ti);
39 [N(ti)]=j(ti);
40 [K5(ti)]=d8(ti);
41 disp(ti)
42 fprintf('\n     Time(min)        J_w(L/dm2/min)            pH          Cf,a(mole/l)↙
W(kg)\n');
43 fprintf('\n%11.2f%20.4e%19.4f%19.4e%19.4e\n' ,T(ti),j(ti),p(ti),c(ti),d(ti));
44 format short e
45 %Result of Best guess
46 fprintf('\n%8.3e%15.4e%13.3e%14.3e%11.3e%13.3e%13.3e%13.3e%13.3e%\n'...
47     ,r(1,1),r(1,2),r(1,3),r(1,4),r(1,5),r(1,6),r(1,7),r(1,8),r(1,9));
```

```
48
49  fprintf('\n%8.3e%15.4e%13.3e%14.3e%11.3e%13.3e%13.3e%13.3e%\n'...
50     ,r(1,10),r(1,11),r(1,12),r(1,13),r(1,14),r(1,15),r(1,16),r(1,17),r(1,18));
51
52  fprintf('\n%8.3e%15.4e%13.3e%14.3e%11.3e%13.3e%13.3e%13.3e%\n'...
53     ,r(1,19),r(1,20),r(1,21),r(1,22),r(1,23),r(1,24),r(1,25),r(1,26),r(1,27));
54
55  fprintf('\n%8.3e%15.4e%13.3e%14.3e%11.3e%13.3e%13.3e%13.3e%\n'...
56     ,r(1,28),r(1,29),r(1,30),r(1,31),r(1,32),r(1,33),r(1,34),r(1,35),r(1,36));
57
58  fprintf('\n%8.3e%15.4e%13.3e%14.3e%11.3e%13.3e%13.3e%13.3e%13.3e%13.3e%\n'...
59     ,r(1,37),r(1,38),r(1,39),r(1,40),r(1,41),r(1,42),r(1,43),r(1,44),r(1,45));
60
61  fprintf('\n%8.3e%15.4e%13.3e%14.3e%11.3e%13.3e%13.3e%13.3e%13.3e%\n'...
62     ,r(1,46),r(1,47),r(1,48),r(1,49),r(1,50),r(1,51),r(1,52),r(1,53),r(1,54));
63
64  fprintf('\n%8.3e%15.4e%13.3e%14.3e%11.3e%13.3e%13.3e%13.3e%\n'...
65     ,r(1,55),r(1,56),r(1,57),r(1,58),r(1,59),r(1,60),r(1,61),r(1,62));
66  end
67
68  %Time Interval%%%%%%%%%%%%%%%%%%%%%%%%%%%%
69  Ai=0:30:1800;
70  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
71  format short
72  Ci=spline(A,C,Ai);
73  format short e
74  Bi=spline(A,B,Ai);
75  %Experiment Value W and pH%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
76  F=[ 0     0.0270    0.0430    0.0570    0.0710    0.0830    0.0980    0.1100    0.1250
0.1390    0.1490    0.1650    0.1760...
77  0.1880    0.2030    0.2140    0.2280    0.2380    0.2530    0.2630    0.2750
0.2870    0.2970    0.3110    0.3210    0.3300...
78  0.3440    0.3530    0.3620    0.3720    0.3860    0.3950    0.4030    0.4150
0.4250    0.4340    0.4460    0.4560    0.4670...
79  0.4750    0.4790    0.4840    0.4910    0.5000    0.5130    0.5220    0.5310
0.5390    0.5460    0.5530    0.5600    0.5660...
80  0.5700    0.5750    0.5820    0.5900    0.5970    0.6060    0.6140    0.6200
0.6200];
81
82  G=[ 3.4090    3.4100    3.4090    3.4090    3.4040    3.3900    3.3900    3.3790
3.3780    3.3730    3.3610    3.3560    3.3490...
83  3.3410    3.3390    3.3270    3.3180    3.3150    3.3040    3.3010    3.2930
3.2880    3.2770    3.2640    3.2630    3.2540...
84  3.2460    3.2390    3.2310    3.2230    3.2110    3.2070    3.1980    3.1880
3.1840    3.1750    3.1610    3.1550    3.1480...
85  3.1450    3.1330    3.1240    3.1170    3.1100    3.0960    3.0920    3.0890
3.0840    3.0780    3.0700    3.0630    3.0530...
86  3.0410    3.0340    3.0280    3.0210    3.0150    3.0080    3.0030    3.0020
2.9990];
87  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
88
89  F_m=mean(F);
90  G_m=mean(G);
91
92  sum1=0;sum2=0;
93  sum3=0;sum4=0;
```

```matlab
94 %%% Number of Data %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
95 for k=1:1:61
96 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
97 sum1=(Bi(1,k)-F(1,k))^2+sum1;
98 sum2=(Bi(1,k)-F_m)^2+sum2;
99 sum3=(Ci(1,k)-G(1,k))^2+sum3;
100 sum4=(Ci(1,k)-G_m)^2+sum4;
101 end
102
103 format short
104 R2_W=1-sum1/sum2;
105 R2_pH=1-sum3/sum4;
106 MSE_W=sum1/k;
107 MSE_pH=sum3/k;
108 RMSE_W=sqrt(MSE_W)
109 RMSE_pH=sqrt(MSE_pH)
110 SEP_W=RMSE_W/F_m*100
111 SEP_pH=RMSE_pH/G_m*100
112
113 % Plotting
114 subplot(2,3,1)
115 plot([A]/60,[B],'-g',[Ai]/60,[F],'xr')
116 xlabel('Time (hr)')
117 ylabel('Weight (Kg)')
118 title('Weight Change as a Function of Time')
119 legend('Model','Experiment',2)
120
121 subplot(2,3,2)
122 plot([A]/60,[C],'-g',[Ai]/60,[G],'xr')
123 xlabel('Time (hr)')
124 ylabel('pH')
125 title('pH as a Function of Time')
126 legend('Model','Experiment',1)
127
128 subplot(2,3,3)
129 plot([A]/60,[M],'-g')
130 xlabel('Time (hr)')
131 ylabel('%Rejection')
132 title('%Rejection as a Function of Time')
133 legend('Model',1)
134
135 subplot(2,3,4)
136 plot([N],[M],'-g')
137 xlabel('Jw (L/dm2/min)')
138 ylabel('%Rejection')
139 title('%Rejection as a Function of Jw')
140 legend('Model',1)
141
142 subplot(2,3,5)
143 plot([A]/60,[K5],'-g')
144 xlabel('Time (hr)')
145 ylabel('Cfa/Cf0')
146 title('Cfa/Cf0 as a Function of Time')
147 legend('Model',1)
148
149 subplot(2,3,6)
```

```
150 plot([N],[K5],'-g')
151 xlabel('Jw (L/dm2/min)')
152 ylabel('Cfa/Cf0')
153 title('Cfa/Cf0 as a Function of Jw')
154 legend('Model',1)
155
156 fprintf('\n     \n')
157 fprintf('\n     \n')
158 fprintf('\n CONCLUSION     \n')
159 fprintf('\n     R2_W          MSE_W                    R2_pH          MSE_pH\n');
160 fprintf('\n%11.3f%20.4e%19.4f%19.4e%\n',R2_W,MSE_W,R2_pH,MSE_pH)
161 fprintf('\n     \n')
162
163 fprintf('\n    Time(min)       J_w(L/dm2/min)              pH              Cf,a(mole/l)↙
W(kg)\n');for x=5:5:200000000;
164 z=x*100;
165 fprintf('\n%11.4f %20.4e%19.4f%19.4e%19.4e%\n',T(z),j(z),p(z),c(z),d(z));
166
167 end
168
169
```

```
1
2 function fu=Valeric_NaCl(z,t)
3
4 %Independent Variable
5 V_f_0=1;C_d_s_0=1;C_f_a_0=10*10^(-3);V_d_0=0.5;
6 %Fixed Variable
7 T=301;f=1.58;
8 % T=300..normal f=1.58...normal
9 L=0.9207;D=0.023;W=0.4572;A_m=42*10^(-2);d_h=0.0438;
10 %Constant Variable
11 ↵
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%↵
%
12 B_a=0.491/60/100;% B of Valeric Acid
13 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
14 B_s=0.256/60/100;
15 A=0.442/60/100;
16 S=500*10^-5;
17 K_a_T=10^-(921.38/T-1.8574+0.012105*T);%Valeric Acid
18 D_A_298=5.226*10^(-6);D_HA_298=4.902*10^(-6);%Valeric Physicocmical Property
19 D_H_298=5.587*10^(-5);
20 ↵
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%↵
%5
21 o_s=0.936;
22 den_f=999.65+2.0438/10*(T-273)-6.174*10^-2*(T-273)^1.5;
23 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
24 n_s=2;o_a=1;R=0.08314;
25 %Temperature Correction
26 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
27 K_CO2_T=10^-(3404.71/T-14.8435+0.032786*T);
28 K_w_T=10^(-4470.99/T+6.0875-0.017060*T);
29 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
30 E=2727.586+0.6224107*T-466.9151*log(T)-52000.87/T;
31 d=1-(((T-273)-3.9863)^2*((T-273)+288.9414))/(508929.2*((T-273)+68.12963))...
32 +0.011445*exp((-374.3)/(T-273));
33 A_I=(1.82483*10^6*d^0.5)/(E*T)^1.5;
34 K=1/K_a_T;
35 visco_298=((298-273)+246)/((0.05594*(298-273)+5.2842)*(298-273)+137.37);
36 visco_T=((T-273)+246)/((0.05594*(T-273)+5.2842)*(T-273)+137.37);
37 D_H_T=D_H_298*T/298*visco_298/visco_T ;
38 D_A_T=D_A_298*T/298*visco_298/visco_T;
39 D_HA_T=D_HA_298*T/298*visco_298/visco_T;
40 D_1_T=2/(1/D_H_T +1/D_A_T);
41 K_H=0.034*exp(-2400*(1/T-1/298));%************************
42 %For carbonic acid
43 P_M=0.0025;
44 %Unknown Variable
45 J_w=z(1);P_i=z(2);P_m=z(3);J_s=z(4);C_i_s=z(5);
46 C_mf_s=z(6);J_a=z(7);C_mf_a=z(8);C_i_a=z(9);C_d_a=z(10);
47 C_d_H=z(11);gamma_d_H=z(12);I_i=z(13);C_d_s=z(14);
48 C_mf_H=z(15);gamma_m_H=z(16);I_m=z(17);C_md_s=z(18);
49 D_d_s=z(19);k_d=z(20);Sh_d=z(21);Sc_d=z(22);Re_d=z(23);
50 v_d=z(24);den_d=z(25);mul_d=z(26);f_2=z(27);D_f_a=z(28);
51 k_f_a=z(29);C_f_a=z(30);Sh_f_a=z(31);Re_f=z(32);v_f=z(33);
52 f_1=z(34);Sc_f_a=z(35);D_f_s=z(36);k_f_s=z(37);C_f_s=z(38);
```

```
53 C_f_ai=z(39);C_f_si=z(40);C_d_ai=z(41);C_d_si=z(42);pH_f=z(43);
54 gamma_f_H=z(44);C_f_A=z(45);I_f=z(46);mul_f=z(47);
55 Sh_f_s=z(48);Sc_f_s=z(49);w=z(50);C_f_H=z(51);
56 C_f_co2=z(52);C_f_co2i=z(53);C_f_HCO3c=z(54);
57 J_co2=z(55);C_d_co2=z(56);C_d_HCO3=z(57);I_d=z(58);C_d_A=z(59);
58 C_i_H=z(60);gamma_i_H=z(61);C_d_co2a=z(62);
59 %Math Model
60 fu(1)=A*(P_i-P_m )-J_w;
61 fu(2)= B_s *(C_i_s-C_mf_s)-J_s;
62 fu(3)= B_a*(C_mf_a-C_i_a)-J_a;
63 fu(4)=C_d_a-C_i_a;
64 fu(5)=J_a/J_w-C_d_a;
65 fu(6)= o_s*n_s*C_i_s*R*T+o_a*(C_i_a+C_i_H)*R*T-P_i;
66 fu(7)=(-(K_a_T)+((K_a_T)^2+4*K_a_T*gamma_i_H*gamma_i_H*C_d_a)^(1/2))...
67     /(2*gamma_i_H*gamma_i_H)-C_i_H;
68 fu(8)=A_I*(I_i^(1/2)/(1+I_i^(1/2))- 0.3*I_i)+log10(gamma_i_H);%"Davies"
69 fu(9)=0.5*(2*C_i_s+2*C_i_H)- I_i;
70 fu(10)=o_s*n_s*C_mf_s*R*T+o_a*(C_mf_a+C_mf_H)*R*T-P_m;
71 fu(11)=(-(K_a_T)+((K_a_T)^2+4*K_a_T*gamma_m_H*gamma_m_H*C_mf_a)^(1/2))...
72     /(2*gamma_m_H*gamma_m_H)-C_mf_H;
73 fu(12)=A_I*(I_m^(1/2)/(1+I_m^(1/2))- 0.3*I_m)+log10(gamma_m_H);%"Davies"
74 fu(13)=0.5*(2*C_mf_s+2*C_mf_H)-I_m;
75
76 fu(14)=((C_md_s-C_i_s)*(12.625*10^(-11)*((0.789*C_i_s+0.211*C_md_s)^4 ...
77 /(J_w*(0.789*C_i_s+0.211*C_md_s)+ J_s)+(0.211*C_i_s+0.789*C_md_s)^4 ...
78 /(J_w*(0.211*C_i_s+0.789*C_md_s)+ J_s ))...
79 -3.885*10^(-9)*((0.789*C_i_s+0.211*C_md_s)^3 ...
80 /(J_w*(0.789*C_i_s+0.211*C_md_s)+ J_s)+(0.211*C_i_s+0.789*C_md_s)^3 ...
81 /(J_w*(0.211*C_i_s+0.789*C_md_s)+ J_s ))...
82 +6.975*10^(-8)*((0.789*C_i_s+0.211*C_md_s)^2 ...
83 /(J_w*(0.789*C_i_s+0.211*C_md_s)+ J_s)+(0.211*C_i_s+0.789*C_md_s)^2 ...
84 /(J_w*(0.211*C_i_s+0.789*C_md_s)+ J_s ))...
85 -4.03*10^(-7)*((0.789*C_i_s+0.211*C_md_s)...
86 /(J_w*(0.789*C_i_s+0.211*C_md_s)+ J_s)+(0.211*C_i_s+0.789*C_md_s)...
87 /(J_w*(0.211*C_i_s+0.789*C_md_s)+ J_s))...
88 +0.53*10^(-5)*(1/(J_w*(0.789*C_i_s+0.211*C_md_s)+J_s)...
89 +1/(J_w*(0.211*C_i_s+0.789*C_md_s)+ J_s))))-S;
90
91 fu(15)=((C_d_s-C_md_s)*(12.625*10^(-11)*((0.789*C_md_s+0.211*C_d_s)^4 ...
92 /(J_w*(0.789*C_md_s+0.211*C_d_s)+ J_s)+(0.211*C_md_s+0.789*C_d_s)^4 ...
93 /(J_w*(0.211*C_md_s+0.789*C_d_s)+ J_s ))...
94 -3.885*10^(-9)*((0.789*C_md_s+0.211*C_d_s)^3 ...
95 /(J_w*(0.789*C_md_s+0.211*C_d_s)+ J_s)+(0.211*C_md_s+0.789*C_d_s)^3 ...
96 /(J_w*(0.211*C_md_s+0.789*C_d_s)+ J_s ))...
97 +6.975*10^(-8)*((0.789*C_md_s+0.211*C_d_s)^2 ...
98 /(J_w*(0.789*C_md_s+0.211*C_d_s)+ J_s)+(0.211*C_md_s+0.789*C_d_s)^2 ...
99 /(J_w*(0.211*C_md_s+0.789*C_d_s)+ J_s ))...
100 -4.03*10^(-7)*((0.789*C_md_s+0.211*C_d_s)...
101 /(J_w*(0.789*C_md_s+0.211*C_d_s)+ J_s)+(0.211*C_md_s+0.789*C_d_s)...
102 /(J_w*(0.211*C_md_s+0.789*C_d_s)+ J_s))...
103 +0.53*10^(-5)*(1/(J_w*(0.789*C_md_s+0.211*C_d_s)+J_s)...
104 +1/(J_w*(0.211*C_md_s+0.789*C_d_s)+ J_s))))-D_d_s/k_d;
105
106 fu(16)=Sh_d*D_d_s/d_h-k_d;
107 fu(17)=5.05*10^(-11)*C_d_s^4-2.59*10^(-9)*C_d_s^3+4.65*10^(-8)*C_d_s^2-4.03*10^(-7)
*C_d_s+1.06*10^(-5)-D_d_s;
```

```
108 fu(18)=L*v_d*den_d/mul_d-Re_d;
109 fu(19)=(f_2+f)/2/W/D-v_d;
110 fu(20)=f+J_w*A_m-f_2;
111
112 %↵
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^↵
^^^^^^^^^^^^^^^^^^
113 fu(21)=0.0538*C_d_s+0.4159+5.0095/C_d_s-mul_d/C_d_s;%Math Lab
114 fu(22)=37.0166*C_d_s+997.2911-den_d;%Math lab
115 %↵
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^↵
^^^^^^^^^^^^^^^^^^^^^^
116 fu(23)= 0.04*Re_d^(3/4)*Sc_d^(1/3)-Sh_d;
117 fu(24)= mul_d/den_d/D_d_s-Sc_d;
118 fu(25)=(0.5*(C_f_a-C_mf_a)*((D_1_T-D_HA_T)/...
119     (sqrt(1+4*K*(0.789*C_mf_a+0.211*C_f_a))*(J_w*(0.789*C_mf_a+0.211*C_f_a)-J_a))...
120     +(D_1_T-D_HA_T)/(sqrt(1+4*K*(0.211*C_mf_a+0.789*C_f_a))*(J_w*(0.211*C_mf_a...
121     +0.789*C_f_a)-J_a))+ D_HA_T/(J_w*(0.789*C_mf_a+0.211*C_f_a)-J_a)...
122     +D_HA_T/((J_w*(0.211*C_mf_a+0.789*C_f_a)-J_a))))+D_f_a/k_f_a;
123 fu(26)=Sh_f_a*D_f_a/d_h-k_f_a;
124 fu(27)=(D_1_T/(2*K*C_f_a)*(-1+sqrt(1+4*K*C_f_a))+D_HA_T/...
125     (4*K*C_f_a)*(-1+sqrt(1+4*K*C_f_a))^2)-D_f_a;
126 fu(28)=L*v_f*den_f/mul_f -Re_f;
127 fu(29)=(f+f_1)/W/D/2-v_f;
128 fu(30)=f-J_w*A_m-f_1;
129 fu(31)= 0.04*Re_f^(3/4)*Sc_f_a^(1/3)-Sh_f_a;
130 fu(32)= mul_f/den_f/D_f_a-Sc_f_a;
131
132 fu(33)=((C_f_s-C_mf_s)*(12.625*10^(-11)*((0.789*C_mf_s+0.211*C_f_s)^4 ...
133 /(J_w*(0.789*C_mf_s+0.211*C_f_s)+ J_s)+(0.211*C_mf_s+0.789*C_f_s)^4 ...
134 /(J_w*(0.211*C_mf_s+0.789*C_f_s)+ J_s ))...
135 -3.885*10^(-9)*((0.789*C_mf_s+0.211*C_f_s)^3 ...
136 /(J_w*(0.789*C_mf_s+0.211*C_f_s)+ J_s)+(0.211*C_mf_s+0.789*C_f_s)^3 ...
137 /(J_w*(0.211*C_mf_s+0.789*C_f_s)+ J_s ))...
138 +6.975*10^(-8)*((0.789*C_mf_s+0.211*C_f_s)^2 ...
139 /(J_w*(0.789*C_mf_s+0.211*C_f_s)+ J_s)+(0.211*C_mf_s+0.789*C_f_s)^2 ...
140 /(J_w*(0.211*C_mf_s+0.789*C_f_s)+ J_s ))...
141 -4.03*10^(-7)*((0.789*C_mf_s+0.211*C_f_s)...
142 /(J_w*(0.789*C_mf_s+0.211*C_f_s)+ J_s)+(0.211*C_mf_s+0.789*C_f_s)...
143 /(J_w*(0.211*C_mf_s+0.789*C_f_s)+ J_s))...
144 +0.53*10^(-5)*(1/(J_w*(0.789*C_mf_s+0.211*C_f_s)+J_s)...
145 +1/(J_w*(0.211*C_mf_s+0.789*C_f_s)+ J_s))))+D_f_s/k_f_s;
146
147 fu(34)=Sh_f_s*D_f_s/d_h-k_f_s;
148 fu(35)=5.05*10^(-11)*C_f_s^4-2.59*10^(-9)*C_f_s^3+4.65*10^(-8)*C_f_s^2-4.03*10^(-7)↵
*C_f_s+1.06*10^(-5)-D_f_s;
149
150 %Mole balance Equation
151 fu(36)=C_f_a*(V_f_0+f_1*t-f*t)-(V_f_0*C_f_a_0+C_f_ai*f_1*t-C_f_a*f*t);%new
152 fu(37)=C_f_si*(V_f_0+f_1*t-f*t)-C_f_s*f_1*t+C_f_si*f*t;%old
153 fu(38)=C_f_a*f-J_a*A_m-C_f_ai*f_1;%new
154 fu(39)=C_f_si*f+J_s*A_m-C_f_s*f_1;%old
155 fu(40)=C_d_s_0*V_d_0+C_d_si*f_2*t-C_d_s*f*t-C_d_s*(V_d_0+f_2*t-f*t);%new
156 fu(41)=C_d_a*f_2*t-C_d_ai*f*t-C_d_ai*(V_d_0+f_2*t-f*t);%old
157 fu(42)=C_d_s*f-J_s*A_m-C_d_si*f_2;%new
158 fu(43)=C_d_ai*f+J_a*A_m-C_d_a*f_2;%old
```

```
159
160 %additional Equation
161 fu(44)= 0.04*Re_f^(3/4)*Sc_f_s^(1/3)-Sh_f_s;
162 fu(45)= mul_f/den_f/D_f_s-Sc_f_s;
163 fu(46)=J_w*A_m*t*den_d/1000-w;
164
165 %pH Equation..
166 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
167 fu(47)= pH_f+log10(gamma_f_H*C_f_H);
168 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
169 fu(48)=K_a_T*(C_f_a-C_f_A)/gamma_f_H/gamma_f_H/C_f_H-C_f_A;
170 fu(49)=A_I*(I_f^(1/2)/(1+I_f^(1/2))-0.3*I_f)+log10(gamma_f_H);%Davies;
171 %ionic streanght in feed solution%%%%%%%%%%%%%%%%%%%%
172 fu(50)=0.5*(2*C_f_si+C_f_H+K_w_T/C_f_H/gamma_f_H/gamma_f_H+C_f_A+C_f_HCO3c)- I_f;
173 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
174 %Charge Balance Equation %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
175 fu(51)=(C_f_HCO3c+C_f_A+K_w_T/C_f_H/gamma_f_H/gamma_f_H)-C_f_H;
176 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
177 fu(52)=P_M*C_d_co2-J_co2;
178 fu(53)=C_f_co2i*(V_f_0+f_1*t-f*t)-(C_f_co2*f_1*t-C_f_co2i*f*t);
179 fu(54)=C_f_co2i*f*t+J_co2*A_m*t-C_f_co2*f_1*t;
180 fu(55)=10^-3.45/gamma_f_H/gamma_f_H/C_f_H*(C_f_co2-C_f_HCO3c)-C_f_HCO3c;
181 % Draw solution
182 fu(56)=gamma_d_H*gamma_d_H*C_d_H*C_d_HCO3/K_CO2_T+C_d_HCO3-C_d_co2;%******
183 fu(57)=(-K_CO2_T+sqrt(K_CO2_T^2+4*K_CO2_T*gamma_d_H*gamma_d_H*C_d_co2a))/↙
(2*gamma_d_H*gamma_d_H)-C_d_HCO3;
184 %ionic streanght in draw solution%%%%%%%%%%%%%%%%%%%%
185 fu(58)=0.5*(2*C_d_si+C_d_H+K_w_T/C_d_H/gamma_d_H/gamma_d_H+C_d_A+C_d_HCO3)- I_d;
186 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
187 %Charge balance at draw solution%%%%%%%%%%%%%%%%%%%%
188 fu(59)=(C_d_HCO3+C_d_A+K_w_T/C_f_H/gamma_d_H/gamma_d_H)-C_d_H;
189 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
190 fu(60)=K_a_T*(C_d_a-C_d_A)/gamma_d_H/gamma_d_H/C_d_H-C_d_A;
191 fu(61)=10^-(A_I*(I_d^(1/2)/(1+I_d^(1/2))-0.3*I_d))-gamma_d_H;%Davies;
192 fu(62)=10^-3.408*K_H/10^((((33.5-0.109*(T-273)+0.0014*(T-273)^2)*I_d-(1.5+0.015*(T-↙
273)+0.004*(T-273)^2)*I_d^2))...
193     /T)-C_d_co2a;
194
195
```

G.12 Lactic acid as feed solution and NaCl as draw solution

```
 1 clear
 2 clc
 3 [g]=[1.544e-03     2.1253e+01     2.904e-01      1.935e-05  4.537e-01      1.283e-04↙
4.828e-12     1.020e-02      2.731e-09...
 4 2.724e-09     2.6040e-06     7.898e-01      4.537e-01  9.993e-01      1.162e-03      9.604↙
e-01     1.290e-03     9.892e-01...
 5 9.545e-06     1.4718e-01     6.754e+02      5.550e+02  2.612e+04      1.503e+02      1.034↙
e+03     5.479e+00     1.581e+00...
 6 1.060e-05     1.5607e-01     1.000e-02      6.452e+02  2.545e+04      1.502e+02      1.579↙
e+00     5.129e+02     9.258e-06...
 7 1.426e-01     9.2126e-06     1.001e-02      4.065e-06  2.302e-09      9.989e-01      2.957↙
e+00     9.624e-01     1.148e-03...
 8 1.152e-03     5.4140e+00     6.748e+02      5.870e+02  3.354e-04      1.148e-03      1.401↙
e-08     6.181e-09     3.506e-09...
 9 2.942e-08     1.1770e-05     2.601e-06      9.989e-01  2.692e-09      2.724e-09      7.306↙
e-01     1.176e-05];
10 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
11 ts=0.5:0.0003:1800;
12 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
13 for ti=1:numel(ts);
14 t=ts(ti);
15 options = optimset('Algorithm','Levenberg-Marquardt','TolFun',1*10^-15,'TolX',1*10^-↙
15 ...
16      ,'Maxiter',25,'Maxfun',15000);
17
18 [r,fval,exitflag,output] = fsolve(@(z)Lactic_NaCl(z,t),[g],options)
19 [g]=[r];
20 j(ti)=r(1,1);
21 p(ti)=r(1,43);%pH_f
22 c(ti)=r(1,30);%C_f_a
23 d(ti)=r(1,50);%w
24 T(ti)=t;
25 d1(ti)=r(1,55);%J_co2
26 d2(ti)=r(1,52);%C_f_co2
27 d3(ti)=r(1,54);%C_f_HCO3C
28 d4(ti)=r(1,56);%C_d_co2
29 d5(ti)=r(1,57);%C_d_HCO3
30 d6(ti)=-log10(r(1,12)*r(1,11));%pH_d
31 d7(ti)=(1-r(1,43)/r(1,32))*100;% %Rejection
32 d8(ti)=r(1,32)/0.01;%Concentration Performance
33 %for Plotting
34 [A(ti)]=t;
35 [B(ti)]=d(ti);
36 [C(ti)]=p(ti);
37 [M(ti)]=d7(ti);
38 [N(ti)]=j(ti);
39 [K5(ti)]=d8(ti);
40 disp(ti)
41 fprintf('\n     Time(min)        J_w(L/dm2/min)          pH           Cf,a(mole/l)↙
W(kg)\n');
42 fprintf('\n%11.2f%20.4e%19.4f%19.4e%19.4e%\n' ,T(ti),j(ti),p(ti),c(ti),d(ti));
43 format short e
44 %Result of Best guess
45 fprintf('\n%8.3e%15.4e%13.3e%14.3e%11.3e%13.3e%13.3e%13.3e%13.3e\n'...
46     ,r(1,1),r(1,2),r(1,3),r(1,4),r(1,5),r(1,6),r(1,7),r(1,8),r(1,9));
47
```

```matlab
48  fprintf('\n%8.3e%15.4e%13.3e%14.3e%11.3e%13.3e%13.3e%13.3e%\n'...
49    ,r(1,10),r(1,11),r(1,12),r(1,13),r(1,14),r(1,15),r(1,16),r(1,17),r(1,18));
50
51  fprintf('\n%8.3e%15.4e%13.3e%14.3e%11.3e%13.3e%13.3e%13.3e%\n'...
52     ,r(1,19),r(1,20),r(1,21),r(1,22),r(1,23),r(1,24),r(1,25),r(1,26),r(1,27));
53
54  fprintf('\n%8.3e%15.4e%13.3e%14.3e%11.3e%13.3e%13.3e%13.3e%\n'...
55    ,r(1,28),r(1,29),r(1,30),r(1,31),r(1,32),r(1,33),r(1,34),r(1,35),r(1,36));
56
57  fprintf('\n%8.3e%15.4e%13.3e%14.3e%11.3e%13.3e%13.3e%13.3e%13.3e%13.3e%\n'...
58     ,r(1,37),r(1,38),r(1,39),r(1,40),r(1,41),r(1,42),r(1,43),r(1,44),r(1,45));
59
60  fprintf('\n%8.3e%15.4e%13.3e%14.3e%11.3e%13.3e%13.3e%13.3e%13.3e%\n'...
61    ,r(1,46),r(1,47),r(1,48),r(1,49),r(1,50),r(1,51),r(1,52),r(1,53),r(1,54));
62
63  fprintf('\n%8.3e%15.4e%13.3e%14.3e%11.3e%13.3e%13.3e%13.3e%\n'...
64     ,r(1,55),r(1,56),r(1,57),r(1,58),r(1,59),r(1,60),r(1,61),r(1,62));
65  end
66
67  %Time Interval%%%%%%%%%%%%%%%%%%%%%%%%%%%
68  Ai=0:30:1800;
69  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
70  format short
71  Ci=spline(A,C,Ai);
72  format short e
73  Bi=spline(A,B,Ai);
74
75
76  %Experiment Value W and pH%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
77  F=[0       0.0150    0.0320    0.0490    0.0640    0.0780    0.0910    0.1050    0.1180↙
0.1320    0.1450...
78  0.1620    0.1740    0.1850    0.1970    0.2100    0.2220    0.2350    0.2450↙
0.2570    0.2690    0.2810...
79  0.2930    0.3040    0.3160    0.3270    0.3380    0.3490    0.3600    0.3710↙
0.3810    0.3910    0.4010...
80  0.4120    0.4220    0.4320    0.4410    0.4510    0.4610    0.4710    0.4780↙
0.4890    0.4990    0.5080...
81  0.5170    0.5260    0.5350    0.5430    0.5520    0.5600    0.5670    0.5750↙
0.5820    0.5890    0.5970...
82  0.6040    0.6110    0.6180    0.6260    0.6320    0.6390];
83
84  G=[2.9260    2.9300    2.9260    2.9310    2.9220    2.9190    2.9130    2.9090↙
2.9010    2.8990    2.8990    2.8890    2.8880...
85  2.8800    2.8790    2.8770    2.8740    2.8710    2.8660    2.8630    2.8600↙
2.8550    2.8490    2.8460    2.8400    2.8350...
86  2.8300    2.8240    2.8200    2.8140    2.8080    2.8030    2.7970    2.7920↙
2.7860    2.7810    2.7750    2.7710    2.7650...
87  2.7600    2.7550    2.7510    2.7470    2.7420    2.7380    2.7330    2.7280↙
2.7250    2.7210    2.7160    2.7140    2.7090...
88  2.7050    2.7010    2.7000    2.6950    2.6930    2.6900    2.6850    2.6800↙
2.6760];
89  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
90
91  F_m=mean(F);
92  G_m=mean(G);
93
```

```matlab
 94 sum1=0;sum2=0;
 95 sum3=0;sum4=0;
 96 %%% Number of Data %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 97 for k=1:1:61
 98 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 99 sum1=(Bi(1,k)-F(1,k))^2+sum1;
100 sum2=(Bi(1,k)-F_m)^2+sum2;
101 sum3=(Ci(1,k)-G(1,k))^2+sum3;
102 sum4=(Ci(1,k)-G_m)^2+sum4;
103 end
104
105 format short
106 R2_W=1-sum1/sum2;
107 R2_pH=1-sum3/sum4;
108 MSE_W=sum1/k;
109 MSE_pH=sum3/k;
110 RMSE_W=sqrt(MSE_W)
111 RMSE_pH=sqrt(MSE_pH)
112 SEP_W=RMSE_W/F_m*100
113 SEP_pH=RMSE_pH/G_m*100
114
115 % Plotting
116 subplot(2,3,1)
117 plot([A]/60,[B],'-g',[Ai]/60,[F],'xr')
118 xlabel('Time (hr)')
119 ylabel('Weight (Kg)')
120 title('Weight Change as a Function of Time')
121 legend('Model','Experiment',2)
122
123 subplot(2,3,2)
124 plot([A]/60,[C],'-g',[Ai]/60,[G],'xr')
125 xlabel('Time (hr)')
126 ylabel('pH')
127 title('pH as a Function of Time')
128 legend('Model','Experiment',1)
129
130 subplot(2,3,3)
131 plot([A]/60,[M],'-g')
132 xlabel('Time (hr)')
133 ylabel('%Rejection')
134 title('%Rejection as a Function of Time')
135 legend('Model',1)
136
137 subplot(2,3,4)
138 plot([N],[M],'-g')
139 xlabel('Jw (L/dm2/min)')
140 ylabel('%Rejection')
141 title('%Rejection as a Function of Jw')
142 legend('Model',1)
143
144 subplot(2,3,5)
145 plot([A]/60,[K5],'-g')
146 xlabel('Time (hr)')
147 ylabel('Cfa/Cf0')
148 title('Cfa/Cf0 as a Function of Time')
149 legend('Model',1)
```

```
150
151 subplot(2,3,6)
152 plot([N],[K5],'-g')
153 xlabel('Jw (L/dm2/min)')
154 ylabel('Cfa/Cf0')
155 title('Cfa/Cf0 as a Function of Jw')
156 legend('Model',1)
157
158 fprintf('\n      \n')
159 fprintf('\n      \n')
160 fprintf('\n CONCLUSION    \n')
161 fprintf('\n       R2_W            MSE_W                   R2_pH            MSE_pH\n');
162 fprintf('\n%11.3f%20.4e%19.4f%19.4e%\n',R2_W,MSE_W,R2_pH,MSE_pH)
163 fprintf('\n      \n')
164 fprintf('\n     Time(min)        J_w(L/dm2/min)              pH              Cf,a(mole/l)↙
W(kg)\n');
165
166 for x=5:5:2000000000;
167 z=x*1000;
168 fprintf('\n%11.4f %20.4e%19.4f%19.4e%19.4e\n',T(z),j(z),p(z),c(z),d(z));
169
170 end
171
172
```

```matlab
1
2  function fu=Lactic_NaCl(z,t)
3  %Independent Variable
4  V_f_0=1;C_d_s_0=1;C_f_a_0=10*10^(-3);V_d_0=0.5;
5  %Fixed Variable
6  T=301;f=1.58;
7  % T=300..normal f=1.58...normal
8  L=0.9207;D=0.023;W=0.4572;A_m=42*10^(-2);d_h=0.0438;
9  %Constant Variable
10
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
11 B_a=0.153/60/100;  %B of Lactic Acid
12 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
13 B_s=0.256/60/100;
14 A=0.442/60/100;
15 S=500*10^-5;
16 K_a_T=10^-(1286.49/T-4.8607+0.014776*T);%Lactic Acid
17 D_A_298=6.198*10^(-6);D_HA_298=4.5834*10^(-6);%Lactic Physicocmical Property
18 D_H_298=5.587*10^(-5);
19
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%5
20 o_s=0.936;
21 den_f=999.65+2.0438/10*(T-273)-6.174*10^-2*(T-273)^1.5;
22 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
23 n_s=2;o_a=1;R=0.08314;
24 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
25 K_CO2_T=10^-(3404.71/T-14.8435+0.032786*T);
26 K_w_T=10^(-4470.99/T+6.0875-0.017060*T);
27 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
28 E=2727.586+0.6224107*T-466.9151*log(T)-52000.87/T;
29 d=1-(((T-273)-3.9863)^2*((T-273)+288.9414))/(508929.2*((T-273)+68.12963))...
30 +0.011445*exp((-374.3)/(T-273));
31 A_I=(1.82483*10^6*d^0.5)/(E*T)^1.5;
32 K=1/K_a_T;
33 visco_298=((298-273)+246)/((0.05594*(298-273)+5.2842)*(298-273)+137.37);
34 visco_T=((T-273)+246)/((0.05594*(T-273)+5.2842)*(T-273)+137.37);
35 D_H_T=D_H_298*T/298*visco_298/visco_T ;
36 D_A_T=D_A_298*T/298*visco_298/visco_T;
37 D_HA_T=D_HA_298*T/298*visco_298/visco_T;
38 D_1_T=2/(1/D_H_T +1/D_A_T);
39 K_H=0.034*exp(-2400*(1/T-1/298));%************************
40 %For carbonic acid
41 P_M =0.0025;
42 %Unknown Variable
43 J_w=z(1);P_i=z(2);P_m=z(3);J_s=z(4);C_i_s=z(5);
44 C_mf_s=z(6);J_a=z(7);C_mf_a=z(8);C_i_a=z(9);C_d_a=z(10);
45 C_d_H=z(11);gamma_d_H=z(12);I_i=z(13);C_d_s=z(14);
46 C_mf_H=z(15);gamma_m_H=z(16);I_m=z(17);C_md_s=z(18);
47 D_d_s=z(19);k_d=z(20);Sh_d=z(21);Sc_d=z(22);Re_d=z(23);
48 v_d=z(24);den_d=z(25);mul_d=z(26);f_2=z(27);D_f_a=z(28);
49 k_f_a=z(29);C_f_a=z(30);Sh_f_a=z(31);Re_f=z(32);v_f=z(33);
50 f_1=z(34);Sc_f_a=z(35);D_f_s=z(36);k_f_s=z(37);C_f_s=z(38);
51 C_f_ai=z(39);C_f_si=z(40);C_d_ai=z(41);C_d_si=z(42);pH_f=z(43);
52 gamma_f_H=z(44);C_f_A=z(45);I_f=z(46);mul_f=z(47);
```

```
53 Sh_f_s=z(48);Sc_f_s=z(49);w=z(50);C_f_H=z(51);
54 C_f_co2=z(52);C_f_co2i=z(53);C_f_HCO3c=z(54);
55 J_co2=z(55);C_d_co2=z(56);C_d_HCO3=z(57);I_d=z(58);C_d_A=z(59);
56 C_i_H=z(60);gamma_i_H=z(61);C_d_co2a=z(62);
57 %Math Model
58 fu(1)=A*(P_i-P_m )-J_w;
59 fu(2)= B_s *(C_i_s-C_mf_s)-J_s;
60 fu(3)= B_a*(C_mf_a-C_i_a)-J_a;
61 fu(4)=C_d_a-C_i_a;
62 fu(5)=J_a/J_w-C_d_a;
63 fu(6)= o_s*n_s*C_i_s*R*T+o_a*(C_i_a+C_i_H)*R*T-P_i;
64 fu(7)=(-(K_a_T)+((K_a_T)^2+4*K_a_T*gamma_i_H*gamma_i_H*C_d_a)^(1/2))...
65     /(2*gamma_i_H*gamma_i_H)-C_i_H;
66 fu(8)=A_I*(I_i^(1/2)/(1+I_i^(1/2))- 0.3*I_i)+log10(gamma_i_H);%"Davies"
67 fu(9)=0.5*(2*C_i_s+2*C_i_H)- I_i;
68 fu(10)=o_s*n_s*C_mf_s*R*T+o_a*(C_mf_a+C_mf_H)*R*T-P_m;
69 fu(11)=(-(K_a_T)+((K_a_T)^2+4*K_a_T*gamma_m_H*gamma_m_H*C_mf_a)^(1/2))...
70     /(2*gamma_m_H*gamma_m_H)-C_mf_H;
71 fu(12)=A_I*(I_m^(1/2)/(1+I_m^(1/2))- 0.3*I_m)+log10(gamma_m_H);%"Davies"
72 fu(13)=0.5*(2*C_mf_s+2*C_mf_H)-I_m;
73
74 fu(14)=((C_md_s-C_i_s)*(12.625*10^(-11)*((0.789*C_i_s+0.211*C_md_s)^4 ...
75 /(J_w*(0.789*C_i_s+0.211*C_md_s)+ J_s)+(0.211*C_i_s+0.789*C_md_s)^4 ...
76 /(J_w*(0.211*C_i_s+0.789*C_md_s)+ J_s ))...
77 -3.885*10^(-9)*((0.789*C_i_s+0.211*C_md_s)^3 ...
78 /(J_w*(0.789*C_i_s+0.211*C_md_s)+ J_s)+(0.211*C_i_s+0.789*C_md_s)^3 ...
79 /(J_w*(0.211*C_i_s+0.789*C_md_s)+ J_s ))...
80 +6.975*10^(-8)*((0.789*C_i_s+0.211*C_md_s)^2 ...
81 /(J_w*(0.789*C_i_s+0.211*C_md_s)+ J_s)+(0.211*C_i_s+0.789*C_md_s)^2 ...
82 /(J_w*(0.211*C_i_s+0.789*C_md_s)+ J_s ))...
83 -4.03*10^(-7)*((0.789*C_i_s+0.211*C_md_s)...
84 /(J_w*(0.789*C_i_s+0.211*C_md_s)+ J_s)+(0.211*C_i_s+0.789*C_md_s)...
85 /(J_w*(0.211*C_i_s+0.789*C_md_s)+ J_s))...
86 +0.53*10^(-5)*(1/(J_w*(0.789*C_i_s+0.211*C_md_s)+J_s)...
87 +1/(J_w*(0.211*C_i_s+0.789*C_md_s)+ J_s))))-S;
88
89 fu(15)=((C_d_s-C_md_s)*(12.625*10^(-11)*((0.789*C_md_s+0.211*C_d_s)^4 ...
90 /(J_w*(0.789*C_md_s+0.211*C_d_s)+ J_s)+(0.211*C_md_s+0.789*C_d_s)^4 ...
91 /(J_w*(0.211*C_md_s+0.789*C_d_s)+ J_s ))...
92 -3.885*10^(-9)*((0.789*C_md_s+0.211*C_d_s)^3 ...
93 /(J_w*(0.789*C_md_s+0.211*C_d_s)+ J_s)+(0.211*C_md_s+0.789*C_d_s)^3 ...
94 /(J_w*(0.211*C_md_s+0.789*C_d_s)+ J_s ))...
95 +6.975*10^(-8)*((0.789*C_md_s+0.211*C_d_s)^2 ...
96 /(J_w*(0.789*C_md_s+0.211*C_d_s)+ J_s)+(0.211*C_md_s+0.789*C_d_s)^2 ...
97 /(J_w*(0.211*C_md_s+0.789*C_d_s)+ J_s )...
98 -4.03*10^(-7)*((0.789*C_md_s+0.211*C_d_s)...
99 /(J_w*(0.789*C_md_s+0.211*C_d_s)+ J_s)+(0.211*C_md_s+0.789*C_d_s)...
100 /(J_w*(0.211*C_md_s+0.789*C_d_s)+ J_s))...
101 +0.53*10^(-5)*(1/(J_w*(0.789*C_md_s+0.211*C_d_s)+J_s)...
102 +1/(J_w*(0.211*C_md_s+0.789*C_d_s)+ J_s))))-D_d_s/k_d;
103
104 fu(16)=Sh_d*D_d_s/d_h-k_d;
105 fu(17)=5.05*10^(-11)*C_d_s^4-2.59*10^(-9)*C_d_s^3+4.65*10^(-8)*C_d_s^2-4.03*10^(-7)
*C_d_s+1.06*10^(-5)-D_d_s;
106 fu(18)=L*v_d*den_d/mul_d-Re_d;
107 fu(19)=(f_2+f)/2/W/D-v_d;
```

```matlab
108 fu(20)=f+J_w*A_m-f_2;
109
110 %
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
^^^^^^^^^^^^^^^^^
111 fu(21)=0.0538*C_d_s+0.4159+5.0095/C_d_s-mul_d/C_d_s;%Math Lab
112 fu(22)=37.0166*C_d_s+997.2911-den_d;%Math lab
113 %
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
^^^^^^^^^^^^^^^^^^^^^^
114 fu(23)= 0.04*Re_d^(3/4)*Sc_d^(1/3)-Sh_d;
115 fu(24)= mul_d/den_d/D_d_s-Sc_d;
116 fu(25)=(0.5*(C_f_a-C_mf_a)*((D_1_T-D_HA_T)/...
117     (sqrt(1+4*K*(0.789*C_mf_a+0.211*C_f_a))*(J_w*(0.789*C_mf_a+0.211*C_f_a)-J_a))...
118     +(D_1_T-D_HA_T)/(sqrt(1+4*K*(0.211*C_mf_a+0.789*C_f_a))*(J_w*(0.211*C_mf_a...
119     +0.789*C_f_a)-J_a))+ D_HA_T/(J_w*(0.789*C_mf_a+0.211*C_f_a)-J_a)...
120     +D_HA_T/((J_w*(0.211*C_mf_a+0.789*C_f_a)-J_a))))+D_f_a/k_f_a;
121 fu(26)=Sh_f_a*D_f_a/d_h-k_f_a;
122 fu(27)=(D_1_T/(2*K*C_f_a)*(-1+sqrt(1+4*K*C_f_a))+D_HA_T/...
123     (4*K*C_f_a)*(-1+sqrt(1+4*K*C_f_a))^2)-D_f_a;
124 fu(28)=L*v_f*den_f/mul_f -Re_f;
125 fu(29)=(f+f_1)/W/D/2-v_f;
126 fu(30)=f-J_w*A_m-f_1;
127 fu(31)= 0.04*Re_f^(3/4)*Sc_f_a^(1/3)-Sh_f_a;
128 fu(32)= mul_f/den_f/D_f_a-Sc_f_a;
129
130 fu(33)=((C_f_s-C_mf_s)*(12.625*10^(-11)*((0.789*C_mf_s+0.211*C_f_s)^4 ...
131 /(J_w*(0.789*C_mf_s+0.211*C_f_s)+ J_s)+(0.211*C_mf_s+0.789*C_f_s)^4 ...
132 /(J_w*(0.211*C_mf_s+0.789*C_f_s)+ J_s ))...
133 -3.885*10^(-9)*((0.789*C_mf_s+0.211*C_f_s)^3 ...
134 /(J_w*(0.789*C_mf_s+0.211*C_f_s)+ J_s)+(0.211*C_mf_s+0.789*C_f_s)^3 ...
135 /(J_w*(0.211*C_mf_s+0.789*C_f_s)+ J_s ))...
136 +6.975*10^(-8)*((0.789*C_mf_s+0.211*C_f_s)^2 ...
137 /(J_w*(0.789*C_mf_s+0.211*C_f_s)+ J_s)+(0.211*C_mf_s+0.789*C_f_s)^2 ...
138 /(J_w*(0.211*C_mf_s+0.789*C_f_s)+ J_s ))...
139 -4.03*10^(-7)*((0.789*C_mf_s+0.211*C_f_s)...
140 /(J_w*(0.789*C_mf_s+0.211*C_f_s)+ J_s)+(0.211*C_mf_s+0.789*C_f_s)...
141 /(J_w*(0.211*C_mf_s+0.789*C_f_s)+ J_s))...
142 +0.53*10^(-5)*(1/(J_w*(0.789*C_mf_s+0.211*C_f_s)+J_s)...
143 +1/(J_w*(0.211*C_mf_s+0.789*C_f_s)+ J_s))))+D_f_s/k_f_s;
144
145 fu(34)=Sh_f_s*D_f_s/d_h-k_f_s;
146 fu(35)=5.05*10^(-11)*C_f_s^4-2.59*10^(-9)*C_f_s^3+4.65*10^(-8)*C_f_s^2-4.03*10^(-7)
*C_f_s+1.06*10^(-5)-D_f_s;
147
148 %Mole balance Equation
149 fu(36)=C_f_a*(V_f_0+f_1*t-f*t)-(V_f_0*C_f_a_0+C_f_ai*f_1*t-C_f_a*f*t);%new
150 fu(37)=C_f_si*(V_f_0+f_1*t-f*t)-C_f_s*f_1*t+C_f_si*f*t;%old
151 fu(38)=C_f_a*f-J_a*A_m-C_f_ai*f_1;%new
152 fu(39)=C_f_si*f+J_s*A_m-C_f_s*f_1;%old
153 fu(40)=C_d_s_0*V_d_0+C_d_si*f_2*t-C_d_s*f*t-C_d_s*(V_d_0+f_2*t-f*t);%new
154 fu(41)=C_d_a*f_2*t-C_d_ai*f*t-C_d_ai*(V_d_0+f_2*t-f*t);%old
155 fu(42)=C_d_s*f-J_s*A_m-C_d_si*f_2;%new
156 fu(43)=C_d_ai*f+J_a*A_m-C_d_a*f_2;%old
157
158 %additional Equation
```

```
159 fu(44)= 0.04*Re_f^(3/4)*Sc_f_s^(1/3)-Sh_f_s;
160 fu(45)= mul_f/den_f/D_f_s-Sc_f_s;
161 fu(46)=J_w*A_m*t*den_d/1000-w;
162
163 %pH Equation..
164 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
165 fu(47)= pH_f+log10(gamma_f_H*C_f_H);
166 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
167 fu(48)=K_a_T*(C_f_a-C_f_A)/gamma_f_H/gamma_f_H/C_f_H-C_f_A;
168 fu(49)=A_I*(I_f^(1/2)/(1+I_f^(1/2))-0.3*I_f)+log10(gamma_f_H);%Davies;
169 %ionic streanght in feed solution%%%%%%%%%%%%%%%%%%%%%
170 fu(50)=0.5*(2*C_f_si+C_f_H+K_w_T/C_f_H/gamma_f_H/gamma_f_H+C_f_A+C_f_HCO3c)- I_f;
171 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
172 %Charge Balance Equation %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
173 fu(51)=(C_f_HCO3c+C_f_A+K_w_T/C_f_H/gamma_f_H/gamma_f_H)-C_f_H;
174 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
175 fu(52)=P_M*C_d_co2-J_co2;
176 fu(53)=C_f_co2i*(V_f_0+f_1*t-f*t)-(C_f_co2*f_1*t-C_f_co2i*f*t);
177 fu(54)=C_f_co2i*f*t+J_co2*A_m*t-C_f_co2*f_1*t;
178 fu(55)=10^-3.45/gamma_f_H/gamma_f_H/C_f_H*(C_f_co2-C_f_HCO3c)-C_f_HCO3c;
179 % Draw solution
180 fu(56)=gamma_d_H*gamma_d_H*C_d_H*C_d_HCO3/K_CO2_T+C_d_HCO3-C_d_co2;%******
181 fu(57)=(-K_CO2_T+sqrt(K_CO2_T^2+4*K_CO2_T*gamma_d_H*gamma_d_H*C_d_co2a))/↙
(2*gamma_d_H*gamma_d_H)-C_d_HCO3;
182 %ionic streanght in draw solution%%%%%%%%%%%%%%%%%%%%
183 fu(58)=0.5*(2*C_d_si+C_d_H+K_w_T/C_d_H/gamma_d_H/gamma_d_H+C_d_A+C_d_HCO3)- I_d;
184 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
185 %Charge balance at draw solution%%%%%%%%%%%%%%%%%%%%
186 fu(59)=(C_d_HCO3+C_d_A+K_w_T/C_f_H/gamma_d_H/gamma_d_H)-C_d_H;
187 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
188 fu(60)=K_a_T*(C_d_a-C_d_A)/gamma_d_H/gamma_d_H/C_d_H-C_d_A;
189 fu(61)=10^-(A_I*(I_d^(1/2)/(1+I_d^(1/2))-0.3*I_d))-gamma_d_H;%Davies;
190 fu(62)=10^-3.408*K_H/10^((((33.5-0.109*(T-273)+0.0014*(T-273)^2)*I_d-(1.5+0.015*(T-↙
273)+0.004*(T-273)^2)*I_d^2))...
191      /T)-C_d_co2a;
192
193
```

G.13 A mixture of 10 mM acetic acid and 10 mM valeric acid as feed solution and NaCl as draw solution

```
 1 clear
 2 clc
 3 [g]=[1.534e-03    2.1368e+01    5.433e-01    1.946e-05 4.561e-01    1.266e-04↙
3.988e-11   1.013e-02    2.178e-08...
 4 2.053e-08    2.6311e-06    7.306e-01    4.561e-01 9.997e-01    5.766e-04    9.703↙
e-01   7.031e-04    9.896e-01...
 5 9.545e-06    1.4718e-01    6.754e+02    5.550e+02 2.612e+04    1.503e+02    1.034↙
e+03   5.479e+00    1.581e+00...
 6 1.060e-05    1.5607e-01    1.000e-02    6.452e+02 2.545e+04    1.502e+02    1.579↙
e+00   5.129e+02    9.258e-06...
 7 1.426e-01    6.8090e-06    1.001e-02    1.634e-06 9.595e-09    9.993e-01    3.256↙
e+00   9.730e-01    3.140e-04...
 8 5.721e-04    5.4291e+00    6.748e+02    5.870e+02 1.333e-04    5.704e-04    1.038↙
e-08   2.492e-09    4.116e-09...
 9 2.966e-08    1.1865e-05    2.601e-06    9.993e-01 9.990e+02    1.877e-08    1.020↙
e-02   4.215e-09    8.332e-12...
 10 4.146e-09    3.7171e-09    2.602e-04    1.000e-02 1.060e-05    1.561e-01    6.452↙
e+02   5.129e+02    1.001e-02...
 11 1.937e-09    2.5647e-04    3.163e-04    7.899e-01 1.668e-07    2.167e-08    4.189↙
e-09   1.176e-05];
 12 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 13 ts=0.2:0.006:1800;
 14 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 15 for ti=1:numel(ts);
 16 t=ts(ti);
 17 options = optimset('Algorithm','Levenberg-Marquardt','TolFun',1*10^-17,'TolX',1*10^-↙
17 ...
 18     ,'Maxiter',25,'Maxfun',15000);
 19
 20 [r,fval,exitflag,output] = fsolve(@(z)Acetic10_Valeric10(z,t),[g],options)
 21 [g]=[r];
 22 j(ti)=r(1,1);%Jw
 23 c(ti)=r(1,30);%C_f_a
 24 d(ti)=r(1,50);%W
 25 p(ti)=r(1,43);%pH
 26 T(ti)=t;
 27 e(ti)=r(1,67);%C_f_a1
 28 % Rejection
 29 O(ti)=(1-r(1,43)/r(1,32))*100;
 30 K4(ti)=r(1,32)/0.01;
 31 %for Plotting
 32 K1(ti)=r(1,1);
 33 [A(ti)]=t;
 34 [B(ti)]=d(ti);%w
 35 [C(ti)]=p(ti);%pH
 36 % For Rejection
 37 [M(ti)]=O(ti);
 38 [N(ti)]=K1(ti);
 39 [K5(ti)]=K4(ti);%Cfa/Cf0
 40 disp(ti)
 41 fprintf('\n    Time(min)        J_w(L/dm2/min)            pH           Cf,a(mole/l)↙
W(kg)          Cf,a1(mole/l)\n');
 42 fprintf('\n%11.4f%20.4e%19.4f%19.4e%19.4e%19.4e\n' ,T(ti),j(ti),p(ti),c(ti),d(ti),e↙
(ti));
 43 format short e
 44 %Result of Best guess
```

```
45 fprintf('\n%8.3e%15.4e%13.3e%14.3e%11.3e%13.3e%13.3e%13.3e%\n'...
46     ,r(1,1),r(1,2),r(1,3),r(1,4),r(1,5),r(1,6),r(1,7),r(1,8),r(1,9));
47
48  fprintf('\n%8.3e%15.4e%13.3e%14.3e%11.3e%13.3e%13.3e%13.3e%\n'...
49     ,r(1,10),r(1,11),r(1,12),r(1,13),r(1,14),r(1,15),r(1,16),r(1,17),r(1,18));
50
51 fprintf('\n%8.3e%15.4e%13.3e%14.3e%11.3e%13.3e%13.3e%13.3e%\n'...
52     ,r(1,19),r(1,20),r(1,21),r(1,22),r(1,23),r(1,24),r(1,25),r(1,26),r(1,27));
53
54 fprintf('\n%8.3e%15.4e%13.3e%14.3e%11.3e%13.3e%13.3e%13.3e%\n'...
55     ,r(1,28),r(1,29),r(1,30),r(1,31),r(1,32),r(1,33),r(1,34),r(1,35),r(1,36));
56
57 fprintf('\n%8.3e%15.4e%13.3e%14.3e%11.3e%13.3e%13.3e%13.3e%13.3e%13.3e%\n'...
58     ,r(1,37),r(1,38),r(1,39),r(1,40),r(1,41),r(1,42),r(1,43),r(1,44),r(1,45));
59
60 fprintf('\n%8.3e%15.4e%13.3e%14.3e%11.3e%13.3e%13.3e%13.3e%\n'...
61     ,r(1,46),r(1,47),r(1,48),r(1,49),r(1,50),r(1,51),r(1,52),r(1,53),r(1,54));
62
63 fprintf('\n%8.3e%15.4e%13.3e%14.3e%11.3e%13.3e%13.3e%13.3e%\n'...
64     ,r(1,55),r(1,56),r(1,57),r(1,58),r(1,59),r(1,60),r(1,61),r(1,62),r(1,63));
65
66 fprintf('\n%8.3e%15.4e%13.3e%14.3e%11.3e%13.3e%13.3e%13.3e%\n'...
67     ,r(1,64),r(1,65),r(1,66),r(1,67),r(1,68),r(1,69),r(1,70),r(1,71),r(1,72));
68
69 fprintf('\n%8.3e%15.4e%13.3e%14.3e%11.3e%13.3e%13.3e%13.3e%\n'...
70     ,r(1,73),r(1,74),r(1,75),r(1,76),r(1,77),r(1,78),r(1,79),r(1,80));
71
72 end
73
74 %Time Interval%%%%%%%%%%%%%%%%%%%%%%%%%%%
75 Ai=0:30:1800;
76 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
77 format short
78 Ci=spline(A,C,Ai);
79 format short e
80 Bi=spline(A,B,Ai);
81
82 %Experiment Value W and pH%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
83 F= [0    0.0170    0.0330    0.0490    0.0640    0.0780    0.0930    0.1070    0.1200↙
0.1340    0.1470    0.1600    0.1730...
84 0.1850    0.1970    0.2090    0.2210    0.2320    0.2440    0.2600    0.2710↙
0.2820    0.2920    0.3030    0.3140    0.3230...
85 0.3340    0.3440    0.3540    0.3640    0.3730    0.3820    0.3910    0.4000↙
0.4090    0.4170    0.4260    0.4340    0.4430...
86 0.4500    0.4580    0.4660    0.4790    0.4860    0.4930    0.5000    0.5060↙
0.5130    0.5190    0.5260    0.5320    0.5380...
87 0.5440    0.5490    0.5550    0.5600    0.5650    0.5700    0.5750    0.5800↙
0.5850];
88
89 G= [3.2770    3.2680    3.2640    3.2590    3.2540    3.2460    3.2390    3.2310↙
3.2220    3.2150    3.2040    3.1970    3.1890...
90 3.1810    3.1730    3.1640    3.1550    3.1470    3.1370    3.1290    3.1190↙
3.1100    3.1010    3.0910    3.0820    3.0720...
91 3.0630    3.0540    3.0450    3.0360    3.0270    3.0170    3.0070    2.9980↙
2.9900    2.9790    2.9700    2.9600    2.9520...
92 2.9440    2.9360    2.9260    2.9180    2.9070    2.8950    2.8900    2.8840↙
```

```
   2.8750    2.8660    2.8580    2.8510    2.8400...
93 2.8340    2.8250    2.8160    2.8090    2.8000    2.7930    2.7840    2.7770↵
   2.7710];
94 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
95
96 F_m=mean(F);
97 G_m=mean(G);
98
99 sum1=0;sum2=0;
100 sum3=0;sum4=0;
101 %%% Number of Data %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
102 for k=1:1:61
103 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
104 sum1=(Bi(1,k)-F(1,k))^2+sum1;
105 sum2=(Bi(1,k)-F_m)^2+sum2;
106 sum3=(Ci(1,k)-G(1,k))^2+sum3;
107 sum4=(Ci(1,k)-G_m)^2+sum4;
108 end
109
110 format short
111 R2_W=1-sum1/sum2;
112 R2_pH=1-sum3/sum4;
113 MSE_W=sum1/k;
114 MSE_pH=sum3/k;
115 RMSE_W=sqrt(MSE_W)
116 RMSE_pH=sqrt(MSE_pH)
117 SEP_W=RMSE_W/F_m*100
118 SEP_pH=RMSE_pH/G_m*100
119
120 % Plotting
121 subplot(2,3,1)
122 plot([A]/60,[B],'-g',[Ai]/60,[F],'xr')
123 xlabel('Time (hr)')
124 ylabel('Weight (Kg)')
125 title('Weight Change as a Function of Time')
126 legend('Model','Experiment',2)
127
128 subplot(2,3,2)
129 plot([A]/60,[C],'-g',[Ai]/60,[G],'xr')
130 xlabel('Time (hr)')
131 ylabel('pH')
132 title('pH as a Function of Time')
133 legend('Model','Experiment',1)
134
135 subplot(2,3,3)
136 plot([A]/60,[M],'-g')
137 xlabel('Time (hr)')
138 ylabel('%Rejection')
139 title('%Rejection as a Function of Time')
140 legend('Model',1)
141 %text(2,3.3638,'R2=    MSE=')
142
143 subplot(2,3,4)
144 plot([N],[M],'-g')
145 xlabel('Jw (L/dm2/min)')
146 ylabel('%Rejection')
```

```
147 title('%Rejection as a Function of Jw')
148 legend('Model',1)
149 %text(2,3.3638,'R2=    MSE=')
150
151 subplot(2,3,5)
152 plot([A]/60,[K5],'-g')
153 xlabel('Time (hr)')
154 ylabel('Cfa/Cf0')
155 title('Cfa/Cf0 as a Function of Time')
156 legend('Model',1)
157 %text(2,3.3638,'R2=    MSE=')
158
159 subplot(2,3,6)
160 plot([N],[K5],'-g')
161 xlabel('Jw (L/dm2/min)')
162 ylabel('Cfa/Cf0')
163 title('Cfa/Cf0 as a Function of Jw')
164 legend('Model',1)
165 %text(2,3.3638,'R2=    MSE=')
166
167 fprintf('\n     \n')
168 fprintf('\n     \n')
169 fprintf('\n CONCLUSION    \n')
170 fprintf('\n       R2_W            MSE_W                    R2_pH           MSE_pH\n');
171 fprintf('\n%11.3f%20.4e%19.4f%19.4e%\n',R2_W,MSE_W,R2_pH,MSE_pH)
172 fprintf('\n     \n')
173 fprintf('\n     \n')
174 fprintf('\n    Time(min)        J_w(L/dm2/min)            pH              Cf,a(mole/l)↙
W(kg)        C_f_a1\n');
175 for x=5:5:200000000;
176 z=x*100;
177 fprintf('\n%11.3f%20.4e%19.4f%19.4e%19.4e%19.4e%\n',T(z),j(z),p(z),c(z),d(z),e(z));
178
179 end
180
181
```

```
 1
 2 function fu=Acetic10_Valeric10(z,t)
 3 %Independent Variables
 4 V_f_0=1;C_d_s_0=1;V_d_0=0.5;
 5 %Fixed Variable
 6 T=301;f=1.58;
 7 L=0.9207;D=0.023;W=0.4572;A_m=42*10^(-2);d_h=0.0438;
 8 % Acid Variables
 9 %
```
^^^
```
10 C_f_a_0=10*10^(-3);
11 C_f_a1_0=10*10^(-3);
12 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
13 B_a=2.093/60/100;% B of Acetic Acid
14 B_a1=0.491/60/100;% B of Valeric Acid
15 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
16 D_A_298=6.534*10^(-6);D_HA_298=7.74*10^(-6);%Acetic Physicocmical Property
17 D_A1_298=5.226*10^(-6);D_HA1_298=4.902*10^(-6);%Valeric Physicocmical Property
18 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
19 K_a_T=10^-(1170.48/T-3.1649+0.013399*T);%Acetic acid
20 K_a1_T=10^-(921.38/T-1.8574+0.012105*T);%Valeric Acid
21 %
```
^^
^^^^
```
22 %Constant Variable
23 B_s=0.256/60/100;% normal..B_s(0.256) =4.27*10^(-5)
24 A=0.442/60/100;%..(0.413) New Value
25 o_s=0.936;
26 S=500*10^-5;
27 D_H_298=5.587*10^(-5);
28 visco_298=((298-273)+246)/((0.05594*(298-273)+5.2842)*(298-273)+137.37);
29 visco_T=((T-273)+246)/((0.05594*(T-273)+5.2842)*(T-273)+137.37);
30 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
31 o_a=1;n_s=2;R=0.08314;
32 o_a1=1;
33 %Temperature Correction
34 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
35 K_CO2_T=10^-(3404.71/T-14.8435+0.032786*T);
36 K_w_T=10^(-4470.99/T+6.0875-0.017060*T);
37 E=2727.586+0.6224107*T-466.9151*log(T)-52000.87/T;
38 d=1-(((T-273)-3.9863)^2*((T-273)+288.9414))/(508929.2*((T-273)+68.12963))...
39 +0.011445*exp((-374.3)/(T-273));
40 A_I=(1.82483*10^6*d^0.5)/(E*T)^1.5;
41 K=1/K_a_T;
42 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
43 K1=1/K_a1_T;
44 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
45 D_H_T=D_H_298*T/298*visco_298/visco_T ;
46 D_A_T=D_A_298*T/298*visco_298/visco_T;
47 D_HA_T=D_HA_298*T/298*visco_298/visco_T;
48 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
49 D_A1_T=D_A1_298*T/298*visco_298/visco_T;
50 D_HA1_T=D_HA1_298*T/298*visco_298/visco_T;
51 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
52 D_1_T=2/(1/D_H_T +1/D_A_T);
53 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
```

```
54 D_11_T=2/(1/D_H_T +1/D_A1_T);
55 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
56 K_H=0.034*exp(-2400*(1/T-1/298));%*************************
57 %For carbonic acid
58 P_M =0.0025;
59 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
60 %Unknown Variable
61 J_w=z(1);P_i=z(2);P_m=z(3);J_s=z(4);C_i_s=z(5);
62 C_mf_s=z(6);J_a=z(7);C_mf_a=z(8);C_i_a=z(9);C_d_a=z(10);
63 C_d_H=z(11);gamma_i_H=z(12);I_i=z(13);C_d_s=z(14);
64 C_mf_H=z(15);gamma_mf_H=z(16);I_mf=z(17);C_md_s=z(18);
65 D_d_s=z(19);k_d=z(20);Sh_d=z(21);Sc_d=z(22);Re_d=z(23);
66 v_d=z(24);den_d=z(25);mul_d=z(26);f_2=z(27);D_f_a=z(28);
67 k_f_a=z(29);C_f_a=z(30);Sh_f_a=z(31);Re_f=z(32);v_f=z(33);
68 f_1=z(34);Sc_f_a=z(35);D_f_s=z(36);k_f_s=z(37);C_f_s=z(38);
69 C_f_ai=z(39);C_f_si=z(40);C_d_ai=z(41);C_d_si=z(42);pH_f=z(43);
70 gamma_f_H=z(44);C_f_A=z(45);I_f=z(46);mul_f=z(47);Sh_f_s=z(48);
71 Sc_f_s=z(49);w=z(50);C_f_H=z(51);C_f_co2=z(52);C_f_co2i=z(53);
72 C_f_HCO3c=z(54);J_co2=z(55);C_d_co2=z(56);C_d_HCO3=z(57);
73 I_d=z(58);den_f=z(59);C_d_A=z(60);
74 %For 2 acid
75 C_mf_a1=z(61);C_i_a1=z(62);J_a1=z(63);C_d_a1=z(64);C_d_A1=z(65);C_mf_A1=z(66);
76 C_f_a1=z(67);D_f_a1=z(68);k_f_a1=z(69);Sh_f_a1=z(70);Sc_f_a1=z(71);
77 C_f_ai1=z(72);C_d_ai1=z(73);C_f_A1=z(74);C_mf_A=z(75);gamma_d_H=z(76);C_i_H=z(77);
78 C_i_A=z(78);C_i_A1=z(79);C_d_co2a=z(80);
79 %Math Model
80 fu(1)=A*(P_i-P_m )-J_w;
81 fu(2)= B_s *(C_i_s-C_mf_s)-J_s;
82 fu(3)= B_a*(C_mf_a-C_i_a)-J_a;
83 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
84 fu(4)= B_a1*(C_mf_a1-C_i_a1)-J_a1;
85 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
86 fu(5)=C_d_a-C_i_a;
87 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
88 fu(6)=C_d_a1-C_i_a1;
89 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
90 fu(7)=J_a/J_w-C_d_a;
91 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
92 fu(8)=J_a1/J_w-C_d_a1;
93 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
94 fu(9)= o_s*n_s*C_i_s*R*T+o_a*(C_i_a+C_i_H)*R*T+o_a1*(C_i_a1+C_i_H)*R*T-P_i;
95 %Charge Balance Equation at interface %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
96 fu(10)=C_i_A+C_i_A1+K_w_T/C_i_H/gamma_i_H/gamma_i_H-C_i_H;
97 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
98 fu(11)=A_I*(I_i^(1/2)/(1+I_i^(1/2))- 0.3*I_i)+log10(gamma_i_H);%"Davies"
99 fu(12)=0.5*(2*C_i_s+C_i_H+C_i_A+C_i_A1+K_w_T/C_i_H/gamma_i_H/gamma_i_H)- I_i;
100 fu(13)=o_s*n_s*C_mf_s*R*T+o_a*(C_mf_a+C_mf_H)*R*T+o_a1*(C_mf_a1+C_mf_H)*R*T-P_m;
101 %Charge Balance Equation at membrane active layer %%%%%%%%%%%%%%%%%%%%%%%
102 fu(14)=C_mf_A+C_mf_A1+K_w_T/C_mf_H/gamma_mf_H/gamma_mf_H-C_mf_H;
103 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
104 fu(15)=A_I*(I_mf^(1/2)/(1+I_mf^(1/2))- 0.3*I_mf)+log10(gamma_mf_H);%"Davies"
105 fu(16)=K_a_T*(C_mf_a-C_mf_A)/gamma_mf_H/gamma_mf_H/C_mf_H-C_mf_A;% **additional↙
C_mf_A
106 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
107 fu(17)=K_a1_T*(C_mf_a1-C_mf_A1)/gamma_mf_H/gamma_mf_H/C_mf_H-C_mf_A1;
108 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
```

```
109  fu(18)=0.5*(2*C_mf_s+C_mf_H+K_w_T/C_mf_H/gamma_mf_H/gamma_mf_H+C_mf_A1+C_mf_A)-I_mf;
110
111  fu(19)=((C_md_s-C_i_s)*(12.625*10^(-11)*((0.789*C_i_s+0.211*C_md_s)^4 ...
112  /(J_w*(0.789*C_i_s+0.211*C_md_s)+ J_s)+(0.211*C_i_s+0.789*C_md_s)^4 ...
113  /(J_w*(0.211*C_i_s+0.789*C_md_s)+ J_s ))...
114  -3.885*10^(-9)*((0.789*C_i_s+0.211*C_md_s)^3 ...
115  /(J_w*(0.789*C_i_s+0.211*C_md_s)+ J_s)+(0.211*C_i_s+0.789*C_md_s)^3 ...
116  /(J_w*(0.211*C_i_s+0.789*C_md_s)+ J_s ))...
117  +6.975*10^(-8)*((0.789*C_i_s+0.211*C_md_s)^2 ...
118  /(J_w*(0.789*C_i_s+0.211*C_md_s)+ J_s)+(0.211*C_i_s+0.789*C_md_s)^2 ...
119  /(J_w*(0.211*C_i_s+0.789*C_md_s)+ J_s ))...
120  -4.03*10^(-7)*((0.789*C_i_s+0.211*C_md_s)...
121  /(J_w*(0.789*C_i_s+0.211*C_md_s)+ J_s)+(0.211*C_i_s+0.789*C_md_s)...
122  /(J_w*(0.211*C_i_s+0.789*C_md_s)+ J_s))...
123  +0.53*10^(-5)*(1/(J_w*(0.789*C_i_s+0.211*C_md_s)+J_s)...
124  +1/(J_w*(0.211*C_i_s+0.789*C_md_s)+ J_s))))-S;
125
126
127  fu(20)=((C_d_s-C_md_s)*(12.625*10^(-11)*((0.789*C_md_s+0.211*C_d_s)^4 ...
128  /(J_w*(0.789*C_md_s+0.211*C_d_s)+ J_s)+(0.211*C_md_s+0.789*C_d_s)^4 ...
129  /(J_w*(0.211*C_md_s+0.789*C_d_s)+ J_s ))...
130  -3.885*10^(-9)*((0.789*C_md_s+0.211*C_d_s)^3 ...
131  /(J_w*(0.789*C_md_s+0.211*C_d_s)+ J_s)+(0.211*C_md_s+0.789*C_d_s)^3 ...
132  /(J_w*(0.211*C_md_s+0.789*C_d_s)+ J_s ))...
133  +6.975*10^(-8)*((0.789*C_md_s+0.211*C_d_s)^2 ...
134  /(J_w*(0.789*C_md_s+0.211*C_d_s)+ J_s)+(0.211*C_md_s+0.789*C_d_s)^2 ...
135  /(J_w*(0.211*C_md_s+0.789*C_d_s)+ J_s ))...
136  -4.03*10^(-7)*((0.789*C_md_s+0.211*C_d_s)...
137  /(J_w*(0.789*C_md_s+0.211*C_d_s)+ J_s)+(0.211*C_md_s+0.789*C_d_s)...
138  /(J_w*(0.211*C_md_s+0.789*C_d_s)+ J_s))...
139  +0.53*10^(-5)*(1/(J_w*(0.789*C_md_s+0.211*C_d_s)+J_s)...
140  +1/(J_w*(0.211*C_md_s+0.789*C_d_s)+ J_s))))-D_d_s/k_d;
141
142  fu(21)=Sh_d*D_d_s/d_h-k_d;
143  fu(22)=5.05*10^(-11)*C_d_s^4-2.59*10^(-9)*C_d_s^3+4.65*10^(-8)*C_d_s^2-4.03*10^(-7)↵
*C_d_s+1.06*10^(-5)-D_d_s;
144  fu(23)=L*v_d*den_d/mul_d-Re_d;
145  fu(24)=(f_2+f)/2/W/D-v_d;
146  fu(25)=f+J_w*A_m-f_2;
147  fu(26)=0.0538*C_d_s^2+0.4159*C_d_s+5.0095-mul_d;
148  fu(27)=37.0166*C_d_s+997.2911-den_d;
149  fu(28)= 0.04*Re_d^(3/4)*Sc_d^(1/3)-Sh_d;
150  fu(29)= mul_d/den_d/D_d_s-Sc_d;
151  fu(30)=(0.5*(C_f_a-C_mf_a)*((D_1_T-D_HA_T)/...
152      (sqrt(1+4*K*(0.789*C_mf_a+0.211*C_f_a))*(J_w*(0.789*C_mf_a+0.211*C_f_a)-J_a))...
153      +(D_1_T-D_HA_T)/(sqrt(1+4*K*(0.211*C_mf_a+0.789*C_f_a))*(J_w*(0.211*C_mf_a...
154      +0.789*C_f_a)-J_a))+ D_HA_T/(J_w*(0.789*C_mf_a+0.211*C_f_a)-J_a)...
155      +D_HA_T/((J_w*(0.211*C_mf_a+0.789*C_f_a)-J_a))))+D_f_a/k_f_a;
156  %↵
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
157  fu(31)=(0.5*(C_f_a1-C_mf_a1)*((D_11_T-D_HA1_T)/...
158      (sqrt(1+4*K1*(0.789*C_mf_a1+0.211*C_f_a1))*(J_w*(0.789*C_mf_a1+0.211*C_f_a1)-↵
J_a1))...
159      +(D_11_T-D_HA1_T)/(sqrt(1+4*K1*(0.211*C_mf_a1+0.789*C_f_a1))*(J_w*(0.211↵
*C_mf_a1...
160      +0.789*C_f_a1)-J_a1))+ D_HA1_T/(J_w*(0.789*C_mf_a1+0.211*C_f_a1)-J_a1)...
```

```
161     +D_HA1_T/((J_w*(0.211*C_mf_a1+0.789*C_f_a1)-J_a1))))+D_f_a1/k_f_a1;
162 %↙
%^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
163 fu(32)=Sh_f_a*D_f_a/d_h-k_f_a;
164 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
165 fu(33)=Sh_f_a1*D_f_a1/d_h-k_f_a1;
166 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
167 fu(34)=(D_1_T/(2*K*C_f_a)*(-1+sqrt(1+4*K*C_f_a))+D_HA_T/...
168     (4*K*C_f_a)*(-1+sqrt(1+4*K*C_f_a))^2)-D_f_a;
169 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
170 fu(35)=(D_11_T/(2*K1*C_f_a1)*(-1+sqrt(1+4*K1*C_f_a1))+D_HA1_T/...
171     (4*K1*C_f_a1)*(-1+sqrt(1+4*K1*C_f_a1))^2)-D_f_a1;
172 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
173 fu(36)=L*v_f*den_f/mul_f -Re_f;
174 fu(37)=(f+f_1)/W/D/2-v_f;
175 fu(38)=f-J_w*A_m-f_1;
176 fu(39)= 0.04*Re_f^(3/4)*Sc_f_a^(1/3)-Sh_f_a;
177 fu(40)= mul_f/den_f/D_f_a-Sc_f_a;
178 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
179 fu(41)= 0.04*Re_f^(3/4)*Sc_f_a1^(1/3)-Sh_f_a1;
180 fu(42)= mul_f/den_f/D_f_a1-Sc_f_a1;
181 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
182
183 fu(43)=((C_f_s-C_mf_s)*(12.625*10^(-11)*((0.789*C_mf_s+0.211*C_f_s)^4 ...
184 /(J_w*(0.789*C_mf_s+0.211*C_f_s)+ J_s)+(0.211*C_mf_s+0.789*C_f_s)^4 ...
185 /(J_w*(0.211*C_mf_s+0.789*C_f_s)+ J_s ))...
186 -3.885*10^(-9)*((0.789*C_mf_s+0.211*C_f_s)^3 ...
187 /(J_w*(0.789*C_mf_s+0.211*C_f_s)+ J_s)+(0.211*C_mf_s+0.789*C_f_s)^3 ...
188 /(J_w*(0.211*C_mf_s+0.789*C_f_s)+ J_s ))...
189 +6.975*10^(-8)*((0.789*C_mf_s+0.211*C_f_s)^2 ...
190 /(J_w*(0.789*C_mf_s+0.211*C_f_s)+ J_s)+(0.211*C_mf_s+0.789*C_f_s)^2 ...
191 /(J_w*(0.211*C_mf_s+0.789*C_f_s)+ J_s )...
192 -4.03*10^(-7)*((0.789*C_mf_s+0.211*C_f_s)...
193 /(J_w*(0.789*C_mf_s+0.211*C_f_s)+ J_s)+(0.211*C_mf_s+0.789*C_f_s)...
194 /(J_w*(0.211*C_mf_s+0.789*C_f_s)+ J_s))...
195 +0.53*10^(-5)*(1/(J_w*(0.789*C_mf_s+0.211*C_f_s)+J_s)...
196 +1/(J_w*(0.211*C_mf_s+0.789*C_f_s)+ J_s))))+D_f_s/k_f_s;
197
198 fu(44)=Sh_f_s*D_f_s/d_h-k_f_s;
199 fu(45)=5.05*10^(-11)*C_f_s^4-2.59*10^(-9)*C_f_s^3+4.65*10^(-8)*C_f_s^2-4.03*10^(-7)↙
*C_f_s+1.06*10^(-5)-D_f_s;
200 %Mole balance Equation
201 fu(46)=(V_f_0*C_f_a_0+C_f_ai*f_1*t)/(V_f_0+f_1*t)-C_f_a;
202 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
203 fu(47)=(V_f_0*C_f_a1_0+C_f_ai1*f_1*t)/(V_f_0+f_1*t)-C_f_a1;
204 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
205 fu(48)=C_f_s*f_1*t/(V_f_0+f_1*t)-C_f_si;
206 fu(49)=(C_f_a*f-J_a*A_m)/f_1-C_f_ai;
207 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
208 fu(50)=(C_f_a1*f-J_a1*A_m)/f_1-C_f_ai1;
209 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
210 fu(51)=(C_f_si*f+J_s*A_m)/f_1-C_f_s;
211 fu(52)=(C_d_s_0*V_d_0+C_d_si*f_2*t)/(V_d_0+f_2*t)-C_d_s;
212 fu(53)=(C_d_ai*f*t+C_d_ai*(V_d_0+f_2*t-f*t))/(f_2*t)-C_d_a;
213 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
214 fu(54)=(C_d_ai1*f*t+C_d_ai1*(V_d_0+f_2*t-f*t))/(f_2*t)-C_d_a1;
```

```
215 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
216 fu(55)=(C_d_s*f-J_s*A_m)/f_2-C_d_si;
217 fu(56)=(C_d_a*f_2-J_a*A_m)/f-C_d_ai;
218 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
219 fu(57)=(C_d_a1*f_2-J_a1*A_m)/f-C_d_ai1;
220 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
221 %additional Equation
222 fu(58)= 0.04*Re_f^(3/4)*Sc_f_s^(1/3)-Sh_f_s;
223 fu(59)= mul_f/den_f/D_f_s-Sc_f_s;
224 fu(60)=J_w*A_m*t*den_d/1000-w;
225 %pH Equation..
226 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
227 fu(61)= pH_f+log10(gamma_f_H*C_f_H);
228 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
229 fu(62)=K_a_T*(C_f_a-C_f_A)/gamma_f_H/gamma_f_H/C_f_H-C_f_A;
230 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
231 fu(63)=K_a1_T*(C_f_a1-C_f_A1)/gamma_f_H/gamma_f_H/C_f_H-C_f_A1;
232 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
233 fu(64)=A_I*(I_f^(1/2)/(1+I_f^(1/2))-0.3*I_f)+log10(gamma_f_H);%Davies;
234 %ionic streanght in feed solution
235 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%5
236 fu(65)=0.5*(2*C_f_si+C_f_H+K_w_T/C_f_H/gamma_f_H/gamma_f_H+C_f_A+C_f_A1+C_f_HCO3c)-↙
I_f;
237↙
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
238 %Charge Balance Equation %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
239 fu(66)=(C_f_HCO3c+C_f_A+C_f_A1+K_w_T/C_f_H/gamma_f_H/gamma_f_H)-C_f_H;
240 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
241 %Co2 diffuse to feed solution
242 fu(67)=P_M*C_d_co2-J_co2;
243 fu(68)=C_f_co2i*(V_f_0+f_1*t-f*t)-(C_f_co2*f_1*t-C_f_co2i*f*t);
244 fu(69)=C_f_co2i*f*t+J_co2*A_m*t-C_f_co2*f_1*t;
245 %Carbonic acid produce in feed solutions
246 fu(70)=10^-3.45/gamma_f_H/gamma_f_H/C_f_H*(C_f_co2-C_f_HCO3c)-C_f_HCO3c;
247 % In draw solution
248 fu(71)=gamma_d_H*gamma_d_H*C_d_H*C_d_HCO3/K_CO2_T+C_d_HCO3-C_d_co2;%******
249 fu(72)=(-K_CO2_T+sqrt(K_CO2_T^2+4*K_CO2_T*gamma_d_H*gamma_d_H*C_d_co2a))/↙
(2*gamma_d_H*gamma_d_H)-C_d_HCO3;
250 %ionic streanght in draw
251 %↙
solution%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%↙
%
252 fu(73)=0.5*(2*C_d_si+C_d_H+K_w_T/C_d_H/gamma_d_H/gamma_d_H+C_d_A+C_d_A1+C_d_HCO3)-↙
I_d;
253↙
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%↙
%%
254 %Charge balance at draw solution%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
255 fu(74)=(C_d_HCO3+C_d_A+C_d_A1+K_w_T/C_d_H/gamma_d_H/gamma_d_H)-C_d_H;
256 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
257 fu(75)=K_a_T*(C_d_a-C_d_A)/gamma_d_H/gamma_d_H/C_d_H-C_d_A;
258 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%5
259 fu(76)=K_a1_T*(C_d_a1-C_d_A1)/gamma_d_H/gamma_d_H/C_d_H-C_d_A1;
260 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%5
261 fu(77)=10^-(A_I*(I_d^(1/2)/(1+I_d^(1/2))-0.3*I_d))-gamma_d_H;%Davies;
262 %Additional Equation From Interface
```

```
263 fu(78)=K_a_T*(C_i_a-C_i_A)/gamma_i_H/gamma_i_H/C_i_H-C_i_A;
264 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
265 fu(79)=K_a1_T*(C_i_a1-C_i_A1)/gamma_i_H/gamma_i_H/C_i_H-C_i_A1;
266 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
267 fu(80)=10^-3.408*K_H/10^((((33.5-0.109*(T-273)+0.0014*(T-273)^2)*I_d-(1.5+0.015*(T-↙
273)+0.004*(T-273)^2)*I_d^2))...
268     /T)-C_d_co2a;
269
270
271
```

G.14 A mixture of 10 mM acetic acid and 5 mM valeric acid as feed solution and NaCl as draw solution

```
 1 clear
 2 clc
 3
 4 [g]=[1.5394e-03   2.1309e+01   4.1219e-01   1.9404e-05   4.5491e-01   1.2623e-04↙
4.0138e-11   1.0126e-02   2.1835e-08   2.0592e-08...
 5 2.6293e-06   7.3060e-01   4.5487e-01   9.9974e-01   5.0557e-04   9.7173e-01   6.3180↙
e-04   9.8961e-01   9.5448e-06   1.4718e-01...
 6 6.7539e+02   5.5500e+02   2.6120e+04   1.5028e+02   1.0343e+03   5.4791e+00   1.5806↙
e+00   1.0595e-05   1.5607e-01   1.0001e-02...
 7 6.4517e+02   2.5450e+04   1.5022e+02   1.5794e+00   5.1292e+02   9.2585e-06   1.4265↙
e-01   6.7910e-06   1.0005e-02   1.6302e-06...
 8 9.6220e-09   9.9932e-01   3.3118e+00   9.7467e-01   3.5521e-04   5.0203e-04   5.4291↙
e+00   6.7484e+02   5.8698e+02   1.3375e-04...
 9 5.0040e-04   1.0376e-08   2.4907e-09   4.4346e-09   2.9648e-08   1.1859e-05   2.6010↙
e-06   9.9933e-01   9.9900e+02   1.8828e-08...
10 5.0978e-03   2.1132e-09   4.1929e-12   2.0790e-09   1.8639e-09   1.4740e-04   5.0006↙
e-03   1.0595e-05   1.5607e-01   6.4518e+02...
11 5.1294e+02   5.0027e-03   9.7144e-10   1.4519e-04   3.5816e-04   7.8989e-01   3.2817e-↙
05   3.2807e-05   -9.0963e-09   1.1759e-05];
12
13 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
14 ts=0.2:0.006:1800;
15 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
16
17 for ti=1:numel(ts);
18 t=ts(ti);
19 options = optimset('Algorithm','Levenberg-Marquardt','TolFun',1*10^-17,'TolX',1*10^-↙
17 ...
20      ,'Maxiter',25,'Maxfun',15000);
21
22 [r,fval,exitflag,output] = fsolve(@(z)Acetic10_Valeric5(z,t),[g],options)
23 [g]=[r];
24 j(ti)=r(1,1);%Jw
25 c(ti)=r(1,30);%C_f_a
26 d(ti)=r(1,50);%W
27 p(ti)=r(1,43);%pH
28 T(ti)=t;
29 e(ti)=r(1,67);%C_f_a1
30 % Rejection
31 O(ti)=(1-r(1,43)/r(1,32))*100;
32 K4(ti)=r(1,32)/0.01;
33 %for Plotting
34 K1(ti)=r(1,1);
35 [A(ti)]=t;
36 [B(ti)]=d(ti);%w
37 [C(ti)]=p(ti);%pH
38 % For Rejection
39 [M(ti)]=O(ti);
40 [N(ti)]=K1(ti);
41 [K5(ti)]=K4(ti);%Cfa/Cf0
42 disp(ti)
43 fprintf('\n     Time(min)          J_w(L/dm2/min)          pH          Cf,a(mole/l)↙
W(kg)          Cf,a1(mole/l)\n');
44 fprintf('\n%11.4f%20.4e%19.4f%19.4e%19.4e%19.4e%\n' ,T(ti),j(ti),p(ti),c(ti),d(ti),e↙
(ti));
45 format short e
```

```matlab
46 %Result of Best guess
47 fprintf('\n%8.3e%15.4e%13.3e%14.3e%11.3e%13.3e%13.3e%13.3e%13.3e%\n'...
48     ,r(1,1),r(1,2),r(1,3),r(1,4),r(1,5),r(1,6),r(1,7),r(1,8),r(1,9));
49
50  fprintf('\n%8.3e%15.4e%13.3e%14.3e%11.3e%13.3e%13.3e%13.3e%13.3e%\n'...
51     ,r(1,10),r(1,11),r(1,12),r(1,13),r(1,14),r(1,15),r(1,16),r(1,17),r(1,18));
52
53 fprintf('\n%8.3e%15.4e%13.3e%14.3e%11.3e%13.3e%13.3e%13.3e%13.3e%\n'...
54     ,r(1,19),r(1,20),r(1,21),r(1,22),r(1,23),r(1,24),r(1,25),r(1,26),r(1,27));
55
56 fprintf('\n%8.3e%15.4e%13.3e%14.3e%11.3e%13.3e%13.3e%13.3e%13.3e%\n'...
57     ,r(1,28),r(1,29),r(1,30),r(1,31),r(1,32),r(1,33),r(1,34),r(1,35),r(1,36));
58
59 fprintf('\n%8.3e%15.4e%13.3e%14.3e%11.3e%13.3e%13.3e%13.3e%13.3e%13.3e%\n'...
60     ,r(1,37),r(1,38),r(1,39),r(1,40),r(1,41),r(1,42),r(1,43),r(1,44),r(1,45));
61
62 fprintf('\n%8.3e%15.4e%13.3e%14.3e%11.3e%13.3e%13.3e%13.3e%13.3e%\n'...
63     ,r(1,46),r(1,47),r(1,48),r(1,49),r(1,50),r(1,51),r(1,52),r(1,53),r(1,54));
64
65 fprintf('\n%8.3e%15.4e%13.3e%14.3e%11.3e%13.3e%13.3e%13.3e%13.3e%\n'...
66     ,r(1,55),r(1,56),r(1,57),r(1,58),r(1,59),r(1,60),r(1,61),r(1,62),r(1,63));
67
68 fprintf('\n%8.3e%15.4e%13.3e%14.3e%11.3e%13.3e%13.3e%13.3e%13.3e%\n'...
69     ,r(1,64),r(1,65),r(1,66),r(1,67),r(1,68),r(1,69),r(1,70),r(1,71),r(1,72));
70
71 fprintf('\n%8.3e%15.4e%13.3e%14.3e%11.3e%13.3e%13.3e%13.3e%\n'...
72     ,r(1,73),r(1,74),r(1,75),r(1,76),r(1,77),r(1,78),r(1,79),r(1,80));
73
74 end
75
76 %Time Interval%%%%%%%%%%%%%%%%%%%%%%%%%
77 Ai=0:30:1800;
78 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
79 format short
80 Ci=spline(A,C,Ai);
81 format short e
82 Bi=spline(A,B,Ai);
83
84 %Experiment Value W and pH%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
85 F=[ 0      0.0150    0.0300    0.0450    0.0570    0.0720    0.0850    0.1000
0.1130    0.1270    0.1410    0.1540    0.1680...
86 0.1820    0.1940    0.2080    0.2210    0.2320    0.2450    0.2570    0.2690
0.2810    0.2930    0.3040    0.3160    0.3260...
87 0.3370    0.3490    0.3600    0.3700    0.3810    0.3890    0.3980    0.4100
0.4220    0.4300    0.4390    0.4490    0.4600...
88 0.4690    0.4780    0.4870    0.4965    0.5044    0.5125    0.5200    0.5275
0.5350    0.5420    0.5494    0.5570    0.5635...
89 0.5705    0.5770    0.5840    0.5900    0.5969    0.6025    0.6085    0.6150
0.6210];
90
91 G=[ 3.3600    3.3690    3.3480    3.3300    3.3200    3.3100    3.2990    3.2870
3.2730    3.2640    3.2560    3.2450    3.2340...
92 3.2240    3.2140    3.2010    3.1910    3.1800    3.1710    3.1590    3.1460
3.1340    3.1230    3.1120    3.1000    3.0890...
93 3.0770    3.0660    3.0560    3.0450    3.0350    3.0260    3.0120    3.0010
2.9910    2.9740    2.9670    2.9530    2.9430...
```

```
 94 2.9320    2.9220    2.9120    2.9030    2.8920    2.8810    2.8710    2.8640↙
 2.8550    2.8450    2.8350    2.8270    2.8190...
 95 2.8110    2.8050    2.7970    2.7880    2.7820    2.7780    2.7720    2.7670↙
 2.7670];
 96 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 97
 98 F_m=mean(F);
 99 G_m=mean(G);
100
101 sum1=0;sum2=0;
102 sum3=0;sum4=0;
103 %%% Number of Data %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
104 for k=1:1:61
105 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
106 sum1=(Bi(1,k)-F(1,k))^2+sum1;
107 sum2=(Bi(1,k)-F_m)^2+sum2;
108 sum3=(Ci(1,k)-G(1,k))^2+sum3;
109 sum4=(Ci(1,k)-G_m)^2+sum4;
110 end
111
112 format short
113 R2_W=1-sum1/sum2;
114 R2_pH=1-sum3/sum4;
115 MSE_W=sum1/k;
116 MSE_pH=sum3/k;
117 RMSE_W=sqrt(MSE_W)
118 RMSE_pH=sqrt(MSE_pH)
119 SEP_W=RMSE_W/F_m*100
120 SEP_pH=RMSE_pH/G_m*100
121
122 % Plotting
123 subplot(2,3,1)
124 plot([A]/60,[B],'-g',[Ai]/60,[F],'xr')
125 xlabel('Time (hr)')
126 ylabel('Weight (Kg)')
127 title('Weight Change as a Function of Time')
128 legend('Model','Experiment',2)
129
130 subplot(2,3,2)
131 plot([A]/60,[C],'-g',[Ai]/60,[G],'xr')
132 xlabel('Time (hr)')
133 ylabel('pH')
134 title('pH as a Function of Time')
135 legend('Model','Experiment',1)
136
137 subplot(2,3,3)
138 plot([A]/60,[M],'-g')
139 xlabel('Time (hr)')
140 ylabel('%Rejection')
141 title('%Rejection as a Function of Time')
142 legend('Model',1)
143 %text(2,3.3638,'R2=    MSE=')
144
145 subplot(2,3,4)
146 plot([N],[M],'-g')
147 xlabel('Jw (L/dm2/min)')
```

```
148 ylabel('%Rejection')
149 title('%Rejection as a Function of Jw')
150 legend('Model',1)
151 %text(2,3.3638,'R2=    MSE=')
152
153 subplot(2,3,5)
154 plot([A]/60,[K5],'-g')
155 xlabel('Time (hr)')
156 ylabel('Cfa/Cf0')
157 title('Cfa/Cf0 as a Function of Time')
158 legend('Model',1)
159 %text(2,3.3638,'R2=    MSE=')
160
161 subplot(2,3,6)
162 plot([N],[K5],'-g')
163 xlabel('Jw (L/dm2/min)')
164 ylabel('Cfa/Cf0')
165 title('Cfa/Cf0 as a Function of Jw')
166 legend('Model',1)
167 %text(2,3.3638,'R2=    MSE=')
168
169 fprintf('\n      \n')
170 fprintf('\n      \n')
171 fprintf('\n CONCLUSION    \n')
172 fprintf('\n       R2_W             MSE_W                    R2_pH           MSE_pH\n');
173 fprintf('\n%11.3f%20.4e%19.4f%19.4e%\n',R2_W,MSE_W,R2_pH,MSE_pH)
174 fprintf('\n      \n')
175 fprintf('\n      \n')
176 fprintf('\n     Time(min)       J_w(L/dm2/min)              pH              Cf,a(mole/l)↙
W(kg)         C_f_a1\n');
177 for x=5:5:200000000;
178 z=x*150;
179 fprintf('\n%11.3f%20.4e%19.4f%19.4e%19.4e%19.4e%\n',T(z),j(z),p(z),c(z),d(z),e(z));
180
181 end
182
183
184
```

```
1
2  function fu=Acetic10_Valeric5(z,t)
3  %Independent Variables
4  V_f_0=1;C_d_s_0=1;V_d_0=0.5;
5  %Fixed Variable
6  T=301;f=1.58;
7  L=0.9207;D=0.023;W=0.4572;A_m=42*10^(-2);d_h=0.0438;
8  % Acid Variables
9  %
10 C_f_a_0=10*10^(-3);
11 C_f_a1_0=5*10^(-3);
12 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
13 B_a=2.093/60/100;% B of Acetic Acid
14 B_a1=0.491/60/100;% B of Valeric Acid
15 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
16 D_A_298=6.534*10^(-6);D_HA_298=7.74*10^(-6);%Acetic Physicocmical Property
17 D_A1_298=5.226*10^(-6);D_HA1_298=4.902*10^(-6);%Valeric Physicocmical Property
18 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
19 K_a_T=10^-(1170.48/T-3.1649+0.013399*T);%Acetic acid
20 K_a1_T=10^-(921.38/T-1.8574+0.012105*T);%Valeric Acid
21 %
22 %Constant Variable
23 B_s=0.256/60/100;% normal..B_s(0.256) =4.27*10^(-5)
24 A=0.442/60/100;%..(0.413) New Value
25 o_s=0.936;
26 S=500*10^-5;
27 D_H_298=5.587*10^(-5);
28 visco_298=((298-273)+246)/((0.05594*(298-273)+5.2842)*(298-273)+137.37);
29 visco_T=((T-273)+246)/((0.05594*(T-273)+5.2842)*(T-273)+137.37);
30 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
31 o_a=1;n_s=2;R=0.08314;
32 o_a1=1;
33 %Temperature Correction
34 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
35 K_CO2_T=10^-(3404.71/T-14.8435+0.032786*T);
36 K_w_T=10^(-4470.99/T+6.0875-0.017060*T);
37 E=2727.586+0.6224107*T-466.9151*log(T)-52000.87/T;
38 d=1-(((T-273)-3.9863)^2*((T-273)+288.9414))/(508929.2*((T-273)+68.12963))...
39 +0.011445*exp((-374.3)/(T-273));
40 A_I=(1.82483*10^6*d^0.5)/(E*T)^1.5;
41 K=1/K_a_T;
42 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
43 K1=1/K_a1_T;
44 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
45 D_H_T=D_H_298*T/298*visco_298/visco_T ;
46 D_A_T=D_A_298*T/298*visco_298/visco_T;
47 D_HA_T=D_HA_298*T/298*visco_298/visco_T;
48 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
49 D_A1_T=D_A1_298*T/298*visco_298/visco_T;
50 D_HA1_T=D_HA1_298*T/298*visco_298/visco_T;
51 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
52 D_1_T=2/(1/D_H_T +1/D_A_T);
53 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
```

```
54 D_11_T=2/(1/D_H_T +1/D_A1_T);
55 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
56 K_H=0.034*exp(-2400*(1/T-1/298));%************************
57 %For carbonic acid
58 P_M =0.0025;
59 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
60 %Unknown Variable
61 J_w=z(1);P_i=z(2);P_m=z(3);J_s=z(4);C_i_s=z(5);
62 C_mf_s=z(6);J_a=z(7);C_mf_a=z(8);C_i_a=z(9);C_d_a=z(10);
63 C_d_H=z(11);gamma_i_H=z(12);I_i=z(13);C_d_s=z(14);
64 C_mf_H=z(15);gamma_mf_H=z(16);I_mf=z(17);C_md_s=z(18);
65 D_d_s=z(19);k_d=z(20);Sh_d=z(21);Sc_d=z(22);Re_d=z(23);
66 v_d=z(24);den_d=z(25);mul_d=z(26);f_2=z(27);D_f_a=z(28);
67 k_f_a=z(29);C_f_a=z(30);Sh_f_a=z(31);Re_f=z(32);v_f=z(33);
68 f_1=z(34);Sc_f_a=z(35);D_f_s=z(36);k_f_s=z(37);C_f_s=z(38);
69 C_f_ai=z(39);C_f_si=z(40);C_d_ai=z(41);C_d_si=z(42);pH_f=z(43);
70 gamma_f_H=z(44);C_f_A=z(45);I_f=z(46);mul_f=z(47);Sh_f_s=z(48);
71 Sc_f_s=z(49);w=z(50);C_f_H=z(51);C_f_co2=z(52);C_f_co2i=z(53);
72 C_f_HCO3c=z(54);J_co2=z(55);C_d_co2=z(56);C_d_HCO3=z(57);
73 I_d=z(58);den_f=z(59);C_d_A=z(60);
74 %For 2 acid
75 C_mf_a1=z(61);C_i_a1=z(62);J_a1=z(63);C_d_a1=z(64);C_d_A1=z(65);C_mf_A1=z(66);
76 C_f_a1=z(67);D_f_a1=z(68);k_f_a1=z(69);Sh_f_a1=z(70);Sc_f_a1=z(71);
77 C_f_ai1=z(72);C_d_ai1=z(73);C_f_A1=z(74);C_mf_A=z(75);gamma_d_H=z(76);C_i_H=z(77);
78 C_i_A=z(78);C_i_A1=z(79);C_d_co2a=z(80);
79 %Math Model
80 fu(1)=A*(P_i-P_m )-J_w;
81 fu(2)= B_s *(C_i_s-C_mf_s)-J_s;
82 fu(3)=B_a*(C_mf_a-C_i_a)-J_a;
83 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
84 fu(4)= B_a1*(C_mf_a1-C_i_a1)-J_a1;
85 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
86 fu(5)=C_d_a-C_i_a;
87 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
88 fu(6)=C_d_a1-C_i_a1;
89 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
90 fu(7)=J_a/J_w-C_d_a;
91 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
92 fu(8)=J_a1/J_w-C_d_a1;
93 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
94 fu(9)= o_s*n_s*C_i_s*R*T+o_a*(C_i_a+C_i_H)*R*T+o_a1*(C_i_a1+C_i_H)*R*T-P_i;
95 %Charge Balance Equation at interface %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
96 fu(10)=C_i_A+C_i_A1+K_w_T/C_i_H/gamma_i_H/gamma_i_H-C_i_H;
97 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
98 fu(11)=A_I*(I_i^(1/2)/(1+I_i^(1/2))- 0.3*I_i)+log10(gamma_i_H);%"Davies"
99 fu(12)=0.5*(2*C_i_s+C_i_H+C_i_A+C_i_A1+K_w_T/C_i_H/gamma_i_H/gamma_i_H)- I_i;
100 fu(13)=o_s*n_s*C_mf_s*R*T+o_a*(C_mf_a+C_mf_H)*R*T+o_a1*(C_mf_a1+C_mf_H)*R*T-P_m;
101 %Charge Balance Equation at membrane active layer %%%%%%%%%%%%%%%%%%%%%%%%%
102 fu(14)=C_mf_A+C_mf_A1+K_w_T/C_mf_H/gamma_mf_H/gamma_mf_H-C_mf_H;
103 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
104 fu(15)=A_I*(I_mf^(1/2)/(1+I_mf^(1/2))- 0.3*I_mf)+log10(gamma_mf_H);%"Davies"
105 fu(16)=K_a_T*(C_mf_a-C_mf_A)/gamma_mf_H/gamma_mf_H/C_mf_H-C_mf_A;% **additional↙
C_mf_A
106 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
107 fu(17)=K_a1_T*(C_mf_a1-C_mf_A1)/gamma_mf_H/gamma_mf_H/C_mf_H-C_mf_A1;
108 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
```

```
109 fu(18)=0.5*(2*C_mf_s+C_mf_H+K_w_T/C_mf_H/gamma_mf_H/gamma_mf_H+C_mf_A1+C_mf_A)-I_mf;
110
111 fu(19)=((C_md_s-C_i_s)*(12.625*10^(-11)*((0.789*C_i_s+0.211*C_md_s)^4 ...
112 /(J_w*(0.789*C_i_s+0.211*C_md_s)+ J_s)+(0.211*C_i_s+0.789*C_md_s)^4 ...
113 /(J_w*(0.211*C_i_s+0.789*C_md_s)+ J_s ))...
114 -3.885*10^(-9)*((0.789*C_i_s+0.211*C_md_s)^3 ...
115 /(J_w*(0.789*C_i_s+0.211*C_md_s)+ J_s)+(0.211*C_i_s+0.789*C_md_s)^3 ...
116 /(J_w*(0.211*C_i_s+0.789*C_md_s)+ J_s ))...
117 +6.975*10^(-8)*((0.789*C_i_s+0.211*C_md_s)^2 ...
118 /(J_w*(0.789*C_i_s+0.211*C_md_s)+ J_s)+(0.211*C_i_s+0.789*C_md_s)^2 ...
119 /(J_w*(0.211*C_i_s+0.789*C_md_s)+ J_s ))...
120 -4.03*10^(-7)*((0.789*C_i_s+0.211*C_md_s)...
121 /(J_w*(0.789*C_i_s+0.211*C_md_s)+ J_s)+(0.211*C_i_s+0.789*C_md_s)...
122 /(J_w*(0.211*C_i_s+0.789*C_md_s)+ J_s))...
123 +0.53*10^(-5)*(1/(J_w*(0.789*C_i_s+0.211*C_md_s)+J_s)...
124 +1/(J_w*(0.211*C_i_s+0.789*C_md_s)+ J_s))))-S;
125
126
127 fu(20)=((C_d_s-C_md_s)*(12.625*10^(-11)*((0.789*C_md_s+0.211*C_d_s)^4 ...
128 /(J_w*(0.789*C_md_s+0.211*C_d_s)+ J_s)+(0.211*C_md_s+0.789*C_d_s)^4 ...
129 /(J_w*(0.211*C_md_s+0.789*C_d_s)+ J_s ))...
130 -3.885*10^(-9)*((0.789*C_md_s+0.211*C_d_s)^3 ...
131 /(J_w*(0.789*C_md_s+0.211*C_d_s)+ J_s)+(0.211*C_md_s+0.789*C_d_s)^3 ...
132 /(J_w*(0.211*C_md_s+0.789*C_d_s)+ J_s ))...
133 +6.975*10^(-8)*((0.789*C_md_s+0.211*C_d_s)^2 ...
134 /(J_w*(0.789*C_md_s+0.211*C_d_s)+ J_s)+(0.211*C_md_s+0.789*C_d_s)^2 ...
135 /(J_w*(0.211*C_md_s+0.789*C_d_s)+ J_s ))...
136 -4.03*10^(-7)*((0.789*C_md_s+0.211*C_d_s)...
137 /(J_w*(0.789*C_md_s+0.211*C_d_s)+ J_s)+(0.211*C_md_s+0.789*C_d_s)...
138 /(J_w*(0.211*C_md_s+0.789*C_d_s)+ J_s))...
139 +0.53*10^(-5)*(1/(J_w*(0.789*C_md_s+0.211*C_d_s)+J_s)...
140 +1/(J_w*(0.211*C_md_s+0.789*C_d_s)+ J_s))))-D_d_s/k_d;
141
142 fu(21)=Sh_d*D_d_s/d_h-k_d;
143 fu(22)=5.05*10^(-11)*C_d_s^4-2.59*10^(-9)*C_d_s^3+4.65*10^(-8)*C_d_s^2-4.03*10^(-7)
*C_d_s+1.06*10^(-5)-D_d_s;
144 fu(23)=L*v_d*den_d/mul_d-Re_d;
145 fu(24)=(f_2+f)/2/W/D-v_d;
146 fu(25)=f+J_w*A_m-f_2;
147 fu(26)=0.0538*C_d_s^2+0.4159*C_d_s+5.0095-mul_d;
148 fu(27)=37.0166*C_d_s+997.2911-den_d;
149 fu(28)= 0.04*Re_d^(3/4)*Sc_d^(1/3)-Sh_d;
150 fu(29)= mul_d/den_d/D_d_s-Sc_d;
151 fu(30)=(0.5*(C_f_a-C_mf_a)*((D_1_T-D_HA_T)/...
152     (sqrt(1+4*K*(0.789*C_mf_a+0.211*C_f_a))*(J_w*(0.789*C_mf_a+0.211*C_f_a)-J_a))...
153     +(D_1_T-D_HA_T)/(sqrt(1+4*K*(0.211*C_mf_a+0.789*C_f_a))*(J_w*(0.211*C_mf_a...
154     +0.789*C_f_a)-J_a))+ D_HA_T/(J_w*(0.789*C_mf_a+0.211*C_f_a)-J_a)...
155     +D_HA_T/((J_w*(0.211*C_mf_a+0.789*C_f_a)-J_a))))+D_f_a/k_f_a;
156 %
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
157 fu(31)=(0.5*(C_f_a1-C_mf_a1)*((D_11_T-D_HA1_T)/...
158     (sqrt(1+4*K1*(0.789*C_mf_a1+0.211*C_f_a1))*(J_w*(0.789*C_mf_a1+0.211*C_f_a1)-
J_a1))...
159     +(D_11_T-D_HA1_T)/(sqrt(1+4*K1*(0.211*C_mf_a1+0.789*C_f_a1))*(J_w*(0.211
*C_mf_a1...
160     +0.789*C_f_a1)-J_a1))+ D_HA1_T/(J_w*(0.789*C_mf_a1+0.211*C_f_a1)-J_a1)...
```

```
161     +D_HA1_T/((J_w*(0.211*C_mf_a1+0.789*C_f_a1)-J_a1))))+D_f_a1/k_f_a1;
162 %↙
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
163 fu(32)=Sh_f_a*D_f_a/d_h-k_f_a;
164 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
165 fu(33)=Sh_f_a1*D_f_a1/d_h-k_f_a1;
166 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
167 fu(34)=(D_1_T/(2*K*C_f_a)*(-1+sqrt(1+4*K*C_f_a))+D_HA_T/...
168     (4*K*C_f_a)*(-1+sqrt(1+4*K*C_f_a))^2)-D_f_a;
169 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
170 fu(35)=(D_11_T/(2*K1*C_f_a1)*(-1+sqrt(1+4*K1*C_f_a1))+D_HA1_T/...
171     (4*K1*C_f_a1)*(-1+sqrt(1+4*K1*C_f_a1))^2)-D_f_a1;
172 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
173 fu(36)=L*v_f*den_f/mul_f -Re_f;
174 fu(37)=(f+f_1)/W/D/2-v_f;
175 fu(38)=f-J_w*A_m-f_1;
176 fu(39)= 0.04*Re_f^(3/4)*Sc_f_a^(1/3)-Sh_f_a;
177 fu(40)= mul_f/den_f/D_f_a-Sc_f_a;
178 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
179 fu(41)= 0.04*Re_f^(3/4)*Sc_f_a1^(1/3)-Sh_f_a1;
180 fu(42)= mul_f/den_f/D_f_a1-Sc_f_a1;
181 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
182
183 fu(43)=((C_f_s-C_mf_s)*(12.625*10^(-11)*((0.789*C_mf_s+0.211*C_f_s)^4 ...
184 /(J_w*(0.789*C_mf_s+0.211*C_f_s)+ J_s)+(0.211*C_mf_s+0.789*C_f_s)^4 ...
185 /(J_w*(0.211*C_mf_s+0.789*C_f_s)+ J_s ))...
186 -3.885*10^(-9)*((0.789*C_mf_s+0.211*C_f_s)^3 ...
187 /(J_w*(0.789*C_mf_s+0.211*C_f_s)+ J_s)+(0.211*C_mf_s+0.789*C_f_s)^3 ...
188 /(J_w*(0.211*C_mf_s+0.789*C_f_s)+ J_s ))...
189 +6.975*10^(-8)*((0.789*C_mf_s+0.211*C_f_s)^2 ...
190 /(J_w*(0.789*C_mf_s+0.211*C_f_s)+ J_s)+(0.211*C_mf_s+0.789*C_f_s)^2 ...
191 /(J_w*(0.211*C_mf_s+0.789*C_f_s)+ J_s ))...
192 -4.03*10^(-7)*((0.789*C_mf_s+0.211*C_f_s)...
193 /(J_w*(0.789*C_mf_s+0.211*C_f_s)+ J_s)+(0.211*C_mf_s+0.789*C_f_s)...
194 /(J_w*(0.211*C_mf_s+0.789*C_f_s)+ J_s))...
195 +0.53*10^(-5)*(1/(J_w*(0.789*C_mf_s+0.211*C_f_s)+J_s)...
196 +1/(J_w*(0.211*C_mf_s+0.789*C_f_s)+ J_s))))+D_f_s/k_f_s;
197
198 fu(44)=Sh_f_s*D_f_s/d_h-k_f_s;
199 fu(45)=5.05*10^(-11)*C_f_s^4-2.59*10^(-9)*C_f_s^3+4.65*10^(-8)*C_f_s^2-4.03*10^(-7)↙
*C_f_s+1.06*10^(-5)-D_f_s;
200 %Mole balance Equation
201 fu(46)=(V_f_0*C_f_a_0+C_f_ai*f_1*t)/(V_f_0+f_1*t)-C_f_a;
202 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
203 fu(47)=(V_f_0*C_f_a1_0+C_f_ai1*f_1*t)/(V_f_0+f_1*t)-C_f_a1;
204 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
205 fu(48)=C_f_s*f_1*t/(V_f_0+f_1*t)-C_f_si;
206 fu(49)=(C_f_a*f-J_a*A_m)/f_1-C_f_ai;
207 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
208 fu(50)=(C_f_a1*f-J_a1*A_m)/f_1-C_f_ai1;
209 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
210 fu(51)=(C_f_si*f+J_s*A_m)/f_1-C_f_s;
211 fu(52)=(C_d_s_0*V_d_0+C_d_si*f_2*t)/(V_d_0+f_2*t)-C_d_s;
212 fu(53)=(C_d_ai*f*t+C_d_ai*(V_d_0+f_2*t-f*t))/(f_2*t)-C_d_a;
213 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
214 fu(54)=(C_d_ai1*f*t+C_d_ai1*(V_d_0+f_2*t-f*t))/(f_2*t)-C_d_a1;
```

```
215 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
216 fu(55)=(C_d_s*f-J_s*A_m)/f_2-C_d_si;
217 fu(56)=(C_d_a*f_2-J_a*A_m)/f-C_d_ai;
218 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
219 fu(57)=(C_d_a1*f_2-J_a1*A_m)/f-C_d_ai1;
220 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
221 %additional Equation
222 fu(58)= 0.04*Re_f^(3/4)*Sc_f_s^(1/3)-Sh_f_s;
223 fu(59)= mul_f/den_f/D_f_s-Sc_f_s;
224 fu(60)=J_w*A_m*t*den_d/1000-w;
225 %pH Equation..
226 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
227 fu(61)= pH_f+log10(gamma_f_H*C_f_H);
228 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
229 fu(62)=K_a_T*(C_f_a-C_f_A)/gamma_f_H/gamma_f_H/C_f_H-C_f_A;
230 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
231 fu(63)=K_a1_T*(C_f_a1-C_f_A1)/gamma_f_H/gamma_f_H/C_f_H-C_f_A1;
232 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
233 fu(64)=A_I*(I_f^(1/2)/(1+I_f^(1/2))-0.3*I_f)+log10(gamma_f_H);%Davies;
234 %ionic streanght in feed solution
235 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%5
236 fu(65)=0.5*(2*C_f_si+C_f_H+K_w_T/C_f_H/gamma_f_H/gamma_f_H+C_f_A+C_f_A1+C_f_HCO3c)-↙
I_f;
237↙
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
238 %Charge Balance Equation %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
239 fu(66)=(C_f_HCO3c+C_f_A+C_f_A1+K_w_T/C_f_H/gamma_f_H/gamma_f_H)-C_f_H;
240 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
241 %Co2 diffuse to feed solution
242 fu(67)=P_M*C_d_co2-J_co2;
243 fu(68)=C_f_co2i*(V_f_0+f_1*t-f*t)-(C_f_co2*f_1*t-C_f_co2i*f*t);
244 fu(69)=C_f_co2i*f*t+J_co2*A_m*t-C_f_co2*f_1*t;
245 %Carbonic acid produce in feed solutions
246 fu(70)=10^-3.45/gamma_f_H/gamma_f_H/C_f_H*(C_f_co2-C_f_HCO3c)-C_f_HCO3c;
247 % In draw solution
248 fu(71)=gamma_d_H*gamma_d_H*C_d_H*C_d_HCO3/K_CO2_T+C_d_HCO3-C_d_co2;%******
249 fu(72)=(-K_CO2_T+sqrt(K_CO2_T^2+4*K_CO2_T*gamma_d_H*gamma_d_H*C_d_co2a))/↙
(2*gamma_d_H*gamma_d_H)-C_d_HCO3;
250 %ionic streanght in draw
251 %↙
solution%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%↙
%
252 fu(73)=0.5*(2*C_d_si+C_d_H+K_w_T/C_d_H/gamma_d_H/gamma_d_H+C_d_A+C_d_A1+C_d_HCO3)-↙
I_d;
253↙
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%↙
%%
254 %Charge balance at draw solution%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
255 fu(74)=(C_d_HCO3+C_d_A+C_d_A1+K_w_T/C_d_H/gamma_d_H/gamma_d_H)-C_d_H;
256 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
257 fu(75)=K_a_T*(C_d_a-C_d_A)/gamma_d_H/gamma_d_H/C_d_H-C_d_A;
258 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%5
259 fu(76)=K_a1_T*(C_d_a1-C_d_A1)/gamma_d_H/gamma_d_H/C_d_H-C_d_A1;
260 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%5
261 fu(77)=10^-(A_I*(I_d^(1/2)/(1+I_d^(1/2))-0.3*I_d))-gamma_d_H;%Davies;
262 %Additional Equation From Interface
```

```
263 fu(78)=K_a_T*(C_i_a-C_i_A)/gamma_i_H/gamma_i_H/C_i_H-C_i_A;
264 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
265 fu(79)=K_a1_T*(C_i_a1-C_i_A1)/gamma_i_H/gamma_i_H/C_i_H-C_i_A1;
266 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
267 fu(80)=10^-3.408*K_H/10^((((33.5-0.109*(T-273)+0.0014*(T-273)^2)*I_d-(1.5+0.015*(T-↙
273)+0.004*(T-273)^2)*I_d^2))...
268     /T)-C_d_co2a;
269
270
271
```

G.15 A mixture of 5 mM acetic acid and 10 mM valeric acid as feed solution and NaCl as draw solution

```
 1 clear
 2 clc
 3 [g]=[ 1.5394e-03   2.1310e+01   4.1227e-01   1.9404e-05   4.5491e-01   1.2623e-04
1.9969e-11   5.0626e-03   1.0859e-08   1.0243e-08...
 4 2.6217e-06   7.3060e-01   4.5488e-01   9.9974e-01   4.8905e-04   9.7209e-01   6.1528
e-04   9.8961e-01   9.5448e-06   1.4718e-01...
 5 6.7539e+02   5.5500e+02   2.6120e+04   1.5028e+02   1.0343e+03   5.4791e+00   1.5806
e+00   1.0595e-05   1.5607e-01   5.0006e-03...
 6 6.4517e+02   2.5450e+04   1.5022e+02   1.5794e+00   5.1292e+02   9.2585e-06   1.4265
e-01   6.7909e-06   5.0027e-03   1.6301e-06...
 7 4.7862e-09   9.9932e-01   3.3266e+00   9.7509e-01   1.8345e-04   4.8508e-04   5.4291
e+00   6.7484e+02   5.8698e+02   1.3375e-04...
 8 4.8345e-04   1.0353e-08   2.4851e-09   4.5100e-09   2.9581e-08   1.1832e-05   2.6010
e-06   9.9933e-01   9.9900e+02   9.3675e-09...
 9 1.0197e-02   4.2187e-09   8.3702e-12   4.1503e-09   3.7220e-09   3.0429e-04   1.0001
e-02   1.0595e-05   1.5607e-01   6.4517e+02...
10 5.1293e+02   1.0005e-02   1.9393e-09   3.0000e-04   1.8477e-04   7.8989e-01  -2.6634
e-05   5.7807e-08  -2.6691e-05   1.1759e-05];
11 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
12 ts=0.2:0.006:1800;
13 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
14
15 for ti=1:numel(ts);
16 t=ts(ti);
17 options = optimset('Algorithm','Levenberg-Marquardt','TolFun',1*10^-17,'TolX',1*10^-
17 ...
18      ,'Maxiter',25,'Maxfun',15000);
19
20 [r,fval,exitflag,output] = fsolve(@(z)Acetic5_Valeric10(z,t),[g],options)
21 [g]=[r];
22 j(ti)=r(1,1);%Jw
23 c(ti)=r(1,30);%C_f_a
24 d(ti)=r(1,50);%W
25 p(ti)=r(1,43);%pH
26 T(ti)=t;
27 e(ti)=r(1,67);%C_f_a1
28 % Rejection
29 O(ti)=(1-r(1,43)/r(1,32))*100;
30 K4(ti)=r(1,32)/0.01;
31 %for Plotting
32 K1(ti)=r(1,1);
33 [A(ti)]=t;
34 [B(ti)]=d(ti);%w
35 [C(ti)]=p(ti);%pH
36 % For Rejection
37 [M(ti)]=O(ti);
38 [N(ti)]=K1(ti);
39 [K5(ti)]=K4(ti);%Cfa/Cf0
40 disp(ti)
41 fprintf('\n    Time(min)       J_w(L/dm2/min)            pH            Cf,a(mole/l)
W(kg)          Cf,a1(mole/l)\n');
42 fprintf('\n%11.4f%20.4e%19.4f%19.4e%19.4e%19.4e%\n' ,T(ti),j(ti),p(ti),c(ti),d(ti),e
(ti));
43 format short e
44 %Result of Best guess
45 fprintf('\n%8.3e%15.4e%13.3e%14.3e%11.3e%13.3e%13.3e%13.3e%13.3e%\n'...
```

```
46        ,r(1,1),r(1,2),r(1,3),r(1,4),r(1,5),r(1,6),r(1,7),r(1,8),r(1,9));
47
48   fprintf('\n%8.3e%15.4e%13.3e%14.3e%11.3e%13.3e%13.3e%13.3e\n'...
49        ,r(1,10),r(1,11),r(1,12),r(1,13),r(1,14),r(1,15),r(1,16),r(1,17),r(1,18));
50
51 fprintf('\n%8.3e%15.4e%13.3e%14.3e%11.3e%13.3e%13.3e%13.3e\n'...
52        ,r(1,19),r(1,20),r(1,21),r(1,22),r(1,23),r(1,24),r(1,25),r(1,26),r(1,27));
53
54 fprintf('\n%8.3e%15.4e%13.3e%14.3e%11.3e%13.3e%13.3e%13.3e\n'...
55        ,r(1,28),r(1,29),r(1,30),r(1,31),r(1,32),r(1,33),r(1,34),r(1,35),r(1,36));
56
57 fprintf('\n%8.3e%15.4e%13.3e%14.3e%11.3e%13.3e%13.3e%13.3e%13.3e%13.3e\n'...
58        ,r(1,37),r(1,38),r(1,39),r(1,40),r(1,41),r(1,42),r(1,43),r(1,44),r(1,45));
59
60 fprintf('\n%8.3e%15.4e%13.3e%14.3e%11.3e%13.3e%13.3e%13.3e%13.3e\n'...
61        ,r(1,46),r(1,47),r(1,48),r(1,49),r(1,50),r(1,51),r(1,52),r(1,53),r(1,54));
62
63 fprintf('\n%8.3e%15.4e%13.3e%14.3e%11.3e%13.3e%13.3e%13.3e%13.3e\n'...
64        ,r(1,55),r(1,56),r(1,57),r(1,58),r(1,59),r(1,60),r(1,61),r(1,62),r(1,63));
65
66 fprintf('\n%8.3e%15.4e%13.3e%14.3e%11.3e%13.3e%13.3e%13.3e%13.3e\n'...
67        ,r(1,64),r(1,65),r(1,66),r(1,67),r(1,68),r(1,69),r(1,70),r(1,71),r(1,72));
68
69 fprintf('\n%8.3e%15.4e%13.3e%14.3e%11.3e%13.3e%13.3e%13.3e\n'...
70        ,r(1,73),r(1,74),r(1,75),r(1,76),r(1,77),r(1,78),r(1,79),r(1,80));
71
72 end
73
74 %Time Interval%%%%%%%%%%%%%%%%%%%%%%%%%
75 Ai=0:30:1800;
76 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
77 format short
78 Ci=spline(A,C,Ai);
79 format short e
80 Bi=spline(A,B,Ai);
81 %Experiment Value W and pH%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
82 F=[ 0    0.0150    0.0310    0.0460    0.0630    0.0760    0.0910    0.1040    0.1190↙
0.1320    0.1460    0.1600    0.1730...
83 0.1860    0.1990    0.2130    0.2250    0.2380    0.2500    0.2620    0.2730↙
0.2850    0.2960    0.3080    0.3190    0.3300...
84 0.3410    0.3510    0.3620    0.3720    0.3820    0.3930    0.4030    0.4120↙
0.4220    0.4320    0.4420    0.4520    0.4610...
85 0.4700    0.4790    0.4890    0.4980    0.5070    0.5160    0.5240    0.5340↙
0.5420    0.5500    0.5590    0.5670    0.5760...
86 0.5840    0.5920    0.6010    0.6090    0.6160    0.6240    0.6290    0.6290↙
0.6290];
87
88 G=[ 3.3110    3.2990    3.2890    3.2810    3.2680    3.2610    3.2530    3.2480↙
3.2380    3.2300    3.2210    3.2130    3.2040...
89 3.1950    3.1880    3.1790    3.1690    3.1590    3.1500    3.1410    3.1330↙
3.1240    3.1140    3.1060    3.0960    3.0880...
90 3.0770    3.0680    3.0580    3.0500    3.0420    3.0330    3.0250    3.0180↙
3.0100    3.0030    2.9960    2.9890    2.9820...
91 2.9750    2.9680    2.9620    2.9560    2.9510    2.9450    2.9370    2.9300↙
2.9230    2.9180    2.9110    2.9060    2.9010...
92 2.8950    2.8910    2.8880    2.8790    2.8750    2.8730    2.8670    2.8620↙
```

```matlab
2.8590];
 93 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 94 F_m=mean(F);
 95 G_m=mean(G);
 96
 97 sum1=0;sum2=0;
 98 sum3=0;sum4=0;
 99 %%% Number of Data %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
100 for k=1:1:61
101 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
102 sum1=(Bi(1,k)-F(1,k))^2+sum1;
103 sum2=(Bi(1,k)-F_m)^2+sum2;
104 sum3=(Ci(1,k)-G(1,k))^2+sum3;
105 sum4=(Ci(1,k)-G_m)^2+sum4;
106 end
107
108 format short
109 R2_W=1-sum1/sum2;
110 R2_pH=1-sum3/sum4;
111 MSE_W=sum1/k;
112 MSE_pH=sum3/k;
113 RMSE_W=sqrt(MSE_W)
114 RMSE_pH=sqrt(MSE_pH)
115 SEP_W=RMSE_W/F_m*100
116 SEP_pH=RMSE_pH/G_m*100
117
118 % Plotting
119 subplot(2,3,1)
120 plot([A]/60,[B],'-g',[Ai]/60,[F],'xr')
121 xlabel('Time (hr)')
122 ylabel('Weight (Kg)')
123 title('Weight Change as a Function of Time')
124 legend('Model','Experiment',2)
125
126 subplot(2,3,2)
127 plot([A]/60,[C],'-g',[Ai]/60,[G],'xr')
128 xlabel('Time (hr)')
129 ylabel('pH')
130 title('pH as a Function of Time')
131 legend('Model','Experiment',1)
132
133 subplot(2,3,3)
134 plot([A]/60,[M],'-g')
135 xlabel('Time (hr)')
136 ylabel('%Rejection')
137 title('%Rejection as a Function of Time')
138 legend('Model',1)
139 %text(2,3.3638,'R2=    MSE=')
140
141 subplot(2,3,4)
142 plot([N],[M],'-g')
143 xlabel('Jw (L/dm2/min)')
144 ylabel('%Rejection')
145 title('%Rejection as a Function of Jw')
146 legend('Model',1)
147 %text(2,3.3638,'R2=    MSE=')
```

```
148
149 subplot(2,3,5)
150 plot([A]/60,[K5],'-g')
151 xlabel('Time (hr)')
152 ylabel('Cfa/Cf0')
153 title('Cfa/Cf0 as a Function of Time')
154 legend('Model',1)
155 %text(2,3.3638,'R2=    MSE=')
156
157 subplot(2,3,6)
158 plot([N],[K5],'-g')
159 xlabel('Jw (L/dm2/min)')
160 ylabel('Cfa/Cf0')
161 title('Cfa/Cf0 as a Function of Jw')
162 legend('Model',1)
163 %text(2,3.3638,'R2=    MSE=')
164
165 fprintf('\n      \n')
166 fprintf('\n      \n')
167 fprintf('\n CONCLUSION     \n')
168 fprintf('\n      R2_W           MSE_W                R2_pH          MSE_pH\n');
169 fprintf('\n%11.3f%20.4e%19.4f%19.4e%\n',R2_W,MSE_W,R2_pH,MSE_pH)
170 fprintf('\n      \n')
171 fprintf('\n      \n')
172 fprintf('\n      Time(min)      J_w(L/dm2/min)          pH            Cf,a(mole/l) ↙
W(kg)          C_f_a1\n');
173 for x=5:5:200000000;
174 z=x*150;
175 fprintf('\n%11.3f%20.4e%19.4f%19.4e%19.4e%\n',T(z),j(z),p(z),c(z),d(z),e(z));
176
177 end
178
179
180
```

```matlab
1
2 function fu=Acetic5_Valeric10(z,t)
3 %Independent Variables
4 V_f_0=1;C_d_s_0=1;V_d_0=0.5;
5 %Fixed Variable
6 T=301;f=1.58;
7 L=0.9207;D=0.023;W=0.4572;A_m=42*10^(-2);d_h=0.0438;
8 % Acid Variables
9 %
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
10 C_f_a_0=5*10^(-3);
11 C_f_a1_0=10*10^(-3);
12 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
13 B_a=2.093/60/100;% B of Acetic Acid
14 B_a1=0.491/60/100;% B of Valeric Acid
15 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
16 D_A_298=6.534*10^(-6);D_HA_298=7.74*10^(-6);%Acetic Physicocmical Property
17 D_A1_298=5.226*10^(-6);D_HA1_298=4.902*10^(-6);%Valeric Physicocmical Property
18 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
19 K_a_T=10^-(1170.48/T-3.1649+0.013399*T);%Acetic acid
20 K_a1_T=10^-(921.38/T-1.8574+0.012105*T);%Valeric Acid
21 %
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
^^^^
22 %Constant Variable
23 B_s=0.256/60/100;% normal..B_s(0.256) =4.27*10^(-5)
24 A=0.442/60/100;%..(0.413) New Value
25 o_s=0.936;
26 S=500*10^-5;
27 D_H_298=5.587*10^(-5);
28 visco_298=((298-273)+246)/((0.05594*(298-273)+5.2842)*(298-273)+137.37);
29 visco_T=((T-273)+246)/((0.05594*(T-273)+5.2842)*(T-273)+137.37);
30 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
31 o_a=1;n_s=2;R=0.08314;
32 o_a1=1;
33 %Temperature Correction
34 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
35 K_CO2_T=10^-(3404.71/T-14.8435+0.032786*T);
36 K_w_T=10^(-4470.99/T+6.0875-0.017060*T);
37 E=2727.586+0.6224107*T-466.9151*log(T)-52000.87/T;
38 d=1-(((T-273)-3.9863)^2*((T-273)+288.9414))/(508929.2*((T-273)+68.12963))...
39 +0.011445*exp((-374.3)/(T-273));
40 A_I=(1.82483*10^6*d^0.5)/(E*T)^1.5;
41 K=1/K_a_T;
42 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
43 K1=1/K_a1_T;
44 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
45 D_H_T=D_H_298*T/298*visco_298/visco_T ;
46 D_A_T=D_A_298*T/298*visco_298/visco_T;
47 D_HA_T=D_HA_298*T/298*visco_298/visco_T;
48 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
49 D_A1_T=D_A1_298*T/298*visco_298/visco_T;
50 D_HA1_T=D_HA1_298*T/298*visco_298/visco_T;
51 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
52 D_1_T=2/(1/D_H_T +1/D_A_T);
53 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
```

```
 54 D_11_T=2/(1/D_H_T +1/D_A1_T);
 55 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
 56 K_H=0.034*exp(-2400*(1/T-1/298));%************************
 57 %For carbonic acid
 58 P_M =0.0025;
 59 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 60 %Unknown Variable
 61 J_w=z(1);P_i=z(2);P_m=z(3);J_s=z(4);C_i_s=z(5);
 62 C_mf_s=z(6);J_a=z(7);C_mf_a=z(8);C_i_a=z(9);C_d_a=z(10);
 63 C_d_H=z(11);gamma_i_H=z(12);I_i=z(13);C_d_s=z(14);
 64 C_mf_H=z(15);gamma_mf_H=z(16);I_mf=z(17);C_md_s=z(18);
 65 D_d_s=z(19);k_d=z(20);Sh_d=z(21);Sc_d=z(22);Re_d=z(23);
 66 v_d=z(24);den_d=z(25);mul_d=z(26);f_2=z(27);D_f_a=z(28);
 67 k_f_a=z(29);C_f_a=z(30);Sh_f_a=z(31);Re_f=z(32);v_f=z(33);
 68 f_1=z(34);Sc_f_a=z(35);D_f_s=z(36);k_f_s=z(37);C_f_s=z(38);
 69 C_f_ai=z(39);C_f_si=z(40);C_d_ai=z(41);C_d_si=z(42);pH_f=z(43);
 70 gamma_f_H=z(44);C_f_A=z(45);I_f=z(46);mul_f=z(47);Sh_f_s=z(48);
 71 Sc_f_s=z(49);w=z(50);C_f_H=z(51);C_f_co2=z(52);C_f_co2i=z(53);
 72 C_f_HCO3c=z(54);J_co2=z(55);C_d_co2=z(56);C_d_HCO3=z(57);
 73 I_d=z(58);den_f=z(59);C_d_A=z(60);
 74 %For 2 acid
 75 C_mf_a1=z(61);C_i_a1=z(62);J_a1=z(63);C_d_a1=z(64);C_d_A1=z(65);C_mf_A1=z(66);
 76 C_f_a1=z(67);D_f_a1=z(68);k_f_a1=z(69);Sh_f_a1=z(70);Sc_f_a1=z(71);
 77 C_f_ai1=z(72);C_d_ai1=z(73);C_f_A1=z(74);C_mf_A=z(75);gamma_d_H=z(76);C_i_H=z(77);
 78 C_i_A=z(78);C_i_A1=z(79);C_d_co2a=z(80);
 79 %Math Model
 80 fu(1)=A*(P_i-P_m )-J_w;
 81 fu(2)= B_s *(C_i_s-C_mf_s)-J_s;
 82 fu(3)= B_a*(C_mf_a-C_i_a)-J_a;
 83 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
 84 fu(4)= B_a1*(C_mf_a1-C_i_a1)-J_a1;
 85 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
 86 fu(5)=C_d_a-C_i_a;
 87 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
 88 fu(6)=C_d_a1-C_i_a1;
 89 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
 90 fu(7)=J_a/J_w-C_d_a;
 91 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
 92 fu(8)=J_a1/J_w-C_d_a1;
 93 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
 94 fu(9)= o_s*n_s*C_i_s*R*T+o_a*(C_i_a+C_i_H)*R*T+o_a1*(C_i_a1+C_i_H)*R*T-P_i;
 95 %Charge Balance Equation at interface %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 96 fu(10)=C_i_A+C_i_A1+K_w_T/C_i_H/gamma_i_H/gamma_i_H-C_i_H;
 97 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 98 fu(11)=A_I*(I_i^(1/2)/(1+I_i^(1/2))- 0.3*I_i)+log10(gamma_i_H);%"Davies"
 99 fu(12)=0.5*(2*C_i_s+C_i_H+C_i_A+C_i_A1+K_w_T/C_i_H/gamma_i_H/gamma_i_H)- I_i;
100 fu(13)=o_s*n_s*C_mf_s*R*T+o_a*(C_mf_a+C_mf_H)*R*T+o_a1*(C_mf_a1+C_mf_H)*R*T-P_m;
101 %Charge Balance Equation at membrane active layer %%%%%%%%%%%%%%%%%%%%%%%%%
102 fu(14)=C_mf_A+C_mf_A1+K_w_T/C_mf_H/gamma_mf_H/gamma_mf_H-C_mf_H;
103 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
104 fu(15)=A_I*(I_mf^(1/2)/(1+I_mf^(1/2))- 0.3*I_mf)+log10(gamma_mf_H);%"Davies"
105 fu(16)=K_a_T*(C_mf_a-C_mf_A)/gamma_mf_H/gamma_mf_H/C_mf_H-C_mf_A;% **additional↵
C_mf_A
106 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
107 fu(17)=K_a1_T*(C_mf_a1-C_mf_A1)/gamma_mf_H/gamma_mf_H/C_mf_H-C_mf_A1;
108 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
```

```
109 fu(18)=0.5*(2*C_mf_s+C_mf_H+K_w_T/C_mf_H/gamma_mf_H/gamma_mf_H+C_mf_A1+C_mf_A)-I_mf;
110
111 fu(19)=((C_md_s-C_i_s)*(12.625*10^(-11)*((0.789*C_i_s+0.211*C_md_s)^4 ...
112 /(J_w*(0.789*C_i_s+0.211*C_md_s)+ J_s)+(0.211*C_i_s+0.789*C_md_s)^4 ...
113 /(J_w*(0.211*C_i_s+0.789*C_md_s)+ J_s ))...
114 -3.885*10^(-9)*((0.789*C_i_s+0.211*C_md_s)^3 ...
115 /(J_w*(0.789*C_i_s+0.211*C_md_s)+ J_s)+(0.211*C_i_s+0.789*C_md_s)^3 ...
116 /(J_w*(0.211*C_i_s+0.789*C_md_s)+ J_s ))...
117 +6.975*10^(-8)*((0.789*C_i_s+0.211*C_md_s)^2 ...
118 /(J_w*(0.789*C_i_s+0.211*C_md_s)+ J_s)+(0.211*C_i_s+0.789*C_md_s)^2 ...
119 /(J_w*(0.211*C_i_s+0.789*C_md_s)+ J_s ))...
120 -4.03*10^(-7)*((0.789*C_i_s+0.211*C_md_s)...
121 /(J_w*(0.789*C_i_s+0.211*C_md_s)+ J_s)+(0.211*C_i_s+0.789*C_md_s)...
122 /(J_w*(0.211*C_i_s+0.789*C_md_s)+ J_s))...
123 +0.53*10^(-5)*(1/(J_w*(0.789*C_i_s+0.211*C_md_s)+J_s)...
124 +1/(J_w*(0.211*C_i_s+0.789*C_md_s)+ J_s))))-S;
125
126
127 fu(20)=((C_d_s-C_md_s)*(12.625*10^(-11)*((0.789*C_md_s+0.211*C_d_s)^4 ...
128 /(J_w*(0.789*C_md_s+0.211*C_d_s)+ J_s)+(0.211*C_md_s+0.789*C_d_s)^4 ...
129 /(J_w*(0.211*C_md_s+0.789*C_d_s)+ J_s ))...
130 -3.885*10^(-9)*((0.789*C_md_s+0.211*C_d_s)^3 ...
131 /(J_w*(0.789*C_md_s+0.211*C_d_s)+ J_s)+(0.211*C_md_s+0.789*C_d_s)^3 ...
132 /(J_w*(0.211*C_md_s+0.789*C_d_s)+ J_s ))...
133 +6.975*10^(-8)*((0.789*C_md_s+0.211*C_d_s)^2 ...
134 /(J_w*(0.789*C_md_s+0.211*C_d_s)+ J_s)+(0.211*C_md_s+0.789*C_d_s)^2 ...
135 /(J_w*(0.211*C_md_s+0.789*C_d_s)+ J_s ))...
136 -4.03*10^(-7)*((0.789*C_md_s+0.211*C_d_s)...
137 /(J_w*(0.789*C_md_s+0.211*C_d_s)+ J_s)+(0.211*C_md_s+0.789*C_d_s)...
138 /(J_w*(0.211*C_md_s+0.789*C_d_s)+ J_s))...
139 +0.53*10^(-5)*(1/(J_w*(0.789*C_md_s+0.211*C_d_s)+J_s)...
140 +1/(J_w*(0.211*C_md_s+0.789*C_d_s)+ J_s))))-D_d_s/k_d;
141
142 fu(21)=Sh_d*D_d_s/d_h-k_d;
143 fu(22)=5.05*10^(-11)*C_d_s^4-2.59*10^(-9)*C_d_s^3+4.65*10^(-8)*C_d_s^2-4.03*10^(-7)↵
*C_d_s+1.06*10^(-5)-D_d_s;
144 fu(23)=L*v_d*den_d/mul_d-Re_d;
145 fu(24)=(f_2+f)/2/W/D-v_d;
146 fu(25)=f+J_w*A_m-f_2;
147 fu(26)=0.0538*C_d_s^2+0.4159*C_d_s+5.0095-mul_d;
148 fu(27)=37.0166*C_d_s+997.2911-den_d;
149 fu(28)= 0.04*Re_d^(3/4)*Sc_d^(1/3)-Sh_d;
150 fu(29)= mul_d/den_d/D_d_s-Sc_d;
151 fu(30)=(0.5*(C_f_a-C_mf_a)*((D_1_T-D_HA_T)/...
152     (sqrt(1+4*K*(0.789*C_mf_a+0.211*C_f_a))*(J_w*(0.789*C_mf_a+0.211*C_f_a)-J_a))...
153     +(D_1_T-D_HA_T)/(sqrt(1+4*K*(0.211*C_mf_a+0.789*C_f_a))*(J_w*(0.211*C_mf_a...
154     +0.789*C_f_a)-J_a))+ D_HA_T/(J_w*(0.789*C_mf_a+0.211*C_f_a)-J_a)...
155     +D_HA_T/((J_w*(0.211*C_mf_a+0.789*C_f_a)-J_a))))+D_f_a/k_f_a;
156 %↵
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
157 fu(31)=(0.5*(C_f_a1-C_mf_a1)*((D_11_T-D_HA1_T)/...
158     (sqrt(1+4*K1*(0.789*C_mf_a1+0.211*C_f_a1))*(J_w*(0.789*C_mf_a1+0.211*C_f_a1)-↵
J_a1))...
159     +(D_11_T-D_HA1_T)/(sqrt(1+4*K1*(0.211*C_mf_a1+0.789*C_f_a1))*(J_w*(0.211↵
*C_mf_a1...
160     +0.789*C_f_a1)-J_a1))+ D_HA1_T/(J_w*(0.789*C_mf_a1+0.211*C_f_a1)-J_a1)...
```

```
161     +D_HA1_T/((J_w*(0.211*C_mf_a1+0.789*C_f_a1)-J_a1))))+D_f_a1/k_f_a1;
162 %↙
%^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
163 fu(32)=Sh_f_a*D_f_a/d_h-k_f_a;
164 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
165 fu(33)=Sh_f_a1*D_f_a1/d_h-k_f_a1;
166 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
167 fu(34)=(D_1_T/(2*K*C_f_a)*(-1+sqrt(1+4*K*C_f_a))+D_HA_T/...
168     (4*K*C_f_a)*(-1+sqrt(1+4*K*C_f_a))^2)-D_f_a;
169 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
170 fu(35)=(D_11_T/(2*K1*C_f_a1)*(-1+sqrt(1+4*K1*C_f_a1))+D_HA1_T/...
171     (4*K1*C_f_a1)*(-1+sqrt(1+4*K1*C_f_a1))^2)-D_f_a1;
172 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
173 fu(36)=L*v_f*den_f/mul_f -Re_f;
174 fu(37)=(f+f_1)/W/D/2-v_f;
175 fu(38)=f-J_w*A_m-f_1;
176 fu(39)= 0.04*Re_f^(3/4)*Sc_f_a^(1/3)-Sh_f_a;
177 fu(40)= mul_f/den_f/D_f_a-Sc_f_a;
178 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
179 fu(41)= 0.04*Re_f^(3/4)*Sc_f_a1^(1/3)-Sh_f_a1;
180 fu(42)= mul_f/den_f/D_f_a1-Sc_f_a1;
181 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
182
183 fu(43)=((C_f_s-C_mf_s)*(12.625*10^(-11)*((0.789*C_mf_s+0.211*C_f_s)^4 ...
184 /(J_w*(0.789*C_mf_s+0.211*C_f_s)+ J_s)+(0.211*C_mf_s+0.789*C_f_s)^4 ...
185 /(J_w*(0.211*C_mf_s+0.789*C_f_s)+ J_s ))...
186 -3.885*10^(-9)*((0.789*C_mf_s+0.211*C_f_s)^3 ...
187 /(J_w*(0.789*C_mf_s+0.211*C_f_s)+ J_s)+(0.211*C_mf_s+0.789*C_f_s)^3 ...
188 /(J_w*(0.211*C_mf_s+0.789*C_f_s)+ J_s ))...
189 +6.975*10^(-8)*((0.789*C_mf_s+0.211*C_f_s)^2 ...
190 /(J_w*(0.789*C_mf_s+0.211*C_f_s)+ J_s)+(0.211*C_mf_s+0.789*C_f_s)^2 ...
191 /(J_w*(0.211*C_mf_s+0.789*C_f_s)+ J_s ))...
192 -4.03*10^(-7)*((0.789*C_mf_s+0.211*C_f_s)...
193 /(J_w*(0.789*C_mf_s+0.211*C_f_s)+ J_s)+(0.211*C_mf_s+0.789*C_f_s)...
194 /(J_w*(0.211*C_mf_s+0.789*C_f_s)+ J_s))...
195 +0.53*10^(-5)*(1/(J_w*(0.789*C_mf_s+0.211*C_f_s)+J_s)...
196 +1/(J_w*(0.211*C_mf_s+0.789*C_f_s)+ J_s))))+D_f_s/k_f_s;
197
198 fu(44)=Sh_f_s*D_f_s/d_h-k_f_s;
199 fu(45)=5.05*10^(-11)*C_f_s^4-2.59*10^(-9)*C_f_s^3+4.65*10^(-8)*C_f_s^2-4.03*10^(-7)↙
*C_f_s+1.06*10^(-5)-D_f_s;
200 %Mole balance Equation
201 fu(46)=(V_f_0*C_f_a_0+C_f_ai*f_1*t)/(V_f_0+f_1*t)-C_f_a;
202 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
203 fu(47)=(V_f_0*C_f_a1_0+C_f_ai1*f_1*t)/(V_f_0+f_1*t)-C_f_a1;
204 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
205 fu(48)=C_f_s*f_1*t/(V_f_0+f_1*t)-C_f_si;
206 fu(49)=(C_f_a*f-J_a*A_m)/f_1-C_f_ai;
207 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
208 fu(50)=(C_f_a1*f-J_a1*A_m)/f_1-C_f_ai1;
209 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
210 fu(51)=(C_f_si*f+J_s*A_m)/f_1-C_f_s;
211 fu(52)=(C_d_s_0*V_d_0+C_d_si*f_2*t)/(V_d_0+f_2*t)-C_d_s;
212 fu(53)=(C_d_ai*f*t+C_d_ai*(V_d_0+f_2*t-f*t))/(f_2*t)-C_d_a;
213 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
214 fu(54)=(C_d_ai1*f*t+C_d_ai1*(V_d_0+f_2*t-f*t))/(f_2*t)-C_d_a1;
```

```
215 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
216 fu(55)=(C_d_s*f-J_s*A_m)/f_2-C_d_si;
217 fu(56)=(C_d_a*f_2-J_a*A_m)/f-C_d_ai;
218 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
219 fu(57)=(C_d_a1*f_2-J_a1*A_m)/f-C_d_ai1;
220 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
221 %additional Equation
222 fu(58)= 0.04*Re_f^(3/4)*Sc_f_s^(1/3)-Sh_f_s;
223 fu(59)= mul_f/den_f/D_f_s-Sc_f_s;
224 fu(60)=J_w*A_m*t*den_d/1000-w;
225 %pH Equation..
226 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
227 fu(61)= pH_f+log10(gamma_f_H*C_f_H);
228 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
229 fu(62)=K_a_T*(C_f_a-C_f_A)/gamma_f_H/gamma_f_H/C_f_H-C_f_A;
230 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
231 fu(63)=K_a1_T*(C_f_a1-C_f_A1)/gamma_f_H/gamma_f_H/C_f_H-C_f_A1;
232 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
233 fu(64)=A_I*(I_f^(1/2)/(1+I_f^(1/2))-0.3*I_f)+log10(gamma_f_H);%Davies;
234 %ionic streanght in feed solution
235 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%5
236 fu(65)=0.5*(2*C_f_si+C_f_H+K_w_T/C_f_H/gamma_f_H/gamma_f_H+C_f_A+C_f_A1+C_f_HCO3c)-↵
I_f;
237 ↵
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
238 %Charge Balance Equation %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
239 fu(66)=(C_f_HCO3c+C_f_A+C_f_A1+K_w_T/C_f_H/gamma_f_H/gamma_f_H)-C_f_H;
240 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
241 %Co2 diffuse to feed solution
242 fu(67)=P_M*C_d_co2-J_co2;
243 fu(68)=C_f_co2i*(V_f_0+f_1*t-f*t)-(C_f_co2*f_1*t-C_f_co2i*f*t);
244 fu(69)=C_f_co2i*f*t+J_co2*A_m*t-C_f_co2*f_1*t;
245 %Carbonic acid produce in feed solutions
246 fu(70)=10^-3.45/gamma_f_H/gamma_f_H/C_f_H*(C_f_co2-C_f_HCO3c)-C_f_HCO3c;
247 % In draw solution
248 fu(71)=gamma_d_H*gamma_d_H*C_d_H*C_d_HCO3/K_CO2_T+C_d_HCO3-C_d_co2;%******
249 fu(72)=(-K_CO2_T+sqrt(K_CO2_T^2+4*K_CO2_T*gamma_d_H*gamma_d_H*C_d_co2a))/↵
(2*gamma_d_H*gamma_d_H)-C_d_HCO3;
250 %ionic streanght in draw
251 %↵
solution%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%↵
%
252 fu(73)=0.5*(2*C_d_si+C_d_H+K_w_T/C_d_H/gamma_d_H/gamma_d_H+C_d_A+C_d_A1+C_d_HCO3)-↵
I_d;
253 ↵
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%↵
%%
254 %Charge balance at draw solution%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
255 fu(74)=(C_d_HCO3+C_d_A+C_d_A1+K_w_T/C_d_H/gamma_d_H/gamma_d_H)-C_d_H;
256 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
257 fu(75)=K_a_T*(C_d_a-C_d_A)/gamma_d_H/gamma_d_H/C_d_H-C_d_A;
258 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%5
259 fu(76)=K_a1_T*(C_d_a1-C_d_A1)/gamma_d_H/gamma_d_H/C_d_H-C_d_A1;
260 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%5
261 fu(77)=10^-(A_I*(I_d^(1/2)/(1+I_d^(1/2))-0.3*I_d))-gamma_d_H;%Davies;
262 %Additional Equation From Interface
```

```
263 fu(78)=K_a_T*(C_i_a-C_i_A)/gamma_i_H/gamma_i_H/C_i_H-C_i_A;
264 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
265 fu(79)=K_a1_T*(C_i_a1-C_i_A1)/gamma_i_H/gamma_i_H/C_i_H-C_i_A1;
266 %^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
267 fu(80)=10^-3.408*K_H/10^(((33.5-0.109*(T-273)+0.0014*(T-273)^2)*I_d-(1.5+0.015*(T-↙
273)+0.004*(T-273)^2)*I_d^2))...
268     /T)-C_d_co2a;
269
270
271
```

Appendix H Experimental Data

H1. A single carboxylic acid as feed solution and NaCl as draw solution

	Acetic acid			Butyric acid	
Time(hr)	Weight Change of Draw solution (Kgs)	pH of feed solution		Weight Change of Draw solution (Kgs)	pH of feed solution
0.0	0.000	3.427		0.000	3.448
0.5	0.017	3.384		0.016	3.432
1.0	0.031	3.360		0.032	3.427
1.5	0.045	3.357		0.046	3.422
2.0	0.060	3.371		0.063	3.411
2.5	0.077	3.351		0.079	3.405
3.0	0.090	3.361		0.094	3.401
3.5	0.099	3.357		0.107	3.394
4.0	0.113	3.314		0.121	3.389
4.5	0.130	3.327		0.135	3.383
5.0	0.145	3.300		0.149	3.376
5.5	0.159	3.285		0.161	3.365
6.0	0.171	3.289		0.173	3.353
6.5	0.182	3.276		0.187	3.357
7.0	0.196	3.266		0.197	3.348
7.5	0.205	3.252		0.210	3.334
8.0	0.221	3.241		0.223	3.336
8.5	0.231	3.229		0.234	3.327
9.0	0.244	3.219		0.245	3.314
9.5	0.255	3.207		0.261	3.308
10.0	0.263	3.198		0.273	3.305
10.5	0.275	3.188		0.285	3.292
11.0	0.289	3.177		0.296	3.288
11.5	0.298	3.168		0.306	3.275
12.0	0.308	3.160		0.317	3.262
12.5	0.323	3.140		0.328	3.254
13.0	0.332	3.126		0.339	3.248
13.5	0.340	3.113		0.350	3.246
14.0	0.354	3.114		0.360	3.223
14.5	0.367	3.091		0.371	3.211
15.0	0.376	3.087		0.380	3.204
15.5	0.388	3.072		0.389	3.199
16.0	0.399	3.064		0.400	3.197
16.5	0.413	3.058		0.409	3.182
17.0	0.422	3.040		0.419	3.174
17.5	0.429	3.036		0.428	3.165
18.0	0.440	3.025		0.437	3.153
18.5	0.451	3.011		0.446	3.145
19.0	0.461	3.005		0.455	3.133
19.5	0.468	2.995		0.462	3.136
20.0	0.477	2.980		0.471	3.128
21.5	0.486	2.968		0.480	3.111
21.0	0.494	2.961		0.488	3.103
21.5	0.503	2.951		0.496	3.092
22.0	0.512	2.942		0.504	3.081
22.5	0.522	2.928		0.512	3.078
23.0	0.530	2.921		0.519	3.074
23.5	0.539	2.911		0.527	3.069
24.0	0.547	2.903		0.535	3.055
24.5	0.556	2.895		0.542	3.046
25.0	0.564	2.886		0.549	3.035
25.5	0.572	2.877		0.556	3.031
26.0	0.580	2.870		0.563	3.026
26.5	0.588	2.861		0.569	3.018
27.0	0.598	2.851		0.575	3.013
27.5	0.604	2.844		0.582	3.005
28.0	0.614	2.836		0.588	2.997
28.5	0.619	2.832		0.594	2.993
29.0	0.628	2.825		0.599	2.989
29.5	0.634	2.818		0.606	2.978
30.0	0.645	2.811		0.610	2.974

Time(hr)	Lactic acid Weight Change of Draw solution (Kgs)	pH of feed solution	Valeric acid Weight Change of Draw solution (Kgs)	pH of feed solution
0.0	0.000	2.926	0.000	3.409
0.5	0.015	2.930	0.027	3.410
1.0	0.032	2.926	0.043	3.409
1.5	0.049	2.931	0.057	3.409
2.0	0.064	2.922	0.071	3.404
2.5	0.078	2.919	0.083	3.390
3.0	0.091	2.913	0.098	3.390
3.5	0.105	2.909	0.110	3.379
4.0	0.118	2.901	0.125	3.378
4.5	0.132	2.899	0.139	3.373
5.0	0.145	2.899	0.149	3.361
5.5	0.162	2.889	0.165	3.356
6.0	0.174	2.888	0.176	3.349
6.5	0.185	2.880	0.188	3.341
7.0	0.197	2.879	0.203	3.339
7.5	0.210	2.877	0.214	3.327
8.0	0.222	2.874	0.228	3.318
8.5	0.235	2.871	0.238	3.315
9.0	0.245	2.866	0.253	3.304
9.5	0.257	2.863	0.263	3.301
10.0	0.269	2.860	0.275	3.293
10.5	0.281	2.855	0.287	3.288
11.0	0.293	2.849	0.297	3.277
11.5	0.304	2.846	0.311	3.264
12.0	0.316	2.840	0.321	3.263
12.5	0.327	2.835	0.330	3.254
13.0	0.338	2.830	0.344	3.246
13.5	0.349	2.824	0.353	3.239
14.0	0.360	2.820	0.362	3.231
14.5	0.371	2.814	0.372	3.223
15.0	0.381	2.808	0.386	3.211
15.5	0.391	2.803	0.395	3.207
16.0	0.401	2.797	0.403	3.198
16.5	0.412	2.792	0.415	3.188
17.0	0.422	2.786	0.425	3.184
17.5	0.432	2.781	0.434	3.175
18.0	0.441	2.775	0.446	3.161
18.5	0.451	2.771	0.456	3.155
19.0	0.461	2.765	0.467	3.148
19.5	0.471	2.760	0.475	3.145
20.0	0.478	2.755	0.479	3.133
21.5	0.489	2.751	0.484	3.124
21.0	0.499	2.747	0.491	3.117
21.5	0.508	2.742	0.500	3.110
22.0	0.517	2.738	0.513	3.096
22.5	0.526	2.733	0.522	3.092
23.0	0.535	2.728	0.531	3.089
23.5	0.543	2.725	0.539	3.084
24.0	0.552	2.721	0.546	3.078
24.5	0.560	2.716	0.553	3.070
25.0	0.567	2.714	0.560	3.063
25.5	0.575	2.709	0.566	3.053
26.0	0.582	2.705	0.570	3.041
26.5	0.589	2.701	0.575	3.034
27.0	0.597	2.700	0.582	3.028
27.5	0.604	2.695	0.590	3.021
28.0	0.611	2.693	0.597	3.015
28.5	0.618	2.690	0.606	3.008
29.0	0.626	2.685	0.614	3.003
29.5	0.632	2.680	0.620	3.002
30.0	0.639	2.676	0.620	2.999

H2. A mixture of two carboxylic acids as feed solution and NaCl as draw solution

Time(hr)	Acetic acid 5 mM+Valeric acid 10 mM		Acetic acid 10 mM+Valeric acid 5 mM	
	Weight Change of Draw solution (Kgs)	pH of feed solution	Weight Change of Draw solution (Kgs)	pH of feed solution
0.0	0.000	3.311	0.000	3.360
0.5	0.015	3.299	0.015	3.369
1.0	0.031	3.289	0.030	3.348
1.5	0.046	3.281	0.045	3.330
2.0	0.063	3.268	0.057	3.320
2.5	0.076	3.261	0.072	3.310
3.0	0.091	3.253	0.085	3.299
3.5	0.104	3.248	0.100	3.287
4.0	0.119	3.238	0.113	3.273
4.5	0.132	3.230	0.127	3.264
5.0	0.146	3.221	0.141	3.256
5.5	0.160	3.213	0.154	3.245
6.0	0.173	3.204	0.168	3.234
6.5	0.186	3.195	0.182	3.224
7.0	0.199	3.188	0.194	3.214
7.5	0.213	3.179	0.208	3.201
8.0	0.225	3.169	0.221	3.191
8.5	0.238	3.159	0.232	3.180
9.0	0.250	3.150	0.245	3.171
9.5	0.262	3.141	0.257	3.159
10.0	0.273	3.133	0.269	3.146
10.5	0.285	3.124	0.281	3.134
11.0	0.296	3.114	0.293	3.123
11.5	0.308	3.106	0.304	3.112
12.0	0.319	3.096	0.316	3.100
12.5	0.330	3.088	0.326	3.089
13.0	0.341	3.077	0.337	3.077
13.5	0.351	3.068	0.349	3.066
14.0	0.362	3.058	0.360	3.056
14.5	0.372	3.050	0.370	3.045
15.0	0.382	3.042	0.381	3.035
15.5	0.393	3.033	0.389	3.026
16.0	0.403	3.025	0.398	3.012
16.5	0.412	3.018	0.410	3.001
17.0	0.422	3.010	0.422	2.991
17.5	0.432	3.003	0.430	2.974
18.0	0.442	2.996	0.439	2.967
18.5	0.452	2.989	0.449	2.953
19.0	0.461	2.982	0.460	2.943
19.5	0.470	2.975	0.469	2.932
20.0	0.479	2.968	0.478	2.922
21.5	0.489	2.962	0.487	2.912
21.0	0.498	2.956	0.497	2.903
21.5	0.507	2.951	0.504	2.892
22.0	0.516	2.945	0.513	2.881
22.5	0.524	2.937	0.520	2.871
23.0	0.534	2.930	0.528	2.864
23.5	0.542	2.923	0.535	2.855
24.0	0.550	2.918	0.542	2.845
24.5	0.559	2.911	0.549	2.835
25.0	0.567	2.906	0.557	2.827
25.5	0.576	2.901	0.564	2.819
26.0	0.584	2.895	0.571	2.811
26.5	0.592	2.891	0.577	2.805
27.0	0.601	2.888	0.584	2.797
27.5	0.609	2.879	0.590	2.788
28.0	0.616	2.875	0.597	2.782
28.5	0.624	2.873	0.603	2.778
29.0	0.629	2.867	0.609	2.772
29.5	0.629	2.862	0.615	2.767
30.0	0.629	2.859	0.621	2.767

Acetic acid 10 mM+Valeric acid 10 mM

Time(hr)	Weight Change of Draw solution (Kgs)	pH of feed solution
0.0	0.000	3.277
0.5	0.017	3.268
1.0	0.033	3.264
1.5	0.049	3.259
2.0	0.064	3.254
2.5	0.078	3.246
3.0	0.093	3.239
3.5	0.107	3.231
4.0	0.120	3.222
4.5	0.134	3.215
5.0	0.147	3.204
5.5	0.160	3.197
6.0	0.173	3.189
6.5	0.185	3.181
7.0	0.197	3.173
7.5	0.209	3.164
8.0	0.221	3.155
8.5	0.232	3.147
9.0	0.244	3.137
9.5	0.260	3.129
10.0	0.271	3.119
10.5	0.282	3.110
11.0	0.292	3.101
11.5	0.303	3.091
12.0	0.314	3.082
12.5	0.323	3.072
13.0	0.334	3.063
13.5	0.344	3.054
14.0	0.354	3.045
14.5	0.364	3.036
15.0	0.373	3.027
15.5	0.382	3.017
16.0	0.391	3.007
16.5	0.400	2.998
17.0	0.409	2.990
17.5	0.417	2.979
18.0	0.426	2.970
18.5	0.434	2.960
19.0	0.443	2.952
19.5	0.450	2.944
20.0	0.458	2.936
21.5	0.466	2.926
21.0	0.479	2.918
21.5	0.486	2.907
22.0	0.493	2.895
22.5	0.500	2.890
23.0	0.506	2.884
23.5	0.513	2.875
24.0	0.519	2.866
24.5	0.526	2.858
25.0	0.532	2.851
25.5	0.538	2.840
26.0	0.544	2.834
26.5	0.549	2.825
27.0	0.555	2.816
27.5	0.560	2.809
28.0	0.565	2.800
28.5	0.570	2.793
29.0	0.575	2.784
29.5	0.580	2.777
30.0	0.585	2.771

H3. A single carboxylic acid as feed solution and NH4Cl as draw solution

	Acetic acid			Butyric acid	
Time(hr)	Weight Change of Draw solution (Kgs)	pH of feed solution		Weight Change of Draw solution (Kgs)	pH of feed solution
0.0	0.000	3.400		0.000	3.443
0.5	0.015	3.395		0.016	3.444
1.0	0.031	3.393		0.033	3.443
1.5	0.047	3.390		0.048	3.445
2.0	0.063	3.385		0.064	3.443
2.5	0.078	3.384		0.079	3.441
3.0	0.092	3.382		0.093	3.438
3.5	0.106	3.378		0.107	3.434
4.0	0.121	3.374		0.121	3.430
4.5	0.134	3.373		0.134	3.427
5.0	0.147	3.369		0.147	3.423
5.5	0.160	3.366		0.160	3.419
6.0	0.173	3.364		0.173	3.415
6.5	0.186	3.360		0.185	3.409
7.0	0.198	3.357		0.198	3.405
7.5	0.210	3.355		0.209	3.400
8.0	0.222	3.352		0.221	3.396
8.5	0.233	3.349		0.232	3.391
9.0	0.245	3.347		0.244	3.388
9.5	0.261	3.344		0.261	3.384
10.0	0.272	3.341		0.272	3.379
10.5	0.283	3.339		0.283	3.376
11.0	0.294	3.337		0.294	3.374
11.5	0.305	3.334		0.304	3.372
12.0	0.315	3.333		0.314	3.369
12.5	0.326	3.329		0.324	3.366
13.0	0.336	3.327		0.335	3.364
13.5	0.346	3.324		0.344	3.361
14.0	0.356	3.322		0.354	3.359
14.5	0.366	3.319		0.363	3.357
15.0	0.375	3.316		0.373	3.354
15.5	0.384	3.314		0.382	3.352
16.0	0.394	3.311		0.391	3.349
16.5	0.403	3.309		0.400	3.347
17.0	0.412	3.306		0.409	3.345
17.5	0.421	3.303		0.418	3.342
18.0	0.430	3.298		0.427	3.340
18.5	0.439	3.294		0.435	3.337
19.0	0.447	3.291		0.444	3.335
19.5	0.456	3.287		0.452	3.333
20.0	0.464	3.284		0.460	3.330
21.5	0.473	3.280		0.468	3.328
21.0	0.485	3.277		0.481	3.325
21.5	0.493	3.274		0.489	3.323
22.0	0.501	3.273		0.497	3.320
22.5	0.508	3.269		0.505	3.317
23.0	0.516	3.267		0.512	3.315
23.5	0.524	3.264		0.519	3.312
24.0	0.531	3.262		0.527	3.310
24.5	0.539	3.260		0.534	3.307
25.0	0.546	3.258		0.541	3.305
25.5	0.553	3.259		0.548	3.302
26.0	0.560	3.257		0.555	3.300
26.5	0.567	3.255		0.561	3.298
27.0	0.574	3.253		0.568	3.297
27.5	0.581	3.252		0.574	3.294
28.0	0.588	3.251		0.581	3.292
28.5	0.595	3.250		0.587	3.290
29.0	0.601	3.249		0.594	3.288
29.5	0.606	3.248		0.600	3.286
30.0	0.611	3.247		0.606	3.284

Time(hr)	Lactic acid		Valeric acid	
	Weight Change of Draw solution (Kgs)	pH of feed solution	Weight Change of Draw solution (Kgs)	pH of feed solution
0.0	0.000	2.941	0.000	3.430
0.5	0.019	2.936	0.006	3.406
1.0	0.036	2.933	0.021	3.417
1.5	0.054	2.929	0.035	3.418
2.0	0.072	2.926	0.049	3.414
2.5	0.090	2.922	0.064	3.400
3.0	0.106	2.918	0.078	3.403
3.5	0.122	2.916	0.092	3.392
4.0	0.139	2.912	0.107	3.394
4.5	0.156	2.908	0.120	3.391
5.0	0.171	2.903	0.134	3.380
5.5	0.184	2.901	0.149	3.378
6.0	0.196	2.897	0.163	3.374
6.5	0.209	2.893	0.176	3.369
7.0	0.220	2.890	0.189	3.370
7.5	0.230	2.886	0.202	3.361
8.0	0.240	2.882	0.215	3.355
8.5	0.252	2.879	0.228	3.355
9.0	0.262	2.875	0.240	3.348
9.5	0.279	2.872	0.253	3.349
10.0	0.288	2.868	0.266	3.344
10.5	0.298	2.864	0.278	3.343
11.0	0.308	2.861	0.290	3.335
11.5	0.317	2.857	0.303	3.325
12.0	0.327	2.854	0.313	3.329
12.5	0.336	2.849	0.326	3.324
13.0	0.344	2.847	0.337	3.320
13.5	0.353	2.842	0.349	3.316
14.0	0.362	2.840	0.360	3.312
14.5	0.370	2.835	0.372	3.309
15.0	0.378	2.831	0.383	3.300
15.5	0.387	2.826	0.393	3.301
16.0	0.394	2.823	0.404	3.296
16.5	0.401	2.819	0.414	3.289
17.0	0.409	2.814	0.427	3.290
17.5	0.417	2.810	0.435	3.285
18.0	0.424	2.807	0.446	3.275
18.5	0.432	2.802	0.457	3.273
19.0	0.441	2.798	0.466	3.271
19.5	0.447	2.793	0.476	3.272
20.0	0.456	2.789	0.486	3.264
21.5	0.464	2.784	0.496	3.258
21.0	0.474	2.781	0.505	3.255
21.5	0.480	2.777	0.515	3.253
22.0	0.487	2.772	0.524	3.242
22.5	0.494	2.767	0.534	3.244
23.0	0.501	2.763	0.542	3.245
23.5	0.507	2.758	0.551	3.245
24.0	0.513	2.753	0.560	3.243
24.5	0.520	2.749	0.569	3.240
25.0	0.526	2.744	0.578	3.237
25.5	0.533	2.739	0.586	3.231
26.0	0.538	2.735	0.594	3.223
26.5	0.544	2.730	0.603	3.221
27.0	0.550	2.725	0.611	3.219
27.5	0.558	2.720	0.620	3.217
28.0	0.566	2.714	0.627	3.215
28.5	0.575	2.709	0.635	3.213
29.0	0.584	2.704	0.643	3.212
29.5	0.592	2.699	0.649	3.210
30.0	0.600	2.693	0.655	3.208

H4. A mixture of two carboxylic acids as feed solution and NH4Cl as draw solution

	Lactic10 mM+Butyric 10 mM		Lactic 10 mM+Butyric 5 mM	
Time(hr)	Weight Change of Draw solution (Kgs)	pH of feed solution	Weight Change of Draw solution (Kgs)	pH of feed solution
0.0	0.000	2.921	0.000	2.991
0.5	0.016	2.914	0.016	2.927
1.0	0.031	2.910	0.031	2.952
1.5	0.043	2.906	0.045	2.942
2.0	0.058	2.900	0.060	2.923
2.5	0.072	2.896	0.074	2.929
3.0	0.086	2.892	0.088	2.941
3.5	0.100	2.889	0.102	2.919
4.0	0.114	2.887	0.115	2.923
4.5	0.127	2.884	0.128	2.920
5.0	0.140	2.880	0.141	2.921
5.5	0.153	2.876	0.153	2.906
6.0	0.165	2.873	0.165	2.912
6.5	0.178	2.870	0.178	2.915
7.0	0.190	2.866	0.190	2.912
7.5	0.201	2.863	0.201	2.892
8.0	0.213	2.860	0.213	2.888
8.5	0.225	2.855	0.224	2.886
9.0	0.236	2.852	0.236	2.892
9.5	0.248	2.850	0.250	2.892
10.0	0.260	2.847	0.263	2.885
10.5	0.271	2.844	0.275	2.877
11.0	0.283	2.840	0.287	2.878
11.5	0.294	2.836	0.297	2.868
12.0	0.305	2.833	0.308	2.867
12.5	0.316	2.829	0.318	2.858
13.0	0.326	2.826	0.328	2.853
13.5	0.337	2.823	0.338	2.854
14.0	0.348	2.819	0.348	2.845
14.5	0.359	2.815	0.357	2.843
15.0	0.369	2.811	0.367	2.840
15.5	0.379	2.808	0.376	2.833
16.0	0.390	2.804	0.386	2.830
16.5	0.400	2.801	0.395	2.823
17.0	0.410	2.798	0.404	2.820
17.5	0.420	2.795	0.412	2.815
18.0	0.430	2.793	0.422	2.809
18.5	0.440	2.789	0.430	2.805
19.0	0.450	2.788	0.439	2.797
19.5	0.460	2.785	0.447	2.792
20.0	0.469	2.782	0.456	2.789
21.5	0.479	2.778	0.462	2.784
21.0	0.488	2.774	0.471	2.781
21.5	0.495	2.771	0.478	2.776
22.0	0.502	2.769	0.485	2.771
22.5	0.509	2.766	0.492	2.764
23.0	0.516	2.763	0.499	2.760
23.5	0.522	2.759	0.505	2.754
24.0	0.529	2.757	0.512	2.749
24.5	0.535	2.754	0.518	2.744
25.0	0.541	2.751	0.524	2.739
25.5	0.549	2.749	0.532	2.733
26.0	0.554	2.744	0.537	2.729
26.5	0.559	2.742	0.542	2.723
27.0	0.566	2.740	0.549	2.718
27.5	0.571	2.737	0.554	2.712
28.0	0.575	2.733	0.558	2.706
28.5	0.579	2.730	0.562	2.700
29.0	0.584	2.728	0.567	2.694
29.5	0.590	2.725	0.573	2.689
30.0	0.594	2.723	0.577	2.683

	Acetic 10 mM+Butyric 10 mM		Acetic 10 mM+Valeric 10 mM	
Time(hr)	Weight Change of Draw solution (Kgs)	pH of feed solution	Weight Change of Draw solution (Kgs)	pH of feed solution
0.0	0.000	3.301	0.000	3.238
0.5	0.012	3.299	0.016	3.230
1.0	0.022	3.296	0.032	3.224
1.5	0.034	3.297	0.047	3.221
2.0	0.046	3.295	0.062	3.217
2.5	0.058	3.293	0.077	3.214
3.0	0.068	3.289	0.090	3.211
3.5	0.080	3.288	0.104	3.208
4.0	0.091	3.282	0.117	3.205
4.5	0.102	3.278	0.131	3.202
5.0	0.113	3.273	0.143	3.200
5.5	0.124	3.268	0.156	3.198
6.0	0.134	3.264	0.167	3.194
6.5	0.145	3.261	0.179	3.191
7.0	0.156	3.257	0.191	3.189
7.5	0.166	3.253	0.202	3.185
8.0	0.176	3.249	0.214	3.183
8.5	0.187	3.245	0.225	3.179
9.0	0.197	3.243	0.236	3.178
9.5	0.208	3.239	0.249	3.175
10.0	0.218	3.236	0.260	3.172
10.5	0.228	3.233	0.271	3.169
11.0	0.237	3.229	0.281	3.167
11.5	0.247	3.226	0.292	3.164
12.0	0.257	3.222	0.303	3.162
12.5	0.267	3.218	0.313	3.159
13.0	0.276	3.215	0.324	3.157
13.5	0.286	3.211	0.334	3.155
14.0	0.295	3.206	0.344	3.152
14.5	0.304	3.202	0.354	3.150
15.0	0.314	3.196	0.363	3.148
15.5	0.323	3.191	0.373	3.145
16.0	0.333	3.186	0.382	3.143
16.5	0.342	3.181	0.391	3.141
17.0	0.351	3.176	0.400	3.139
17.5	0.361	3.171	0.409	3.136
18.0	0.369	3.165	0.418	3.134
18.5	0.378	3.160	0.426	3.132
19.0	0.388	3.155	0.434	3.130
19.5	0.397	3.150	0.443	3.128
20.0	0.405	3.146	0.451	3.126
21.5	0.413	3.142	0.459	3.124
21.0	0.422	3.138	0.473	3.122
21.5	0.431	3.135	0.480	3.120
22.0	0.439	3.133	0.488	3.119
22.5	0.448	3.129	0.496	3.117
23.0	0.457	3.124	0.503	3.115
23.5	0.464	3.123	0.510	3.112
24.0	0.472	3.122	0.517	3.111
24.5	0.480	3.118	0.524	3.109
25.0	0.489	3.114	0.531	3.108
25.5	0.496	3.111	0.538	3.106
26.0	0.504	3.106	0.544	3.105
26.5	0.512	3.106	0.550	3.103
27.0	0.520	3.105	0.557	3.102
27.5	0.527	3.102	0.564	3.101
28.0	0.534	3.101	0.570	3.099
28.5	0.541	3.098	0.576	3.098
29.0	0.548	3.096	0.583	3.097
29.5	0.555	3.094	0.589	3.096
30.0	0.562	3.093	0.598	3.095

VITA

Date of Birth : 18 June 1969

Place of Birth : Bangkok, Thailand

Education:

1990, B.Eng. (Chemical Engineering), Chulalongkorn University

1998, M. Eng (Environmental Engineering), Chulalongkorn University

Employment record:

1996 – Now ;   Managing Director at Pro-Envir Company Limited.

1994 – 1996   Project Engineer at Better Life Company Limited.

1993 – 1994   Design Engineer at Royal Water System Company Limited.

1991 – 1993   Design Engineer at Creation Center Company Limited.