UNIVERSITY
OF TECHNOLOGY
SYDNEY

DESIGN AND OPTIMIZATION OF A FULL SCALE FORWARD OSMOSIS SYSTEM

by Syed Muztuza Ali

Thesis submitted in fulfilment of the requirements for
the degree of

Doctor of Philosophy

under the supervision of Prof. Ho Kyong Shon and Dr.
Sherub Phuntsho

University of Technology Sydney
Faculty of Engineering and Information Technology

August 2020

CERTIFICATE OF ORIGINAL AUTHORSHIP

I, ***Syed Muztuza Ali*** declare that this thesis, is submitted in fulfilment of the requirements for the award of ***Doctor of Philosophy***, in the ***School of Civil and Environmental Engineering/Faculty of Engineering and IT*** at the University of Technology Sydney.

This thesis is wholly my own work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

This document has not been submitted for qualifications at any other academic institution.

This research is supported by the Australian Government Research Training Program.

Production Note:

Signature: Signature removed prior to publication.

(Syed Muztuza Ali)

Date: 10/08/2020

This thesis is dedicated to the memory of my brother

Syed Mortuza Ali

ACKNOWLEDGEMENTS

First of all, I owe my profound gratitude to my supervisor, Prof. Hokyong Shon for providing me with all the resources, scholarly advice, and magnanimous technical, and financial support throughout my PhD study. As a mentor, Prof. Hokyong Shon showed me the quality of being devoted to my work through his contagious passion for research by being a role model. I would also like to convey my appreciation and gratitude to my co-supervisor Dr Sherub Phuntsho for his insightful comments and encouragement. Without their precious support, it would be difficult to achieve this goal. I would like to take this opportunity to thank Dr Leonard Tijing for his comments in my research work.

Besides my supervisors, my sincere appreciation goes to external collaborators Prof. Johannes Vrouwenvelder and Prof. Noreddine Ghaffour who provided me an opportunity to use their laboratories and research facilities at WDRC-KAUST-Saudi Arabia. I would like to thank Dr Adnan Qamar, Dr Youngjin Kim, and Dr Sarah Kerdi for providing me essential support during my research work at KAUST. I would like to acknowledge the contribution of Prof. Am Jang and Dr Sung-Ju Im from SKKU, Republic of Korea for their assistance to conduct pilot scale experiments for this research.

I sincerely acknowledge Dr Md. Johir for his help in the laboratory. A special thanks to my dear friends Dr Nirenkumar Pathak, Van Huy Tran, Mukit Hossain, Nawshad Akther, Ralph Rolly Gonzales, Mingwei Yao, Federico Volpin, Jungeun Kim, Sungil Lim, Myongjun Park, Ugyen Dorji, Pema Dorji, Jiawei Ren, Umakanth and Idris Ibrahim.

I would like to acknowledge full financial support through the Research Training Program (RTP) of the Commonwealth Government of Australia.

Last but not the least, my heartfelt appreciation goes to all my family members, especially my father Syed Mozammel Hossain, mother SZA Motahera Banu, elder brother Syed Muztaba Ali, and my father-in-law SAKM Mokbular Rahman who have been extremely supportive during the entire course of my PhD. I would also extend my gratitude to my wife Syeda Rokhsana Sabrin. Without her cooperation and support, it would not have been possible to accomplish this goal.

Syed Muztuza Ali

Journal Articles Published or Submitted**

1. **Ali, S.M.,** Kim, J. U., Phuntsho, S., Jang, A., Choi, J. Y., Shon, H.K. 2018, ‘Forward osmosis system analysis for optimum design and operating conditions', Water Research, vol. 145, pp. 429-441. **(Chapter 4)**
2. **Ali, S.M.,** Qamar, A., Kerdi, S., Phuntsho, S., Vrouwenvelder, J. S., Ghaffour, N., Shon, H.K., 2019, ‘Energy efficient 3D printed column type feed spacer for membrane filtration', Water Research, vol. 164, pp. 114961.
3. **Ali, S.M.,** Qamar, A., Kerdi, S., Phuntsho, Ghaffour, N., S., Vrouwenvelder, J. S., Shon, H.K., 2020, 'Conceptual design of a dynamic turbospacer for efficient low pressure membrane filtration’, Desalination, vol. 496, pp. 114712.
4. **Ali, S.M.,** Kim, J. U., Phuntsho, S., Jang, A., Choi, J. Y., Shon, H.K. 2018, ‘Forward osmosis system design and optimization using a hollow fibre membrane module for energy efficient desalination', under review in Desalination.
5. Kim, J. U., Phuntsho, S., **Ali, S.M.,** Choi, J. Y., Shon, H.K., 2018, ‘Forward osmosis membrane modular configurations for osmotic dilution of seawater by forward osmosis and reverse osmosis hybrid system’, Water Research, vol. 128, pp. 183-192.
6. Akther, N., **Ali, S.M.,** Phuntsho, S., Shon, H.K. 2020, ‘Surface modification of thin-film composite forward osmosis membranes with polyvinyl alcohol–graphene oxide composite hydrogels for antifouling properties’, Desalination, vol. 491, pp. p. 114591.
7. Qamar, A., Kerdi, S., **Ali, S.M.,** Shon, H.K., Vrouwenvelder, J. S., Ghaffour, N., 2020, ‘Novel hole-pillar spacer design for improved hydrodynamics and

biofouling mitigation in membrane filtration', Under review in Scientific report - Nature.

Conference papers and presentation

1. **Ali, S.M.,** Kim, J. U., Phuntsho, S., Shon, H.K., 'Development of a user friendly software for full scale FO plant design and operation', 8th IWA Membrane Technology Conference & Exhibition on Water and Wastewater Treatment and Reuse (IWA-MTC 2017), 5-9 September, 2017, Singapore.
2. **Ali, S.M.,** Kim, J. U., Phuntsho, S., Shon, H.K., 'Forward Osmosis System Analysis for Optimum Design and Operating Conditions', The 6th IWA Regional Membrane Technology Conference (IWA-RMTC 2018), 10-12, December, 2018, Gujarat, India.
3. **Ali, S.M.,** Kim, J. U., Phuntsho, S., Shon, H.K., 'Development of an optimization algorithm for full scale forward osmosis plant design', Membrane Science & Technology (MST2019) Conference, 13-14 June 2019, Singapore.
4. **Ali, S.M.,** Im, S.J., Phuntsho, S., Jang, A., Choi, J.Y., Shon, H.K., 'Numerical Optimization of Full Scale Forward Osmosis System Using Hollow Fibre Membrane Module', 4th International Conference on Desalination using Membrane Technology, 1-4 December, 2019, Perth, Australia.
5. **Ali, S.M.,** Im, S.J., Phuntsho, S., Jang, A., Choi, J.Y., Shon, H.K., 'Development of system analysis software for optimum design configurations of the forward osmosis process', 3rd International Forward Osmosis Summit, 6 – 8 November, 2019, Korea.

Presentations made during the PhD candidature including proceedings, oral and poster presentations.

LIST OF ABBREVIATIONS

AL-DS	Active layer facing draw solution
AL-FS	Active layer facing feed solution
CTA	Cellulose triacetate
DI	Deionised
DS	Draw solution
ECP	External concentration polarization
FS	Feed solution
FDC	Final draw solution concention
ICP	Internal concentration polarization
LMH	Litres per square meter per hour
MBR	Membrane bioreactor
MF	Microfiltration
NF	Nanofiltration
PAO	Pressure Assisted Osmosis
PRO	Pressure retarded osmosis
RO	Reverse osmosis
RR	Recovery rate
RSF	Reverse salt flux
SWRO	Seawater reverse osmosis
TDS	Total dissolved solids
TFC	Thin-film composite
UF	Ultrafiltration

NOMENCLATURE

Symbol	**Meaning**	**Unit**
A	Water permeability coefficient	$Lm^{-2}hr^{-1}bar^{-1}$
B	Solute permeability coefficient	$Lm^{-2}hr^{-1}$
C	Concentration	M
f	Flexibility	%
J	Flux	LMH ($Lm^{-2}h^{-1}$)
k	Mass transfer coefficient	m/s
K	Diffusivity coefficient	m^2/s
N	Number	---
Q	Volumetric flowrate	m^3/s
R	Ideal gas constant	kJ/kg-K
S	Structural parameter	µm
U	Velocity	m/s
π	Osmotic pressure	bar
ε	Voidage	----
l	Discrete length	m
w	Discrete width	m
RF	Recovery fraction	---
CF	Concentration fraction	---
OPR	Overall performance rating	---
Fl	Flexibility	----

Subscripts	meaning	**Superscripts**	meaning
w	Water	m	Membrane surface
S or slt	Solute	b	Bulk
sp	Spacer	out	Outlet parameters
D	Draw Solution	in	Inlet parameters
F	Feed Solution	F	Final

Abstract

Forward osmosis (FO) process is a promising water filtration technology due to its low energy consumption and less fouling propensity. Performance of a full scale FO system is however significantly influenced by its operating conditions. Moreover, these operating parameters have both favorable and adverse effects on their performance. Therefore, it is very important to optimize its performance for efficient and economic operations. Although numerous studies optimized the lab scale FO system, theoretical optimization of the full scale FO system using commercially available membrane modules was rarely performed. Therefore, this thesis aims to design and optimize a full-scale FO system using different types of commercial membrane modules.

A comprehensive theoretical framework for mass transport through the membrane was developed by coupling the solution diffusion models with the fluid mass and momentum balance equations. The finite difference method was employed to apply the mass transport models at small discretized areas on the membrane surface to estimate the performance of the FO system using a commercial TFC (thin film composite) spiral wound FO module. Fluid flow paths inside the membrane module were simplified to analyze the module scale performance as the full scale analysis considering actual geometry of the flow channel employing computational fluid dynamic simulation technique is very time consuming and computationally expensive process. Analysis results were compared with the experimental results to validate the models. About 5% deviation of simulation results and the experimental findings show a very good agreement between them. A novel optimization algorithm was then developed to estimate the minimum required draw solution (DS) inlet flowrate and the number of elements in a pressure vessel to attain the

design objectives (i.e., the desired final DS concentration and recovery rate at a specific feed solution (FS) flowrate). A detailed parametric study was also conducted to determine the optimum operating conditions for different objectives. It showed that for a specific design objective, a higher recovery rate can be achieved by increasing the DS flowrate and the number of elements in a pressure vessel. In contrast, a lower final concentration can be obtained by lowering the DS flowrate and increasing the number of elements.

Owing to the higher packing density of hollow fiber membrane modules compared to the flat sheet membrane module, this study aimed to design and optimize a full scale FO plant using a hollow fiber module. Mathematical models were developed to simulate the mass transport through the membrane considering the actual geometry of the hollow fiber membrane. Module scale performances were then computed by employing the mass transport models with the fluid conservation laws. Pilot scale experiments were also conducted employing a commercial CTA (cellulose triacetate) hollow fiber FO module to validate the theoretical models. Less than 10% difference between the simulation and experimental results was observed which validated the reliability of the developed simulation models. These mathematical models were then applied to simulate and design a 1,000 m^3/day FO plant using 0.6 M NaCl as draw solution or DS (~seawater) and 0.02 M NaCl feed solution (~MBR effluent) to produce 0.25 M, 0.2 M and 0.15 M NaCl diluted seawater DS. For the full scale design, a single element parallel module arrangement was found more suitable for this commercial hollow fiber membrane element tested in this study. Moreover, the maximum feed solution (FS) inlet flowrate was 3 L/min per hollow fiber element considering the maximum allowable FS inlet pressure. Finally, the numerical simulations revealed that to achieve 0.25 M, 0.20 M, and 0.15 M final DS concentrations from the system, the optimum number of modules

required were 370, 435, and 555 respectively. For the same final concentrations, the DS inlet flowrates to each module were found to be 0.8 L/min, 0.55 L/min, and 0.32 L/min, whereas the FS inlet flowrates were 2 L/min, 2.5 L/min, and 2.5 L/min respectively.

Considering the simple flow configuration and module design, in this study theoretical models were developed for a plate and frame type FO membrane element. These models were validated with the published experimental results and element scale performance data provided by the commercial element manufacturer. The actual flow configuration and the physical dimensions of a commercially available plate and frame element were considered for the simulation. About 10% difference between the experimental and simulation results was observed and hence showed good reliability of the developed models. An overall performance index (based on recovery rate, final draw solution concentration, and membrane elements per module) was then applied to optimize the full-scale FO plant for osmotic dilution of seawater. The simulation results showed that for a 1,000 m^3/day FO plant to produce 0.2 M diluted seawater as final FO product (using a 0.02 M FS (~MBR effluent), inlet flowrate per module of 20 L/min and at 50% feed recovery rate), a total of 47 modules containing 7 plate and frame membrane elements per module and 5 L/min DS inlet flowrate were the optimum design and operating conditions for this particular capacity plant. In addition, for the same recovery rate (50%) the optimum DS inlet flowrates to the system were found about 3, 4, and 6 times lower than the FS inlet flowrate when the desired final DS concentrations were 0.25 M, 0.2 M, and 0.15 M respectively.

From the previous three studies, it was found that designing a large scale FO system is not a trivial job, rather it is a difficult, time consuming, and tedious task. Therefore, this study is aimed at developing a user friendly FO system analysis software (referred to as

FOSA (forward osmosis system analysis) in this study) that can make the design process simple, economical, and efficient. This study first designed a few algorithms to develop the framework of the software. These algorithms were then coded to design the frontend (graphical user interface) and backend (simulation and optimization) layers of the software. The graphical user interface of FOSA receives the input for the design and optimization process from the user and displays the simulation results. However, for the simulation and optimization, FOSA employed the mathematical models and optimization algorithm developed in our previous studies [chapter 4, 5, and 6]. Finally, this software was used to design a 1,000 m^3/day FO plant to produce 0.2 M diluted draw solution using a commercial TFC 8040 spiral wound module, a CTA hollow fiber module, and a plate and frame module. For the same operating conditions (0.6 M NaCl and 0.02 M NaCl draw and feed solution inlet concentration, feed solution to draw solution inlet flowrate ratio was 4), the spiral wound module showed the best performance among the three modules. The CTA hollow fiber module although required smaller system footprint, required larger membrane area due to its lower water permeability. In contrast, the plate and frame module required less membrane area compared to the hollow fiber module due to its better membrane properties, but the lower packing density of this module results in the largest footprint of the system compared to the other modules. Therefore, the developed FO system design and optimization software FOSA is a useful product that came from this research and can play an important role in the uptake of FO by industry.

This thesis finally concludes with the recommendation to improve the accuracy and extend the scope of the FO system analysis software. Development of empirical models for pressure drop across the membrane module and the effects of fouling on the module performance can help to enhance the accuracy of the current FO system design and

optimization studies. Further, the addition of more draw solutions, feed solutions, and membrane modules to the software can enable it to search for numerous potential applications of the FO process. Other osmotically driven processes (such as Pressure retarded osmosis (PRO), pressure assisted osmosis (PAO)) can also be included in the software to widen its scope. Finally, launching this software as a web application to make it available for diverse groups of industrial and academic users will significantly contribute to commercializing the FO process for various applications.

CONTENTS

List of Figures

List of Tables

CHAPTER 1

INTRODUCTION

1 Introduction

1.1 Research background

Water is one of the most important elements for the existence of human beings on earth (Kumar, Bassi & Singh 2020). But increasing population growth, rising development in the industrial sector, growing urbanization contributed to an almost six folds increase in global freshwater demand in the last century which lead to a severe water shortage crisis (Alcamo et al. 2003). World health organization (WHO) reported in 2008 about 884 million people do not have access to clean freshwater and 2.5 billion people have limited access (Wu et al. 2013). United Nations predicted this number will further increase to about seven billion by the year 2050 (Watkins 2006a). As such, sufficient freshwater supply for the increasing global population is one of the major challenges of the current century. However, only 2.5% of the world's natural water reserve is freshwater, remaining (96.5%) is seawater (Ali & Chakraborty 2016). Therefore, Creating new freshwater resources, by desalinating seawater and reusing wastewater are some of the best approaches to a sustainable integrated water management plan to confront this challenge (Li et al. 2018; Luo et al. 2017).

Currently, membrane based reverse osmosis (RO) processes are the most widely used technology for seawater desalination and wastewater reuse owing to their superior product quality and smaller plant size (Peñate & García-Rodríguez 2012; Qasim et al. 2019). However, the energy required to apply hydraulic pressure for the permeation of freshwater through the membrane has limited the energy efficiency of these processes to a great extent. This technology still consumes about 2.5-4 times higher energy as compared to the theoretical minimum energy requirement (1.06 kWh/m^3) for the desalination of seawater (Im et al. 2020; Park et al. 2020). On the contrary, FO is thermodynamically a spontaneous process that is driven by the osmotic gradient of a concentrated (draw solution) and diluted solution (feed solution) separated by a semipermeable membrane. Since no additional hydraulic pressure is needed for the water permeation this process shows great promises as an energy efficient water filtration technology (Kim et al. 2018; Kook et al. 2018). Owing to its high energy efficiency, FO process is engineered and adapted to numerous hybrid processes for clean water production. For example, the integration of FO process with RO desalination system that dilutes the seawater by osmotically using wastewater has already shown a considerable reduction in energy consumption for seawater desalination compared to the standalone RO process (Choi et al. 2017; Im et al. 2020). Therefore, designing an FO process is very important to produce clean fresh water at a smaller energy footprint.

Unlike the laboratory scale or pilot scale FO system, a full scale FO system or FO plant is comprised of a large array of membrane modules or elements arranged in either serial or parallel or a combination of both configurations. Design of an FO plant involves determining the number of membrane modules or elements and their optimal arrangements to obtain the desired performance. Irrespective of any applications, performance of the FO plant is generally expressed by its diluted draw solution production capacity, overall recovery rate, and final draw solution concentration. However, the performance of the FO plant is significantly influenced by its operating conditions. Moreover, these operating conditions have both desirable and undesirable effects on system performance. For example, increasing the DS inlet flowrate increases the FS recovery rate but it also increases the final DS concentration. On the other hand, increasing the number of element/membrane area although increases the recovery rate and decreases the final concentration, it also increases the unit production cost of the system (Lee & Kim 2018; Liyanaarachchi et al. 2016; Phuntsho et al. 2017; Song et al. 2018). Therefore, to design an efficient and economically viable full scale FO system, it is essential to optimize the operating conditions and to evaluate the minimum number of membrane elements required to achieve the desired performance.

However, most of the FO process optimization studies are conducted using a small membrane in a laboratory scale set up. But the performance of a full scale FO system that uses large membrane modules instead of a small membrane token is significantly different from the laboratory scale FO system. Spatial variation of DS and FS concentration over the large membrane area and the effects of external concentration polarization which are almost negligible for the small membrane contribute to the reduction in permeation by the full scale system (Gu et al. 2011b; Song et al. 2018). Therefore, full scale studies are vital to design and optimize the FO system for real applications. Recently, several pilot scale experimental studies have been conducted to optimize the real scale FO plant design. Kim et al. conducted a pilot scale experimental study to optimize the spiral wound membrane module configuration (Kim et al. 2018). Effects of different operating conditions and module type on the performance of a full scale FO process were experimentally investigated in a pilot scale study (Im, Jeong & Jang 2018). In another study, Kook et al. experimentally optimized the pressure-assisted FO modular design by considering the system performance and economic assessment (Kook et al. 2018). Pilot scale experiment was also conducted to investigate the performance of a commercial plate and frame type module at various operating conditions (Song et al. 2018). The same PF elements were used to investigate the performance in terms of water flux and recovery rate as a function of membrane area, concentrations, and flowrates of the solutions (Lee

& Kim 2018). However, most of these pilot scale experimental studies either investigated a single membrane element's performance at several operating conditions or few serially connected element's performances at a very narrow range of operating conditions, because it is not feasible to conduct experiments with many elements and for numerous operating conditions. Therefore, a theoretical study is necessary to design and optimize a full scale FO system for various applications and a broader range of operating conditions.

Several researches have been conducted to theoretically analyze the full scale FO system. However, the FO system analysis studies using various modules are significantly different, due to their dissimilarities in membrane geometry, packing density, and flow configurations. Currently, spiral wound, hollow fiber, and plate and frame type modules are the commercially available modules for FO processes. Having considerably high packing density and commercial availability maximum studies focused on simulating the FO processes using spiral wound membrane module. For example, Xu et al. included the draw solution dilution effects on the permeate flow with the internal concentration polarization (ICP) model in a spiral wound FO module (SWFO) (Xu et al. 2010). The effects of draw solution concentration and flowrate on FO performances were investigated in terms of membrane flux. In another study, the mathematical models for plate and frame module were modified for the unique flow configuration of the spiral wound module. Water flux at different operating conditions such as DS concentration, DS flowrate, FS concentration, and FS flowrate was evaluated by employing the developed models (Gu et al. 2011b). Another theoretical study coupled the solution diffusion models with the differential mass balance equation of the feed and draw channels to evaluate the performance of a spiral wound membrane module (Attarde et al. 2015). Further, a simplified model is developed to quantify the permeate flux of a FO spiral wound module by fitting the experimental performance data at various operating conditions (Jeon et al. 2018).

Research interest has been growing in FO process design using hollow fiber membrane modules also considering its higher packing density and surface area. In 2012, a mathematical model was developed to evaluate the spatial distribution of the FO performance within the fiber and the global performance of a lab-scale hollow fiber module (Xiao et al. 2012). Shibuya et al. presented a modified model by incorporating pressure equation with the external concentration model to investigate the performance of a 5-inch hollow fiber membrane module (Shibuya et al. 2016). FO hollow fiber module performances were also evaluated by modifying the models considering the geometry of the fiber (Lin 2016). Recently, Teklue et al. conducted a simulation at various module lengths of a hollow fiber module and optimized the operating conditions based on permeate flow rate and energy consumption (Teklu, Gautam & Subbiah 2020).

Owing to the relatively lower packing density and less commercial availability, plate and frame type modules are the least studied module so far. However, due to the simpler flow configuration and the lower channel pressure drop, and the growing popularity of commercial (Porifera Inc. USA) plate and frame FO membrane element resulted in some full scale theoretical studies also. Gu et al. explored the effects of various operating conditions such as DS and FS inlet flowrate and concentration on the performance of a single plate and frame element (Gu et al. 2011a). Some of these studies estimated the total membrane area and the number of membrane elements needed for the desired recovery rate (RR) at a given solution flowrate (Deshmukh et al. 2015; Lee & Kim 2018).

Most of these studies considered either co-current or counter-current flow for their analysis but the actual flow configurations of the membrane modules are substantially different (Banchik et al. 2016; Deshmukh et al. 2015; Mondal, Field & Wu 2017). In addition, these large scale studies did not use the actual physical dimensions of a membrane module. These studies optimized the dimensions of the membrane module (such as length or width of the membrane) at different operating conditions, which is very useful to design an optimal membrane module, but did not provide information on the membrane modular arrangement for the design of a full scale FO plant. However, for a full scale plant design using commercially available membrane modules, these parameters cannot be varied because the physical dimensions (length, width, height, and total membrane area) and the flow patterns of these modules are already defined. As such, determining the optimum membrane modular configuration (i.e. number of elements per pressure vessel and the number of pressure vessels) is more important to design an FO plant.

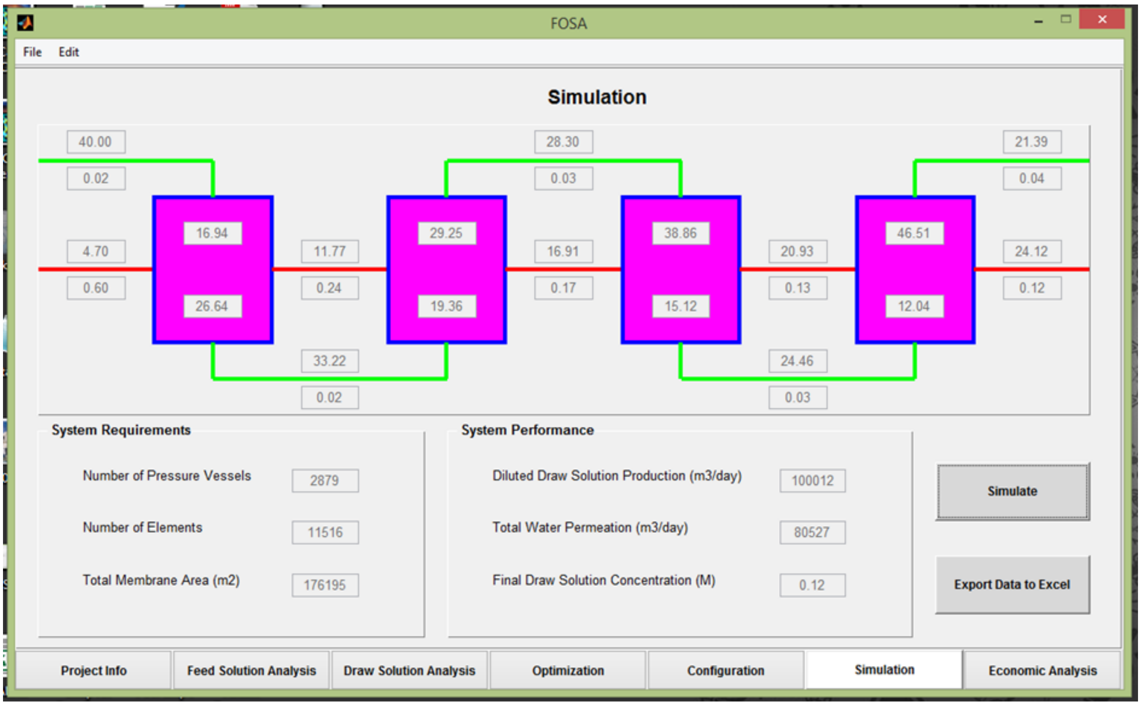
Moreover, these studies maximized the recovery rate of the FO system to evaluate the optimum operating conditions. It is however found that the recovery rate is inversely related to the final concentration of the diluted draw solution produced by the FO system. Consequently, the recovery rate of the system can be maximized but at the expense of reduced dilution factor or increased final DS concentration. In contrast, lower final DS concentration can also be obtained by reducing the recovery rate of the system (Mondal, Field & Wu 2017; Phuntsho et al. 2017; Phuntsho et al. 2016). Therefore, the optimization of an FO system should be based on an overall performance index combining both recovery rate and final DS concentration.

Finally, designing and optimizing a full scale FO system is not a trivial job, rather it is a difficult task. However, a system analysis software can make the design and optimization process efficient by reducing the time to design, estimating the performance before fabrication, and by comparing different membrane modular configurations. Although there are several commercial software available for reverse osmosis (RO) system analysis (such as ROSA, Toray Track, IMSDesign etc), there is no system analysis software for the FO process. So it is very crucial to develop a software for full scale FO system design and optimization.

In order to overcome the limitations of the previous studies, this thesis is aimed at developing an FO system analysis software to design and optimize an FO plant considering the membrane geometry, actual dimensions, and flow configurations of the commercial spiral wound, hollow fiber, and plate, and frame membrane modules. It employed the fluid mass conservation laws and the solution diffusion models to develop mathematical models for the estimation of module scale performances and the information required for the real scale system design. Theoretically investigated performances were then compared with the pilot scale experimental results to validate the models used in this thesis. A novel optimization algorithm based on an overall performance index was developed to optimize the design and operating conditions of the FO system considering both the recovery rate and the final DS concentration. Finally, the graphical user interface of the FO system analysis software were developed that receives the design requirements and the ranges of operating conditions as input. It also employed the models developed for various membrane modules and the optimization algorithm to evaluate the optimum membrane modular configurations and the operating conditions and visually presented the optimum conditions.

1.2 Objectives and scope of the research

This thesis is aimed at developing a software for full scale FO plant design and optimization. It will propose a detailed theoretical framework to numerically simulate the performance of an FO process using various types of membrane modules (spiral wound, hollow fiber, and plate and frame) at different operating conditions. It will also develop a novel optimization algorithm to design an optimum FO plant for any desired performance. Finally, this thesis will develop a FO system analysis software, which will receive design objectives as input and exhibit the optimum design parameters and operating conditions employing the mathematical models for each type of module as shown in Figure 1.1. The software is referred to as "FOSA" in this study.



System Requirements

Number of Pressure Vessels 2879

Number of Elements 11516

Total Membrane Area (m2) 176195

System Performance

Diluted Draw Solution Production (m3/day) 100012

Total Water Permeation (m3/day) 80527

Final Draw Solution Concentration (M) 0.12

Figure 1.1. Graphical user interface of the proposed full scale FO plant design and optimization software, FOSA.

Therefore, the specific aims of the thesis can be summarised as follows:

- Demonstration of the major components of a full scale FO plant identifying its major operating conditions, design parameters, and performance parameters.
- Development of a theoretical framework to analytically estimate the FO plant performance at different operating conditions using a spiral wound, hollow fiber, and plate and frame module.
- Development of an optimization algorithm to determine the optimum operating conditions, number of elements per pressure vessel, and the total number of pressure vessels for certain production capacity, recovery rate, and final DS concentration.
- Development of a system analysis software for the design and optimization of a full scale FO plant.

1.3 Thesis framework

This thesis is organized into total eight chapters where each chapter presents a different aspect of the study. Chapter 1 summarizes the background, objectives, and scope of this study. Detailed background information is provided through a literature review on theoretical fundamentals and main parameters that affect the performance of full scale FO application in Chapter 2. Chapter 3 presents the information on numerical simulation procedures, optimization algorithm development, and pilot scale experimental procedure for the validation of simulation data to describe the methodology of the thesis. Chapter 4 develops mathematical models by combining the fluid mass conservation equation with the solution diffusion model to analyze the performance of a commercial TFC 8040 module. This chapter also reports the optimum draw solution inlet flowrate and the number of elements necessary for various recovery rates and final draw solution concentration. Chapter 5 develops a modified mathematical model for a commercial CTA hollow fiber membrane module by incorporating pressure drop equations with solution diffusion models adopted for the cylindrical geometry of the fiber. This chapter also determines the optimum number of membrane modules, DS inlet flowrate, and FS inlet flowrate for the desired production capacity and final DS concentration. Chapter 6 focuses on designing and optimizing a full scale FO system employing a commercial plate and frame module. It develops the numerical models to investigate the performance of the module at different operating conditions. The novel optimization algorithm developed in this thesis is also used in this chapter for the plate and frame module to determine the information required to design a large scale FO system. Chapter 7 describes the methods of developing the software and its operation procedure in detail. Finally, the findings of the thesis are summarized and the recommendations for future work on this study are provided in Chapter 8.

CHAPTER 2

LITERATURE REVIEW

2 Literature review

2.1 Global water shortage problem

In the current century, the natural reserves of clean freshwater are depleting at an alarming rate as a consequence of an extremely high increase in demand (Stefan 2017). The global usage of water is growing at a rate of more than twice the population growth in the last 10 decades. The global population has reached almost 7.6 billion in 2017 and is projected to rise to 11.2 billion at the end of this century as shown in Figure 2.1(a). In addition to the growing global population, increasing industrialization, high living standards are also contributing to the rise in water demand (Beddington 2011; Suwaileh et al. 2020). Therefore, a severe water scarcity problem is inevitable.

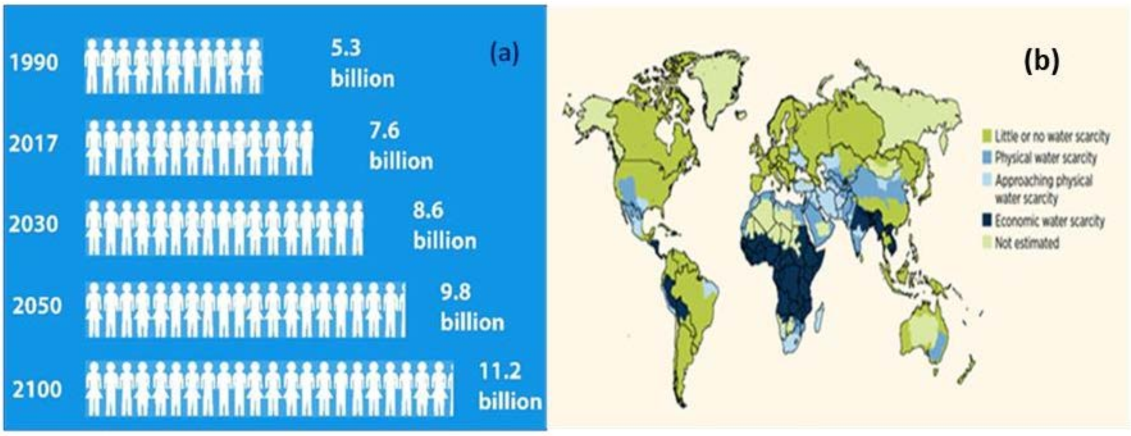


Figure 2.1. (a) World's population growth and (b) water scarcity level globally (2012) (WWAP 2012).

Without any major policy shifts, it is estimated that 2.3 billion more people than today will be living in water stressed regions with a 55% rise in water demand by 2050 (Leflaive et al. 2012) as shown in Figure 2.1(b). Water stressed regions are those where the annual water supply per person falls below 1700 m^3. When the water supply per person drops further to 1000 m^3, the region experiences water scarcity (Molden 2013). However, the global water shortage problem causes serious consequences for public health and sanitation. As a consequence of this problem 1.2 billion people are deprived of potable water and 2.6 million people suffer from proper sanitation. In addition, globally about 3,900 children die every day due to waterborne diseases (Shannon et al., 2008, Stefan, 2017).

The global freshwater reserve is only 2.5% of the total natural water resources, whereas about 96.5% is seawater (Shiklomanov 1993; Trenberth et al. 2007). Therefore, the desalination techniques can be a promising option in minimizing the water shortage problem. However, at present, the available conventional seawater purification technologies are energy-intensive and the produced water is still beyond the affordability of lower income group people (Chekli et al. 2016; Ziolkowska 2015). Reuse of impaired water can be another potential measure to address the water scarcity issue. However, the conventional treatment of wastewater effluent to produce high quality water is also a high energy demanding process (Valladares Linares et al. 2014). Therefore an alternative technology is urgently needed to economically recover freshwater from these unconventional sources for the growing global population.

2.2 FO process and its application in water filtration

In the comparison of commonly deployed reverse osmosis/nanofiltration based membrane technologies FO as an alternative technique demonstrated in the last few decades (Altaee & Hilal 2015; Li 2017). A wide range of potential water filtration applications. The most attractive characteristic of this process is that it is thermodynamically a spontaneous process, which does not require any hydraulic force in operating this process (Wang et al. 2018). Instead of using hydraulic pressure, this process exploits the osmotic pressure difference of two different aqueous solutions separated by a semipermeable membrane to transfer water from the less concentrated feed solution (FS) to the more concentrated draw solution (DS) through the membrane, results in dilution of DS (Awad et al. 2019; Shaffer et al. 2015). The freshwater is produced by subsequent filtration of the diluted DS by using a low pressure RO process at lower energy consumption (Choi et al. 2017; Im et al. 2020). FO processes have few more distinctive advantages, such as a) superior product quality due to the high rejection of the FO membrane resulted from its very small pore size (mean radius 0.25 – 0.30 nm) (Lu & He 2015), b) simple membrane cleaning and high flux recovery as fouling on the dense FO membrane surface is highly reversible (Li et al. 2016). Therefore the FO process shows great promises to confront the global water shortage problem of the current century.

FO processes for water application generally employs an extremely saline draw solute like either ocean water or osmotically less saline but impure water like wastewater feed. This promising feature of the FO process to combine seawater desalination with impaired water reuse encouraged the innovation of numerous water related applications (Ang et al. 2019). For example, FO process was integrated with the MD process (FO-MD) to treat digested sludge concentrate (Xie et al. 2014) This process was also combined with the electrodialysis process (FO-ED) to recover freshwater from secondary effluent (Zhang et al. 2013). OMBR is one of the most widely discussed applications of the FO process which integrates a biological reactor with the FO membrane to generate freshwater from the biologically treated wastewater using a highly concentrated draw solution (Achilli et al. 2009; Yap et al. 2012). Dilution of fertilizer solution for direct fertigation is another widely studied application using the FO process (Phuntsho et al. 2016; Phuntsho et al.

2011). In this application water from a feed solution is permeated to dilute a concentrated fertilizer based draw solution and the diluted draw solution is applied for irrigation. FO technique is also employed in reuse applications for impaired water from textile, oil, and gas industries.

Owing to its wide range of applications and numerous advantages over conventional processes more knowledge on FO membrane material and its mass transport mechanism is required to design and optimize the FO process for its commercial applications.

2.2.1 Mass transport mechanism of the FO process

In FO semipermeable membrane with very high rejection properties exploits osmotic pressure gradients of the solutions to achieve separation of solvent molecules from diluted feed to the concentrated draw side (Anderson & Malone 1974). The transport mechanism for an ideal FO process is explained in Figure 2.2(a). The chemical potential of solvent in the feed solution is higher compared to the draw solution, which creates the osmotic pressure gradient ($\Delta\pi$) between the two solutions. The mass transfer of solvent molecule (J_w) continues until the chemical potentials of the two solutions reach the equilibrium. Most common mass transport equation for an ideal FO process is given by,

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

$$J_s = B\Delta C \tag{2.2}$$

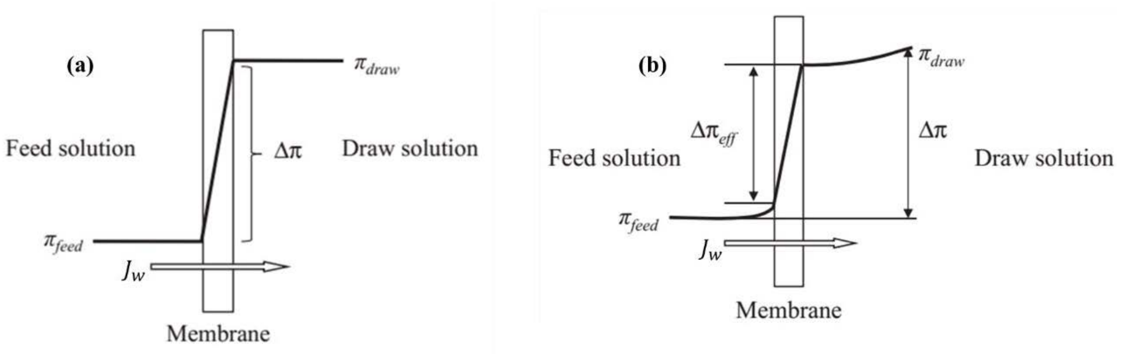


Figure 2.2. Mass transport mechanism for (a) an ideal FO process and (b) an FO process with external concentration polarization (ECP). Adapted from (Shon et al. 2015) .

here, J_W and J_s are the water flux and solute flux through the membrane, A and B are the water permeability coefficient and solute permeability coefficient of the membrane, $\Delta\pi$ is the osmotic pressure difference corresponding to the bulk concentration of feed and draw solution, ΔP is the hydraulic pressure difference across the membrane, and ΔC.is the difference of concentration between FS and DS. Concentration polarization in the membrane system plays an important role in its transport mechanism. External concentration polarization (ECP) changes the osmotic pressure difference in both feed and draw sides of the membrane as shown in Figure 2.2 (b). Due to the velocity boundary layer formed over the membrane in the feed side solution concentration on the membrane surface is higher than the bulk concentration which results in increased osmotic pressure on the feed side. Change in osmotic pressure due to ECP is expressed by,

$$\pi_{F,eff} = \pi_F exp\left(\frac{J_w}{k_F}\right) \tag{2.3}$$

here, $\pi_{F,eff}$ is the actual FS concentration on the membrane due to ECP, π_F stands for the bulk osmotic pressure of FS, J_w represents the water flux through the membrane and k_F is the mass transfer coefficient which is a function of feed channel hydrodynamics.

On the other hand permeation of water from the feed side repels the solute away from the membrane which contributed to a reduction in osmotic pressure in the draw solution side. As a result, the effective osmotic pressure difference $\Delta\pi_{eff}$ is lower than the bulk osmotic pressure difference($\Delta\pi$) which leads to a lower water flux from the system. However, ECP can be minimized by optimizing the operation of the system as it is a function of the operating conditions.

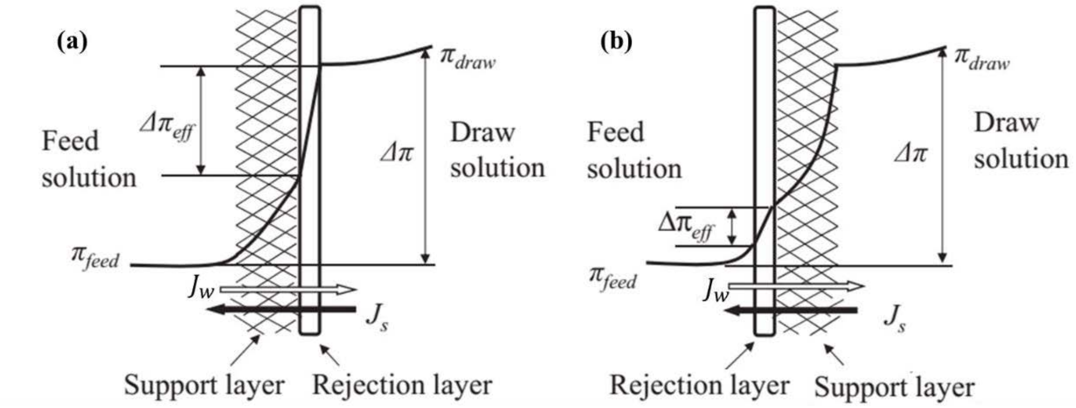


Figure 2.3. Effects of internal concentration polarization on asymmetric FO membrane in (a) AL-DS and (b) AL-FS orientation. Adapted from (Shon et al. 2015)

In addition to ECP, as can be seen in Figure 2.3, another type of concentration polarization occurs with the asymmetric semipermeable membrane structure which is known as internal concentration polarization (ICP). In FO membrane configuration to increase mechanical strength, PA active layer is supported by thick porous support made of PTFE. This layer obstructs the diffusion of solute through it which results in a substantial difference between the bulk osmotic pressure and effective osmotic pressure on the membrane active layer. When the active layer is oriented towards the DS (AL-DS orientation), solute particles accumulate inside the layer and contribute to increasing the osmotic pressure in the feed side. In contrast, for AL-FS orientation (active layer faces

FS) the support layer resists the diffusion back of solute from the bulk phase of DS to membrane which lowers the effective osmotic pressure in the DS side as shown in Figure 2.3(b). Effect of ICP for AL-FS orientation is mathematically expressed by,

$$\pi_{D,eff} = \pi_D exp\left(-\frac{J_w S}{D}\right) \tag{2.4}$$

Where, $\pi_{D,eff}$ and π_D are the effective osmotic pressure and bulk osmotic pressure of DS, D is the diffusivity of the draw solution and S is the structural parameter of the membrane which is a function of tortuosity, porosity, and thickness of the membrane. ICP is very minimally affected by changing the hydraulic conditions in the membrane channel, but it is affected by the membrane support structure.

Apart from ECP and ICP, reverse solute flux (RSF) is another very important aspect of FO mas transport theory. Even though an ideal semipermeable membrane should reject any dissolved solute to transfer from DS to FS, but a small amount of solute particle diffused through the membrane to FS side as shown in Figure 2.3. Considering the effects of ECP and ICP, the common mass transport equation is modified for water flux and reverse salt flux as follows

$$J_w = \frac{A\left(\pi_D \exp\left(-\frac{J_w S}{D}\right) - \pi_F \exp\left(\frac{J_w}{k_F}\right)\right)}{1 + \left(\frac{B}{J_w}\right)\left(\exp\left(\frac{J_w}{k_F}\right) - \exp\left(-\frac{J_w S}{D}\right)\right)} \tag{2.5}$$

$$J_s = \frac{B\left(C_D \exp\left(-\frac{J_w S}{D}\right) - C_F \exp\left(\frac{J_w}{k_F}\right)\right)}{1 + \left(\frac{B}{J_w}\right)\left(\exp\left(\frac{J_w}{k_F}\right) - \exp\left(-\frac{J_w S}{D}\right)\right)} \tag{2.6}$$

The review on mass transport theory indicates that to design an optimum FO system it is essential to optimize the operating conditions and to select the most appropriate type of membrane to minimize the ECP and ICP effects.

2.2.2 FO membrane characteristics

Membrane characteristics play a key role to design an efficient FO system. An ideal FO membrane is characterized by a thin dense layer with high water permeability, and salt rejection as well as a thin and porous support layer with enough mechanical strength. Since both water permeability and solute rejection depend on the dense layer, thickness and the perfection of this layer are some of the most important properties of the membrane. An extremely thin dense layer reduces the transport resistance to enhance the water permeability, whereas a defect-free layer improves the solute rejection or selectivity. However, it is a major challenge to fabricate an ultrathin but defect-free dense layer, which leads to a trade-off between permeability and selectivity of the membrane. Moreover, the selectivity requirement of the membrane changes with the application of the membrane. For water related applications where NaCl is present, a very small pore size (similar to the free volume of the membrane polymer) or almost defect-free membrane is required. Typically, two different types of membranes are used for these applications such as integrally skinned asymmetric membrane and thin film composite (TFC) membranes. Among all available polymeric materials, cellulose triacetate (CTA) was mostly used for the integrally skinned membrane and crosslinked polyamide was used for the TFC membranes. In addition, the structure of the FO membrane support layer is also very important to improve the process performance by minimizing the ICP effects. A thin, porous, and less tortuous membrane is less affected by the ICP effect. In order to achieve this, a thin, porous, and hydrophilic woven/non-woven support is used for the support layer.

Cellulose triacetate (CTA) integrally skinned asymmetric and polyamide thin film composite (TFC) membranes are the most widely available commercial FO membranes in the market at this moment. CTA membranes are fabricated by phase inversion where the dense skin layer and porous sublayer are formed in one step. Hydrophilic properties of the cellulose make this membrane easily wettable. Therefore, high water flux and better fouling resilience are obtained by using this membrane (Luo et al. 2016). Despite these advantages of the CTA membrane, its application is greatly limited by its tendency to hydrolyze and biologically degrade (Hou, Lu & Ren 2016). Moreover, the CTA membranes operate at a narrow pH range of 4 to 7, which is another concern for its application in FO process (Yip et al. 2010).

Another widely used commercial FO membrane is thin film composite (TFC) membrane. This TFC membrane is asymmetric and made of thin polyamide active layer and thick porous support layer. The active layer selectively rejects all salts from feed stream and allows pure water to permeate easily while the support layer provides mechanical strength to the membrane (Akther, Phuntsho, et al. 2019; Yadav et al. 2020). As compared to the CTA FO membrane TFC can offer very high flux and salt rejection. Moreover, it can be operated in a wide pH range typically 2 to 11 in comparison to CTA FO membrane as well as more resistant to biodegradation (Hou, Lu & Ren 2016). This means that the TFC FO membrane can be operated in harsh process conditions for a longer time. The active thin layer of TFC FO membranes is fabricated by the interfacial polymerization method. In this process to monomer molecules such as m-phenylenediamine (MPD), and trimesoyl chloride (TMC) are dissolved in a non-polar organic solvent such as hexane (Akther, Lim, et al. 2019). In the beginning commercial TFC RO membranes were used in FO applications. However, due to different transport mechanisms in the FO process, the thick support layer offered more ICP. So recent research efforts are diverted in the direction to make thinner and highly porous TFC membrane that reduces ICP leads to more efficient FO applications (Hunt 2007).

2.3 FO membrane modules

In the commercial market, four different FO membrane modules are available. The module design variations found are (1) plate and frame module, (2) tubular module, (3) spiral wound module (4) hollow fiber module (Song et al. 2018). The major selection criteria for any membrane module should be packing density. The higher packing density is preferred as it consumes less space means more treated water can be produced using the configuration (Li, Wang & Chung 2004). The plate and frame type module is the oldest version of the module and still widely employed in research laboratories for membrane characterization studies to evaluate the process performance. The tubular and hollow fiber are almost the same module design except for inner dimensions.

2.3.1 Spiral wound membrane module

The spiral wound module is the most widely used membrane module employed in water and wastewater treatment facilities (Attarde, Jain & Gupta 2016). The spiral wound design is as simple as wrapping flat sheet membranes around a porous tube (as shown in Figure 2.4) and this configuration obviously offers higher packing density and less system footprint. This feature makes a spiral wound membrane module one of the most attractive types of modules (Bamaga et al. 2011). However, the downside of this design is that as compared to the RO module spiral wound, FO module offers less packing density. This is due to the provision of cross-flow contacting of two fluid streams (feed side and draw side) in this design warrants attaching spacers between two successive membrane sheets.

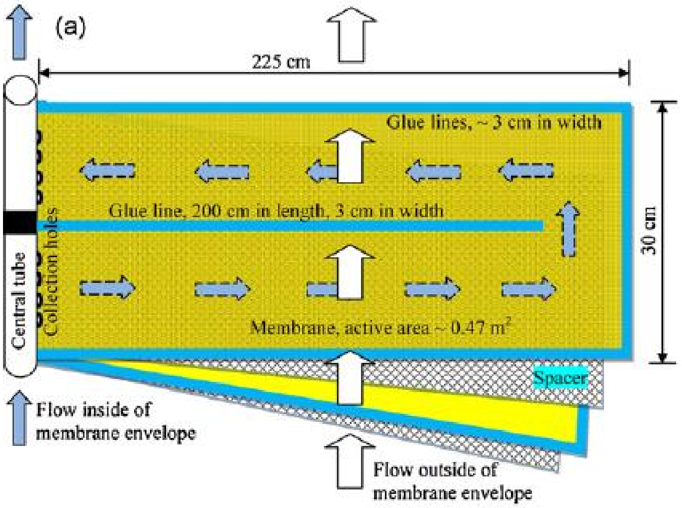


Figure 2.4. Unfurled FO spiral wound module. Adapted from (Xu et al. 2010).

Some advantages and disadvantages of the spiral wound module are given in Table 2.1 (Bamaga et al. 2011; Choi et al. 2017; Gu et al. 2011b).

Table 2.1 Advantages and disadvantages of the spiral wound module.

Packing density	Approximately 600 m^2/m^3.
Advantages	High packing density makes it a more attractive option in large scale applications.
Disadvantages	Fouling is a major challenge for a complicated flow path and large flow dead zones in the channel.

2.3.2 Hollow fiber membrane module

Hollow fiber (HF) membrane module has drawn considerable attention for both FO and PRO processes in both academic communities and industrial researches. The HF module has the following components, 1. HF bundle, 2. Housing, 3. End sheets (Figure 2.5) (Wan et al. 2017). The main component is a bundle in which hundreds of hollow fiber membranes are enclosed within a tube sheet. Once the membrane bundle is installed into the housing it divides it into two parts. Bore side is a space enclosed by membrane fibers and the shell side is a space between the outer space of a membrane and housing (Shibuya et al. 2016). The housing is a cylindrical shaped structure with a uniform diameter with feed or permeate end sheets attached on both sides. In HF module two main configurations are shell-side feed design, Lumen-side feed design. Feed usually passes through either inside or outside the fibers and is collected at the other end of the tube sheets. The shell-side feed design module can be operated at very high pressure but fouling propensity is also very high. In the lumen-side feed design, the module operates at low or moderate pressure characterized by comparatively less fouling propensity and concentration polarisation (Wan et al. 2017). HF module possesses a very high surface to volume ratio, so much compact design and less space are required for given product output than the spiral wound module. Researchers have tried with various hollow fiber modules in an effort to increase the flux and rejection for FO processes albeit full scale production is scarce. Recently ongoing efforts are being made to develop the outer selective hollow fiber FO membrane module with very high flux and less fouling propensity (Lim et al. 2019; Tran et al. 2019).

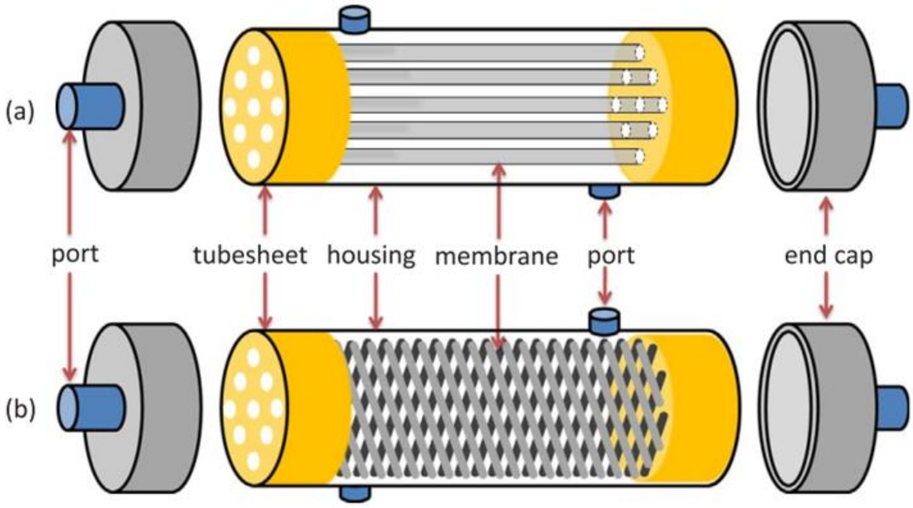


Figure 2.5. Major components of an FO hollow fiber membrane module. Adapted from (Shibuya et al. 2016).

2.3.3 Plate and frame membrane module

Plate and frame type is the simplest configuration to hold flat sheet membranes are available in different sizes and shapes employed in laboratory experiments in coupon form to the commercial scale applications. In this design, CTA or TFC flat sheet membranes are sealed to the frames. Figure 2.6 shows a commercially available plate frame element and two modules. Feed and draw solution flows in between the spacer plates across the semipermeable membrane. Usually, DS is allowed to pass through the small gap between the plate and membranes at very high velocity and the osmosis process is realized. AL-FS (Active layer towards feed side) is a more preferred membrane orientation in this design to avoid internal concentration polarization (Benton & Bakajin 2017; Lee & Kim 2018; Song et al. 2018). Moreover, feed side spacers are not required so that cost saving is possible which makes this module a more attractive alternative in

FO applications. The drawback of this design is the lack of adequate membrane support and low packing density.

Figure 2.6. Commercial plate and frame FO membrane elements and modules. Adapted from (Benton & Bakajin 2017).

Table 2.2 Manufacturers of various FO membrane modules (Awad et al. 2019).

Module type	Manufacturer
Plate and frame	Porifera Inc. USA.
Hollow fiber	Aquaporin, Toyobo
Spiral wound	Toray, FTS H20

Low packing density means more space requirement and higher initial cost as well as operating cost for membrane change over (Belfort 1988). Therefore, this module cannot be used for high volume applications such as sewage treatment or desalination. However, a very specific type of high fouling streams with very high viscosity can be treated. Manufacturers of different types of membrane modules are listed in Table 2.2.

2.4 FO module scale modeling

FO process modeling studies were mostly focused on the evaluation of internal concentration polarization and external concentration polarization for different operating conditions. Outcomes of these studies however are very important for the fundamental understanding of the process and applicable for lab scale studies. But the findings of these studies cannot be translated for the module scale or plant scale FO system performance. These lab scale studies assume that the whole membrane surface is exposed to the same concentration and velocity. However, during the real operation of a membrane module as a result of water flux and RSF, DS is diluted and FS is concentrated along the path of flow, which eventually reduces the solution concentration gradient. Therefore, for a full scale study, the membrane is required to be discretized to sufficiently smaller areas and the models to be applied iteratively to the discrete areas considering the effects of velocities, concentrations, water flux, and RSF of one discrete area on the next area according to the directions of the flow. So far only a few studies have attempted to develop models and simulate the performance of module scale performance of the FO process. Moreover, due to the difference in module structure and flow configurations the mathematical models and simulation techniques significantly vary for each type of module. This section summarizes the full scale modeling and simulation studies using different types of modules with their most important findings.

2.4.1 Models for the Spiral wound module

In one of the early stage module scale simulations, the effect of draw solution concentration and operating conditions on FO performance using a spiral wound module were numerically simulated (Xu et al. 2010). The effects of DS dilution were considered for their model. The spatial distribution of DS concentration in only one dimension was quantified by equation (2.7),

$$C_{d,x} = C_{d,0}\frac{Q_{cf,0}}{Q_{cf,0} + \int_0^x J_v dA_m} \tag{2.7}$$

here, $C_{d,0}$ and $Q_{cf,0}$ are the DS concentration and the crossflow rate at the inlet of the module. It also calculated the permeate flowrate by,

$$Q_p = \int_0^{exit} J_v dA_m \tag{2.8}$$

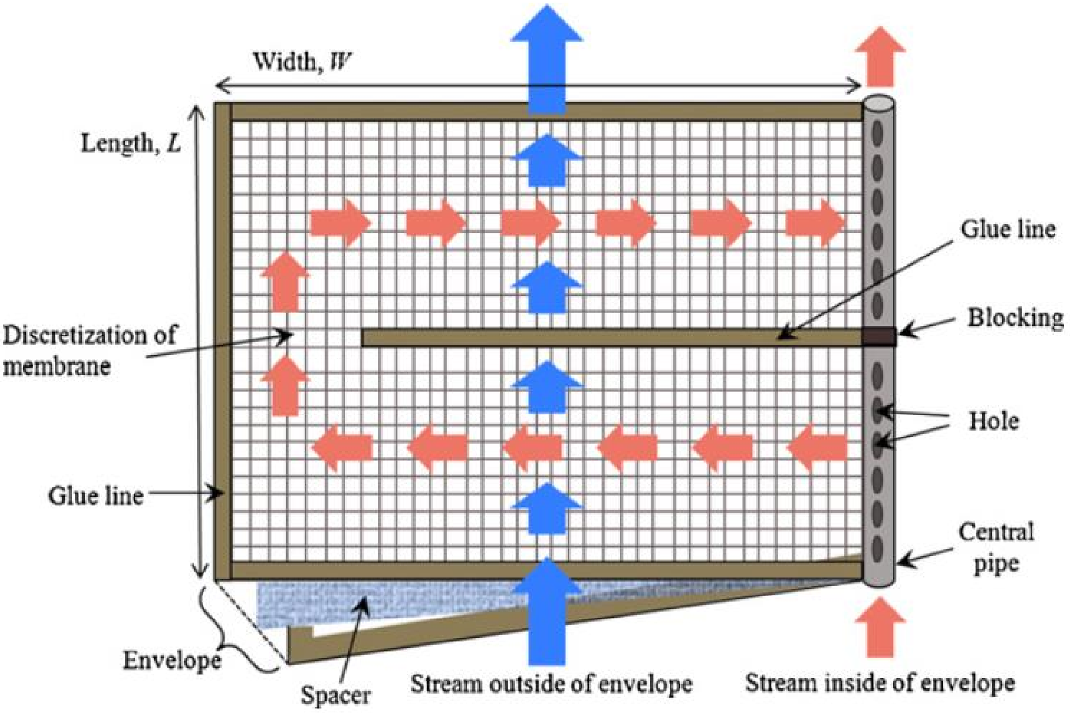


Figure 2.7. Discretization of spiral wound membrane for module scale simulation. Adapted from (Gu et al. 2011b)

This study found the rate of permeation increased with the increase in DS concentration. In addition, it was also observed that at higher DS concentration the effects of ICP are more severe which substantially lowers the water flux through the membrane.

In another study, the mathematical models for u-shaped draw solution flow path and local distribution of concentration and velocity were developed to precisely estimate the spiral wound module performance (Gu et al. 2011b). This study discretised the membrane as shown in Figure 2.7. According to the discretization, it proposed mathematical models for local velocity and bulk concentration in the membrane channel by,

$$U_F(i,j) = U_F(i,j-1) - J_w(i,j)\frac{\Delta x}{0.5H} \tag{2.9}$$

$$C_F(i,j) = \frac{C_F(i,j-1)U_F(i,j-1) - J_s(i,j)(\Delta x/0.5H)}{U_F(i,j)} \tag{2.10}$$

$$U_D(i,j) = U_D(i,j-1) + J_w(i,j)\frac{\Delta x}{0.5H} \tag{2.11}$$

$$C_D(i,j) = \frac{C_D(i,j-1)U_D(i,j-1) + J_s(i,j)(\Delta x/0.5H)}{U_D(i,j)} \tag{2.12}$$

here, U_F and U_D stand for the feed solution and draw solution velocity respectively, whereas C_F and C_D are the feed and draw solution concentration. i and j represent the two dimensional location of the membrane channel, H is the thickness of the channel.

Employing the local concentration and velocity, this study estimated the external concentration polarization and internal concentration polarization effect which eventually calculate the water flux and reverse salt flux. Finally the influences of feed solution and draw solution flowrates and concentrations on the FO system performances were theoretically evaluated.

FO Spiral wound module scale modeling was further modified. This study also coupled the mass balance and solution diffusion equation to determine the local distribution of solution concentration and velocity. However, the Mass transfer coefficient was calculated by considering the effect of spacer filled channel (Attarde et al. 2015). Mass transfer equation for the membrane is expressed as,

$$k = 0.2\left(\frac{d_h v\rho}{\mu}\right)^{0.57}\left(\frac{\mu}{\rho D}\right)^{0.40}\left(\frac{D}{d_h}\right) \qquad (2.13)$$

here, ρ and μ mean the density and viscosity of the solution, v represents the velocity, D is the solute diffusivity in water and d_h represent the hydraulic diameter of the spacer filled membrane channel. The modified model for the hydraulic diameter is given by,

$$d_h = \frac{4\varepsilon}{\frac{2}{H} + (1-\varepsilon)\frac{S_{ws}}{V_s}} \qquad (2.14)$$

here ε, V_s and S_{ws} are the channel porosity, volume, and wetted surface of the spacer respectively. Considering the modified mass transfer coefficient external concentration polarization was determined which provides the water flux and revere salt flux.

FO Spiral wound membrane module's performance was also simulated by using Spiegler-Kedem (SK) model instead of using the conventional solution diffusion model (Attarde, Jain & Gupta 2016). This model is derived from the three parameters based on irreversible thermodynamics. Water flux and solute flux equations for the SK model are given by,

$$J_w = L_p[B\sigma(C_{Dm} - C_{Fm}) - (P_D - P_F)] \qquad (2.15)$$

$$J_s = -\frac{J_w(1-\sigma)\left[C_{Dm} - \left(exp\left[\frac{J_w(1-\sigma)}{B}\right]C_{Fm}\right)\right]}{\exp\left[\frac{J_w(1-\sigma)}{B}\right]} \qquad (2.16)$$

The reflection coefficient σ is the additional parameter compared to the solution diffusion model. L_p and B are the water permeability and solute permeability coefficients.

A simple modeling approach was presented to investigate the performance of a commercial TFC FO 8040 element (Jeon et al. 2018). In this method, a simple ICP and ECP based solution diffusion model was first proposed for the flux permeation through the membrane, which is given by,

$$J_{w,m1} = A(\Delta\pi_{eff} + \Delta P) \qquad (2.17)$$

here A, ΔP and $\Delta\pi_{eff}$ are the water permeability coefficient, transmembrane pressure, and effective osmotic pressure difference. Later on, the error between the experimental and theoretical results was minimized by incorporating hydrodynamic parameters (feed solution flowrate, draw solution flowrate, and transmembrane pressure) in the theoretical model. After modifying the theoretical model the final empirical models were given as,

$$J_{w,m2} = J_{w,m1} + 7.09\Delta P + 0.07Q_f + Q_d \qquad (2.18)$$

here, $J_{w,m2}$ is the predicted water flux value by the empirical model, $J_{w,m1}$is the flux estimated by the theoretical model. In addition, Q_f, Q_d and ΔP are the feed solution inlet flowrate, draw solution inlet flowrate.

2.4.2 Modeling of hollow fiber module

A theoretical framework was developed to simulate the performance of a lab-scale TFC hollow fiber membrane module (Xiao et al. 2012). The one-dimensional model is developed by dividing the spatial area of the module into three subdomains as shown in Figure 2.8. These subdomains are i) the boundary layer on the active side of the membrane, ii) support layer of the membrane, and iii) the boundary layer on the support side of the membrane. Mass transfer coefficient in the three subdomains was defined k_F, k_b and k_m. Incorporating the mass transfer coefficients with solution diffusion equation, the modified equation is,

$$J_w = \left(\frac{1}{k_F} + \frac{1}{k_D} + \frac{1}{k_m}\right)^{-1} ln \frac{A\pi_D + B}{A\pi_F + B + J_w \exp\left(-(J_w/k_F)\right)} \tag{2.19}$$

where, A and B are the water and solute permeability coefficients, π_D and π_F are the bulk osmotic pressure of the solution. Then mass balance equations were coupled with the proposed solution diffusion equation to determine the local distribution of concentration and velocity within the module. The mass balance equations are given by,

$$C_F = C_{F0}\frac{Q_{F0}}{Q_F} + C_{ac}\left(\frac{Q_{F0}}{Q_F} - 1\right) \tag{2.20}$$

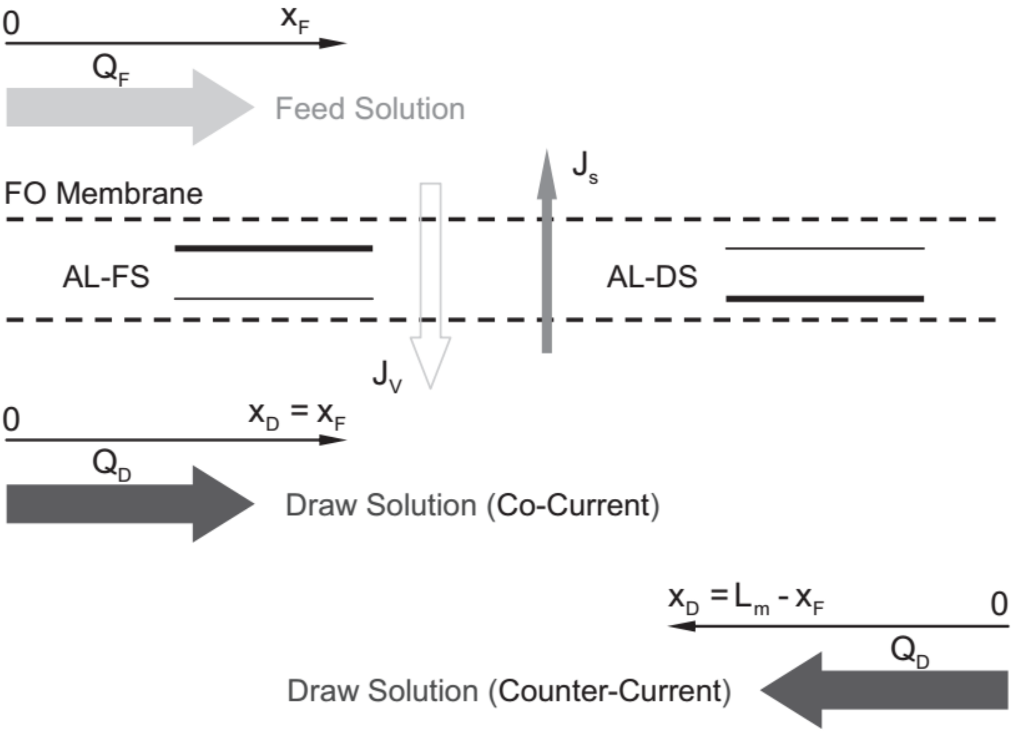


Figure 2.8. Discretization of spiral wound membrane for module scale simulation**.** Adapted from (Xiao et al. 2012)

The Effects of operating conditions and membrane properties on the hollow fiber module performance were theoretically estimated employing the models.

For the hollow fiber FO simulation another friction concentration polarization (FCP) based model was developed by including the effects of feed and draw side pressure drop with the concentration polarization as follows (Shibuya et al. 2016):

$$J_w = A\left(\pi_D exp\left(-\frac{J_w}{k_F}\right) - \pi_F \exp\left(\frac{J_w S}{D}\right) - \Delta P\right) \tag{2.21}$$

where ΔP is the pressure difference between the shell and the bore side of the fiber. It considered the radial flow of solution in the shell side as shown in Figure 2.9. Pressure difference is evaluated by employing the pressure drop on the shell and bore side as follows:

$$\frac{dP_s}{dr} = \frac{150(1-\varepsilon)^2}{\varepsilon^3}\frac{\mu_s V_s}{(1.5d_s)^2} + \frac{1.75(1-\varepsilon)}{\varepsilon^3}\frac{\rho_s {V_s}^2}{1.5d_s} \tag{2.22}$$

$$\frac{dP_b}{dZ} = \frac{128\mu_b u_b}{\pi {d_{in}}^4} \tag{2.23}$$

However, since the pressure difference varies in radial and axial direction, local pressure difference was calculated by discretizing the module as shown in

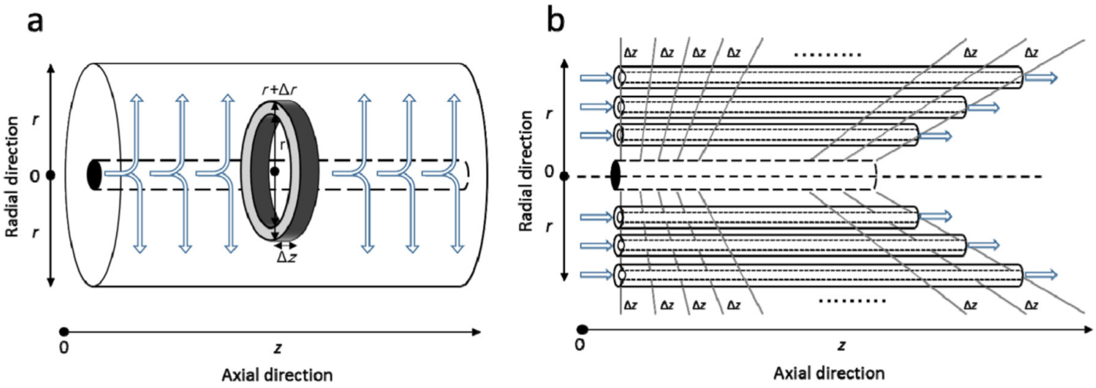


Figure 2.9. Schematic diagram of the hollow fiber membrane module for discretization of (a) the shell side and (b) the bore side. Adapted from (Shibuya et al. 2016)**.**

These models were used to investigate the effects of different operating conditions such as solution flowrate, solute concentration on the performance of a 5-inch scale hollow fiber module. Most hollow fiber modeling studies used the mass transport equation of the flat sheet membrane for hollow fiber simulation neglecting the geometrical difference between these two. A modified model has been developed considering the curvature effect of the hollow fiber (Lin 2016) as shown in Figure 2.10.

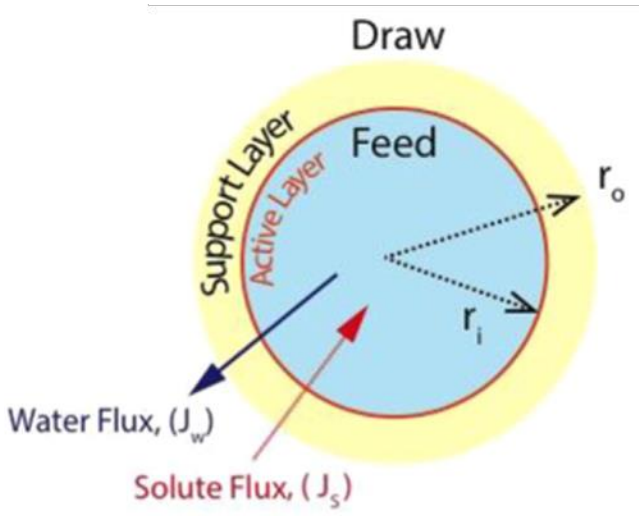


Figure 2.10. Key geometric features of a hollow fiber membrane. Adapted from (Lin 2016)

At any axial distance radial distribution of the solution concentration given by,

$$-J_s(r) = -D_s \frac{dC(r)}{dr} + J_w(r)C(r) \tag{2.24}$$

Considering the curvature effect the modified flux equation is given by,

$$J_w = A\left[\left(\frac{B}{A} + \pi_D\right)\left(\frac{r_0}{r_i}\right)^{-\frac{J_w\sigma}{D}r_i} - \left(\frac{B}{A} + \pi_F\right)\left(\frac{r_i}{r_i - D/k}\right)^{\frac{J_w}{D}r_i}\right] \tag{2.25}$$

here, r_0 and r_i are the inner and outer diameter of the fiber whereas σ is the ratio of tortuosity and porosity of the membrane.

Recently another study added the hydraulic pressure difference across the membrane with the water and solute transport equation to simulate the performance of a commercial CTA hollow fiber membrane module (Teklu, Gautam & Subbiah 2020). The model used in this study is given by,

$$J_w = A\left[\frac{\pi_D exp\left(\frac{-J_w}{K_D}\right) - \pi_F exp\left[J_w\left(\frac{S}{D}+\frac{1}{k_F}\right)\right]}{1+\frac{B}{J_w}\left\{exp\left[J_w\left(\frac{S}{D}+\frac{1}{k_F}\right)\right] - exp\left(\frac{-J_w}{K_D}\right)\right\}} - (P_D - P_F)\right] \qquad (2.26)$$

here, P_D and P_F are the applied hydraulic pressures on the feed and draw side.

2.4.3 Plate and frame modeling

This section summarizes the research effort to model and simulate the performance of a plate and frame type FO module. Although numerical studies for a commercial plate and frame module are very rare, most studies considered a co-current and counter-current flow over a flat sheet membrane.

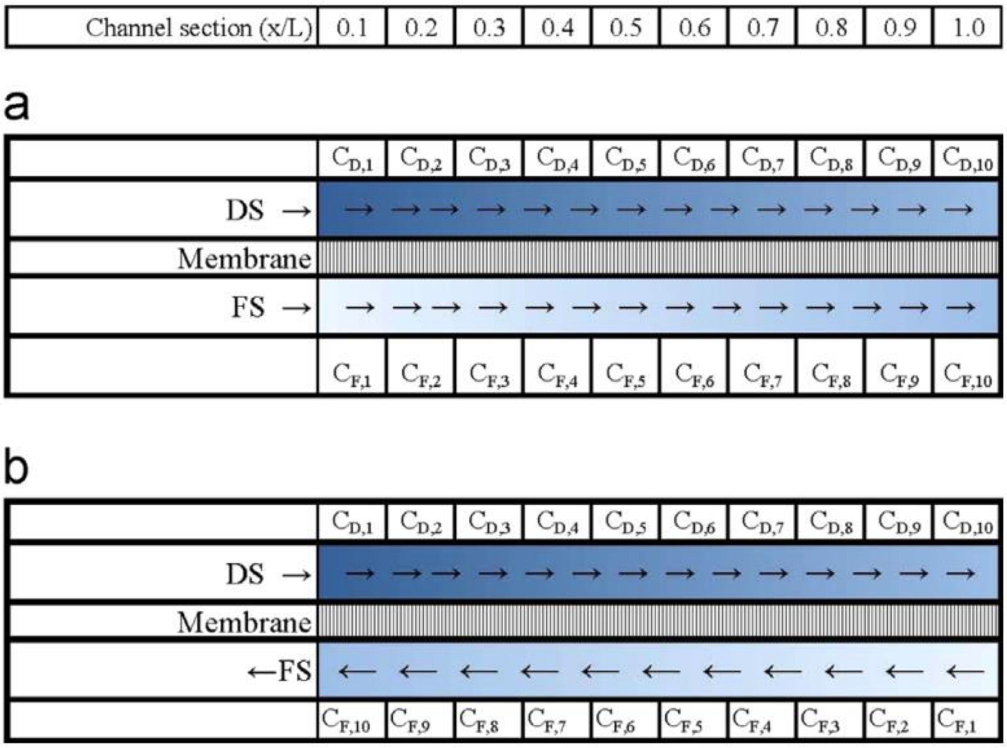


Figure 2.11. Modeling of a flat sheet FO module using (a) co-current and (b) counter-current flow path. Adapted from (Phuntsho et al. 2014).

In the commercial plate and frame module crossflow configuration was observed for the feed and draw solution, which means the draw solution flows at a right angle to the feed solution. However, the studies that considered flat sheet membrane and co or counter-current flow configuration are reviewed as the plate and frame module scale modeling study in this section. The concept of osmotic equilibrium in a FO membrane module was introduced by a module scale modeling and simulation study (Phuntsho et al. 2014). This study used a co-current and counter-current flow behavior (as shown in Figure 2.11) to simulate the performance of a 1 m long membrane module.

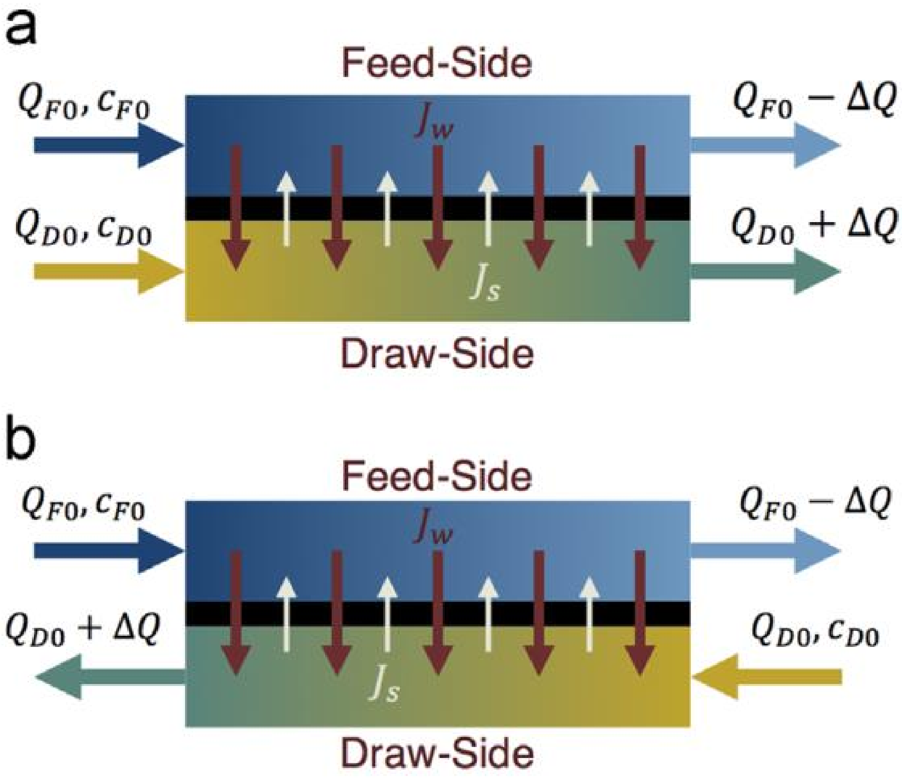


Figure 2.12. Discretization of flat sheet membrane and mass balance for module scale modelling considering (a) co-current and (b) counter-current flow orientation. Adapted from (Deshmukh et al. 2015)

In another module scale study Deshmukh et al. discretised the flat membrane area (Figure 2.12) to determine the limit of FO module operation as a function of membrane properties such as y water permeability, solute permeability, and structural parameter considering co-current and counter-current flow through the membrane channel (Deshmukh et al. 2015). Another set of analytical models was presented to calculate the required membrane area of a flat sheet module for a required recovery ratio or dilution factor (Banchik et al. 2016). A similar discretization technique as presented in the other study was used to apply the mass conservation laws. In addition, a theoretical model for maximum recovery ratio was also presented in this study as follows,

$$RR_{max} = MR\left(\frac{\theta_d}{\theta_f} - 1\right) \tag{2.27}$$

here, RR_{max} is the maximum recovery ratio of the system, MR is the mass flow ratio of the feed and draw solution, θ_d and θ_f are the osmotic pressure of draw and feed solution. It also defined the dilution factor of a membrane module by,

$$DF = \frac{RR}{MR + RR} \tag{2.28}$$

Optimum membrane area of an FO membrane module was analytically calculated by a slightly modified model (Mondal, Field & Wu 2017). Unlike most of the studies, this study considered a log mean osmotic pressure difference as the driving force for the FO process when the other studies consider the osmotic pressure difference between draw solution and feed solution as the driving force. The revised equation for the water permeation calculation using the log mean pressure difference is expressed by (Mazlan, Peshev & Livingston 2016a),

$$Q_P = L_p \left[\frac{ln\left(\frac{\Delta\pi_1}{\Delta\pi_2}\right)}{\Delta\pi_1 - \Delta\pi_2} \right] \nu_{H_2O} A_m \tag{2.29}$$

here, Q_P is the volumetric permeation rate, L_p represents the water permeability coefficient, ν_{H_2O} is the kinematic viscosity and $\frac{ln\left(\frac{\Delta\pi_1}{\Delta\pi_2}\right)}{\Delta\pi_1 - \Delta\pi_2}$ stands for the log mean osmotic pressure difference across the module.

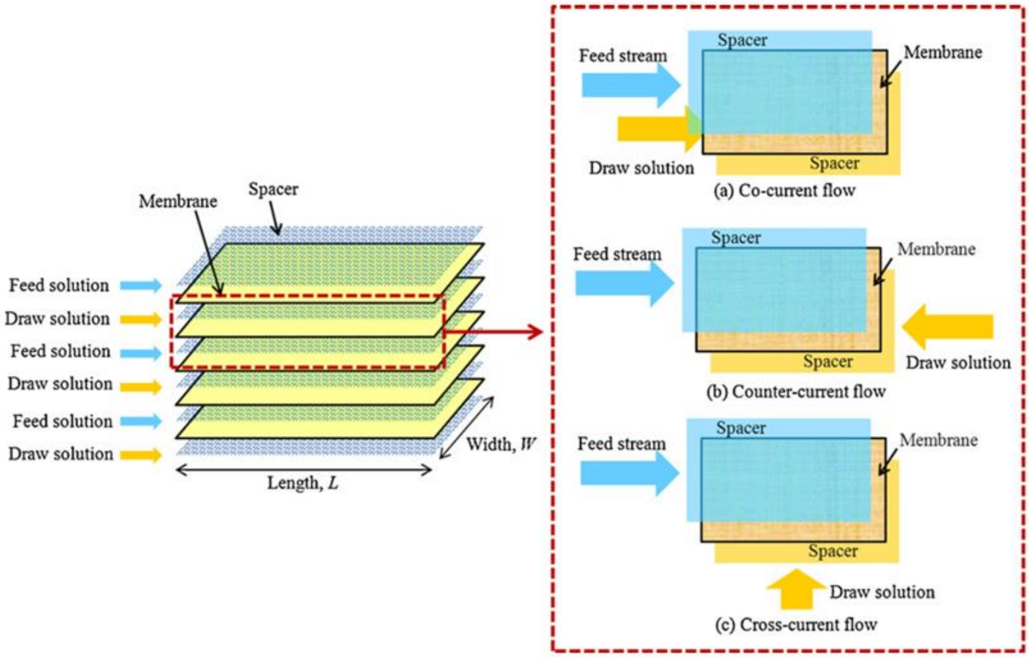


Figure 2.13. Flow configurations considered for the modeling of a plate and frame type module simulation. Adapted from (Gu et al. 2011a).

Plate and frame type module performances were numerically simulated for co-current, counter-current, and cross-current flow (Gu et al. 2011b) as shown in Figure 2.13. For the local distribution of velocity and concentration taking into account the dilution of draw solution, the same models are employed as the spiral wound module which is described in section 2.4.1.

Most of these studies did not consider the actual dimensions of the membrane module. Moreover, these studies considered that the FS and DS flow along the length direction of the membrane only. In addition to this, the reliability of finding of a theoretical study is highly dependent on its agreement with the experimental studies. But these studies did not validate their findings by comparing with the experimental results.

2.5 Module scale FO system optimization

Full scale FO system performance is generally defined by its recovery rate and dilution factor. With some exceptions, most studies in the literature investigated the performance of a module as a function of various operating conditions. Few studies optimized the operating conditions or module design parameters to obtain either the maximum recovery rate or highest dilution factor. For example, Mondal et al. showed the membrane size requirements for different recovery fraction and concentration ratios separately, as shown in Figure 2.14. Performance of FO system is compared in terms of recovery fraction (ϕ) for a specific dilution factor (c20/c10) at different DS to FS flow ratio (V20/V10) for various membrane areas A_0 employing co-current and counter-current configuration flow. Higher recovery rate is achieved for counter-current configuration at higher flow ratios and larger membrane area. In general, these studies optimize the operating conditions by maximizing the recovery rate only. However, it was found that the recovery rate and DS dilution factor are inversely related (Mondal, Field & Wu 2017; Phuntsho et al. 2017; Phuntsho et al. 2016). Therefore, the optimum design variables and operating conditions should be selected considering both recovery rate and concentration requirement.

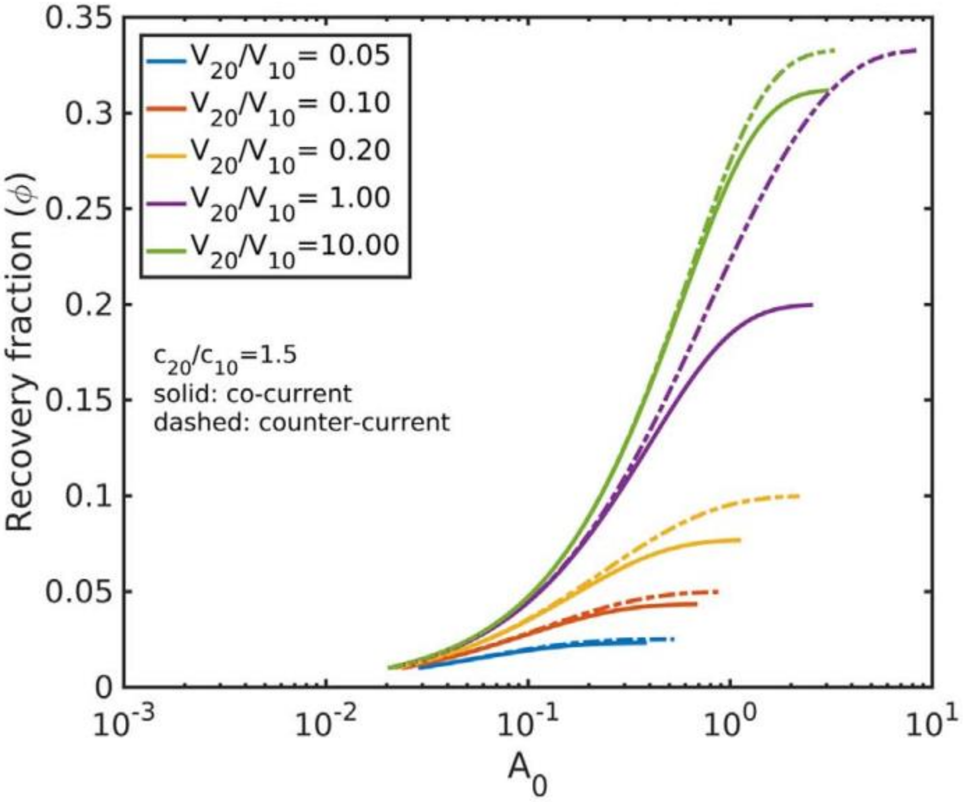


Figure 2.14. Recovery fraction as a function of membrane area at different concentration of the solution. Adapted from (Mondal, Field & Wu 2017).

Effects of the hollow fiber membrane length on the module performances were investigated in terms of maximum recovery ratio and flux production to optimize the module design (Xiao et al. 2012). Minimum power consumption by the system at different operating conditions and membrane length was also considered to optimize the hollow fiber module design. Teklu et al. optimized a CTA commercial hollow fiber module by determining the minimum energy consumption for various membrane length and solution flowrates as shown in Figure 2.15. It considered the axial flow through both shell and bore side of the module. It optimized the operating conditions by comparing the effects of membrane length, DS and FS flowrates, and hydraulic pressure on the energy consumption of the system. However, none of these studies explored the optimal membrane modular arrangement which is essential to design a full scale FO system. Moreover, no design and optimization software is available so far for the FO process.

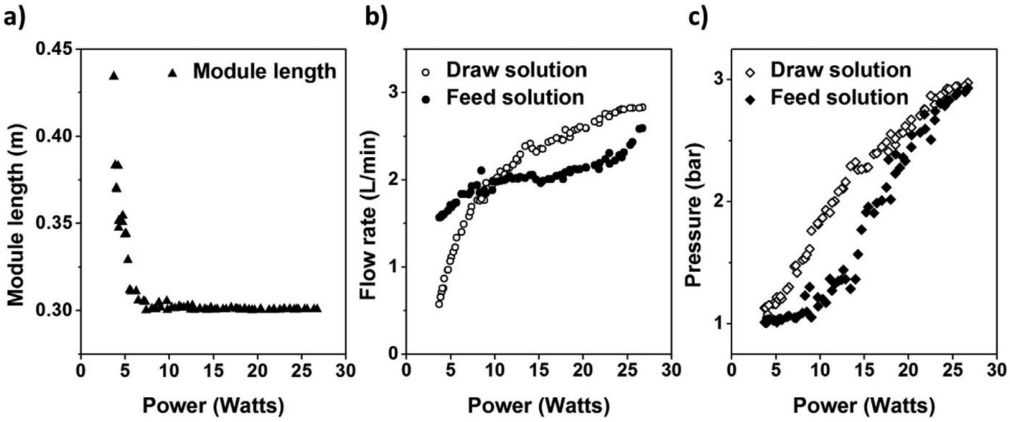


Figure 2.15. Power consumption of commercial CTA hollow fiber module for different (a) module lengths, (b) flowrates, and (c) inlet pressures. Adapted from (Teklu, Gautam & Subbiah 2020)

From the above review of literature, it clearly shows that a novel optimization algorithm should be developed to optimize the design and operating conditions of the FO process. The simulations results are also needed to be compared with the experimental results to validate the models and assumptions used for the simulation. Moreover, it is pivotal to develop a simulation software to design and optimize the FO process efficiently.

2.6 Conclusions

This chapter begins with the introduction of forward osmosis (FO) and its application in water filtration. The mass transport mechanism and FO commercial membrane fundamentals have been discussed. Further, FO membrane modules have been comprehensively reviewed in this chapter. The process simulation and modeling is an effective tool in modern research to accurately predict process performance and its optimization. The review briefly discusses the basic equations, mechanisms, and mathematical models for all typical membrane modules employed in the FO process for water filtration. To the best of the authors' knowledge, this would be the first systematic effort to evaluate the system scale optimization of the FO process to develop a system analysis software for full scale plant design which has been discussed in detail in this dissertation. This will definitely provide deep insight to understand the influence of different operating and design parameters on system performance and help with FO process commercialization for real scale applications.

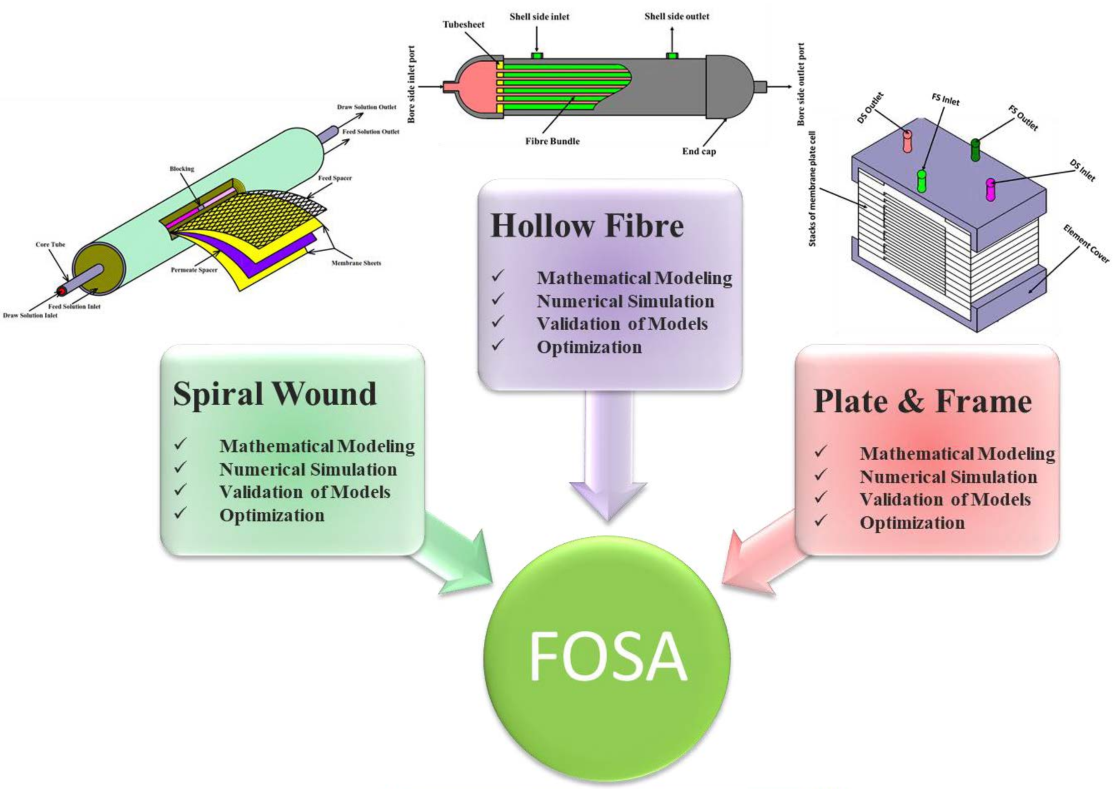
CHAPTER 3

GENERAL METHODOLOGY

3 General methodology

3.1 Introduction

This thesis designed and optimized a FO system for large scale applications. It employed the theoretical analysis technique to determine the design information and optimum operating conditions.



Hollow Fibre

- ✓ Mathematical Modeling
- ✓ Numerical Simulation
- ✓ Validation of Models
- ✓ Optimization

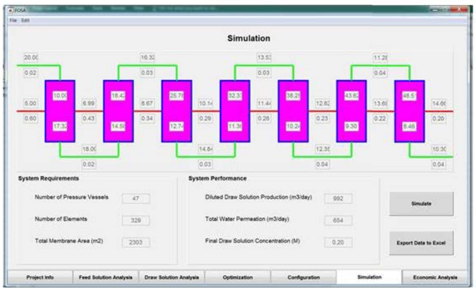
Spiral Wound

- ✓ Mathematical Modeling
- ✓ Numerical Simulation
- ✓ Validation of Models
- ✓ Optimization

Plate & Frame

- ✓ Mathematical Modeling
- ✓ Numerical Simulation
- ✓ Validation of Models
- ✓ Optimization

FOSA



System Requirements

Number of Pressure Vessels 47

Number of Elements 329

Total Membrane Area (m2) 2303

System Performance

Diluted Draw Solution Production (m3/day) 992

Total Water Permeation (m3/day) 654

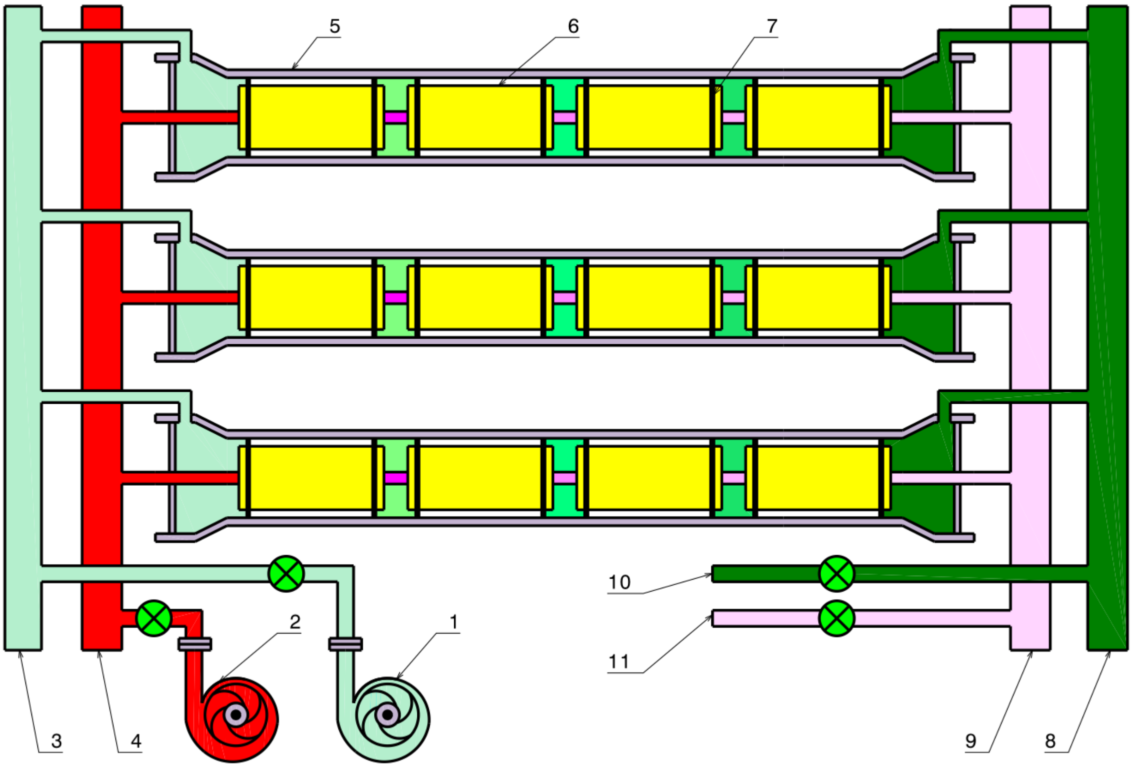
Final Draw Solution Concentration (M) 0.20

Figure 3.1. Flow chart of the research activities

In this theoretical work, mathematical models were first developed to estimate mass transport through the FO membrane. The mass transport models were then coupled with the fluid conservation laws to elucidate the fluid flow behavior in the feed solution and draw solution channel of the membrane module. Since the solution velocity and concentration changes inside the channel along the flow path, the mass transport behavior also varies at different locations over the membrane. Therefore, this study numerically solved the mathematical models to calculate the solution flowrate, velocity, concentration, water flux, and reverse solute flux at discrete locations over the membrane considering the actual flow orientation and physical dimensions of the modules. Analytical models were also developed to evaluate the module scale performance in terms of diluted draw solution production rate, final draw solution concentration, and recovery rate considering the local mass transport, solution flowrates, and concentration. In addition, pilot scale experiments were also conducted to measure the performance of the membrane module at the same operating conditions. Experimental results were then compared with the theoretical findings to assess the accuracy of the mathematical models developed in this work. Finally, after the validation of the models, a novel optimization algorithm was designed to find the optimum membrane modular configuration and operating conditions of a full scale FO system. However, three different types of FO membrane modules are commercially available at this moment, such as spiral wound, hollow fiber and plate and frame module. As these modules are very unique in terms of membrane geometry and flow orientation, the mathematical models to simulate their performance are also different from each other. Therefore, this thesis carried out three different theoretical studies to design and optimize the FO system using each module as shown in Figure 3.1. Finally, combining the findings of these three studies a MATLAB based software FOSA (forward osmosis system analysis) was developed to make the FO system design more efficient and reliable. The detailed methodologies for FO system design using spiral wound, hollow fiber and plate, and frame module are explained in the respective chapters (chapter 4, 5 and 6), but the general methodologies irrespective of the module type such as mathematical modeling, numerical simulation, validation of theoretical models, optimization algorithm and software development are discussed in the subsequent sections of this chapter.

3.2 Full-scale FO system description

Figure 3.2 shows the schematic diagram of a full-scale FO plant. It comprises four main components such as FS pump, DS pump, pressure vessel, and membrane element. Feed and draw solutions of the plant are selected according to their concentrations, availability, cost, and the final product of the system (Corzo et al. 2017). As osmotic pressure is a function of their concentrations, the concentration of DS must be higher than the concentration of the FS. The membrane elements are housed inside the pressure vessel as shown in Figure 3.2. The feed and draw solutions are supplied to the pressure vessels through an inlet distribution manifold by FS and DS pumps respectively. Every single vessel has a minimum of four ports. Generally, two end port connections are used for DS inlet and outlet, whereas the remaining two side port connections are for FS inlet and outlet. FS enters at one end of the vessel through the side port and flows along the length direction of the modules through the feed channels. Brine seals are used to prevent the FS from going around the elements rather than through it. On the other hand, DS enters through the end port of the vessel. Due to the water potential difference between the FS and DS resulted from the concentration difference, pure water permeates from FS to DS across the membrane. Therefore, DS concentration decreases and FS concentration increases according to their direction of flows.



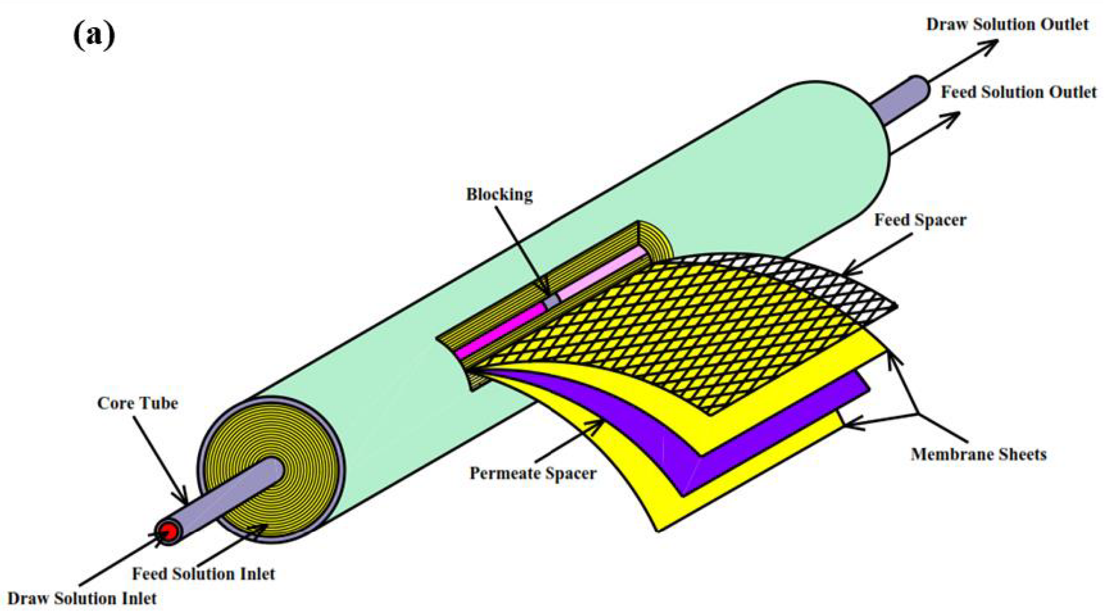
1	FS pump	2	DS pump
3	FS inlet distribution manifold	4	DS inlet distribution manifold
5	Pressure vessel	6	Membrane elements
7	Brine seal	8	FS outlet distribution manifold
9	DS outlet distribution manifold	10	Concentrate disposal pipeline
11	FO product pipeline		

Figure 3.2. Schematic diagram of a full-scale FO system.

FS concentrate and diluted DS outlet of the first element then enters the second element as FS and DS inlet respectively. In the same way, the solutions pass through the elements until the required FS recovery rate and the final concentration of DS are not reached. The number of elements per pressure vessel is decided according to the required recovery rate and product concentration. Finally, the diluted DS or the product of FO system is delivered from the pressure vessel through the DS outlet distribution manifold. On the other hand, concentrated FS is disposed through the FS outlet distribution manifold.

3.3 Description of spiral wound FO module

A commercial thin film composite (TFC) 8040 SWFO element was used for this study which was fabricated by Toray Chemical Inc. The SWFO element is fabricated by integrating membrane sheets, feed spacers, and permeate spacers with a core tube via adhesive to create separate flow channels for DS and FS as shown in Figure 3.3(a). The FS channel of the module is formed by placing a feed spacer between two membranes sheets and gluing the sheets along their length and width. In contrast, the DS channel is formed by obstructing the core tube at the middle as well as gluing the sides and the centreline of the sheets as shown in Figure 3.3(b). As the core tube is blocked at centre and has holes on its body, DS comes out through the holes and flows along the width direction over the membrane sheet as shown in Figure 3.3(a) and (b). In addition to this, since the DS channels are divided by the central glue lines, the solution takes a U-shaped path of flow and returns to the outlet side of the core tube.



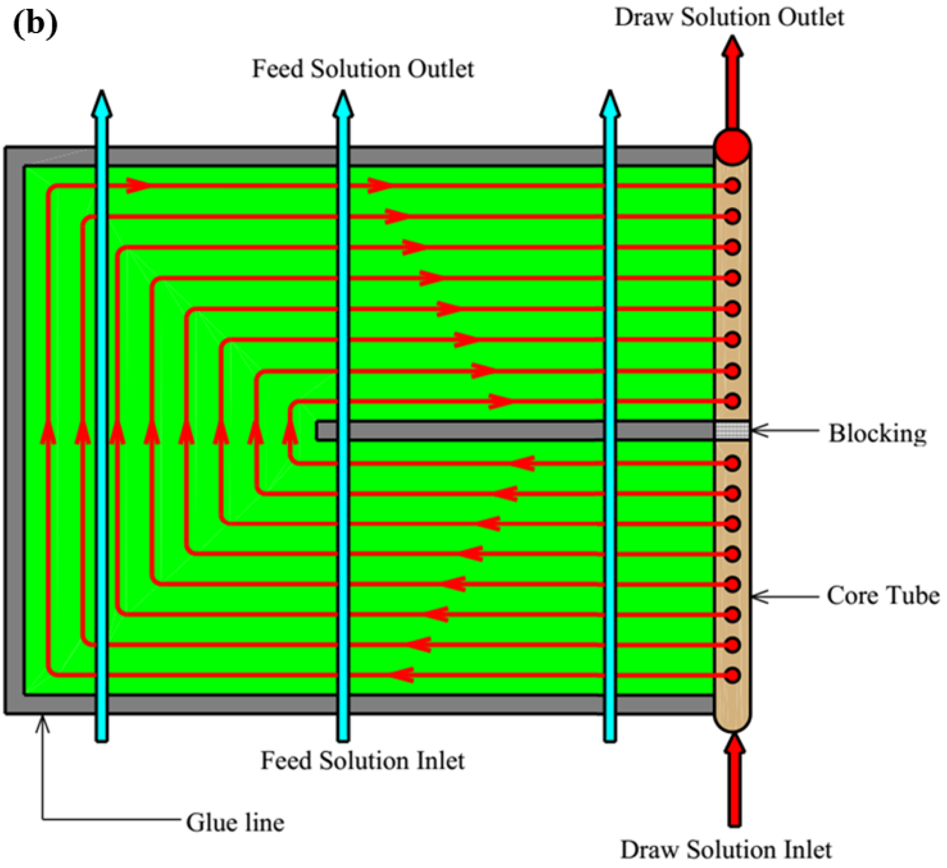


Figure 3.3. Description of (a) spiral wound module and (b) flow configuration.

3.4 Description of hollow fiber module

Membrane module is one of the most important components of a real scale FO plant. This section outlines the physical description of the module and the solution flow configuration inside an outer selective CTA hollow fibre module (manufactured by Toyobo Co. Ltd.) used in this study. Figure 3.4 schematically describes the membrane module used in this study for the analysis.

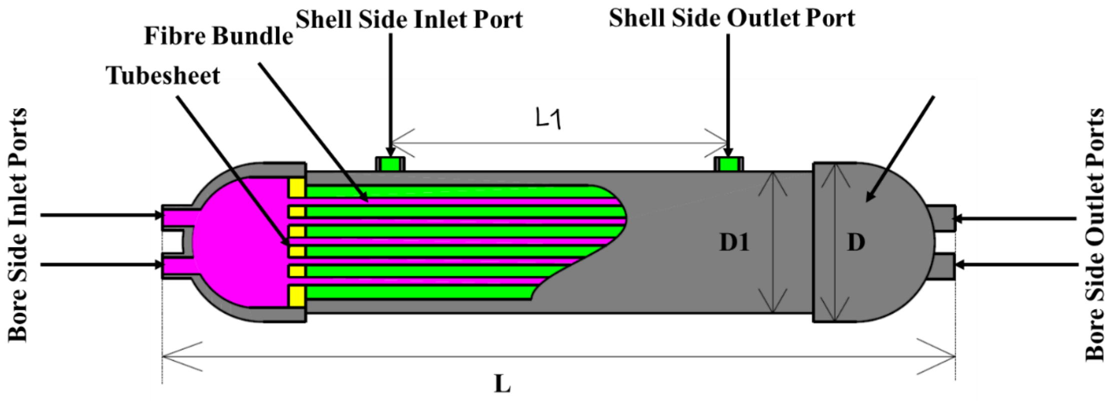


Figure 3.4. Description of the hollow fiber module used for this study (here L = 830 mm, L1 = 430 mm, D = 103 mm, D1 = 90 mm).

The analysed module consists of a bundle of membrane fiber, housing, two tubesheets, and two end caps. Fiber bundle is the main functional unit of the hollow fiber module. Numerous fibers are arranged in different geometric configurations to form the bundle such as parallel, cross-wound, etc. The current module has a parallel fiber bundle. The number of fiber in the bundle defines the packing density of the module. The fiber bundle creates two separate zones within the module. The channel through the bores of the membrane fiber is referred to as the bore side in this study, whereas the area between the housing and the membrane outer surfaces is termed as the shell side. However, the whole fiber bundle is assembled within the cylindrical housing of the module. At the two ends of the module, it has an inlet port and an outlet port for the solution to flow through the shell side. Two end caps are also installed to facilitate another solution to flow through the bore side of the membrane. A tubesheet is used between the end caps and the housing to isolate the flows between the shell and bore side as shown in Figure 3.4. Flow configurations in the proposed hollow follow fiber module is longitudinal as the parallel fiber bundles are used in this module. For FO processes, since the membranes are outer selective FS flows through the shell side of the module and the DS flows through the bore side of the membrane.

3.5 Description plate and frame module

This section describes the plate and frame type membrane module manufactured by Porifera Inc., USA. Each membrane module consists of several membrane elements (PFO100) connected in the serial configuration according to the design requirements. The membrane elements consist of several stacks of membrane plate cells. Each membrane plate cell includes a feed and a draw channel formed by gluing the spacer plates and the membrane sheets on the opposite sides of the spacer as shown in Figure 3.5(a) and (b). Spacer plates have two sets of openings on its structure. One set of openings makes the fluid flow path through the membrane plate cells when those are stacked on each other. The second set of openings brings the water from the first opening to the corresponding feed or draw channel. DS is supplied to the membrane elements through the DS inlet manifolds. It enters the fluid manifolds of the membrane elements from either side of the membrane sheets in the width direction of the membrane sheets. In the current study, it was considered that the DS enters from the left side of the element and passes through the draw channel (formed by the two adjacent membrane sheets separated by the spacer plate) along the length direction of the membrane sheets. On the other hand, FS is supplied by the feed solution supply network to the membrane elements. It enters the elements through the fluid ports in the length direction of the membrane sheets and passes through the feed channel (on the upper or lower side of the draw channel) across the width direction of the membrane sheets. The membrane sheets are bonded to the spacer plate along the width of the plate to isolate the FS from mixing with the DS. Therefore, the isolated feed and draw solution flows orthogonally (90°) to each other.

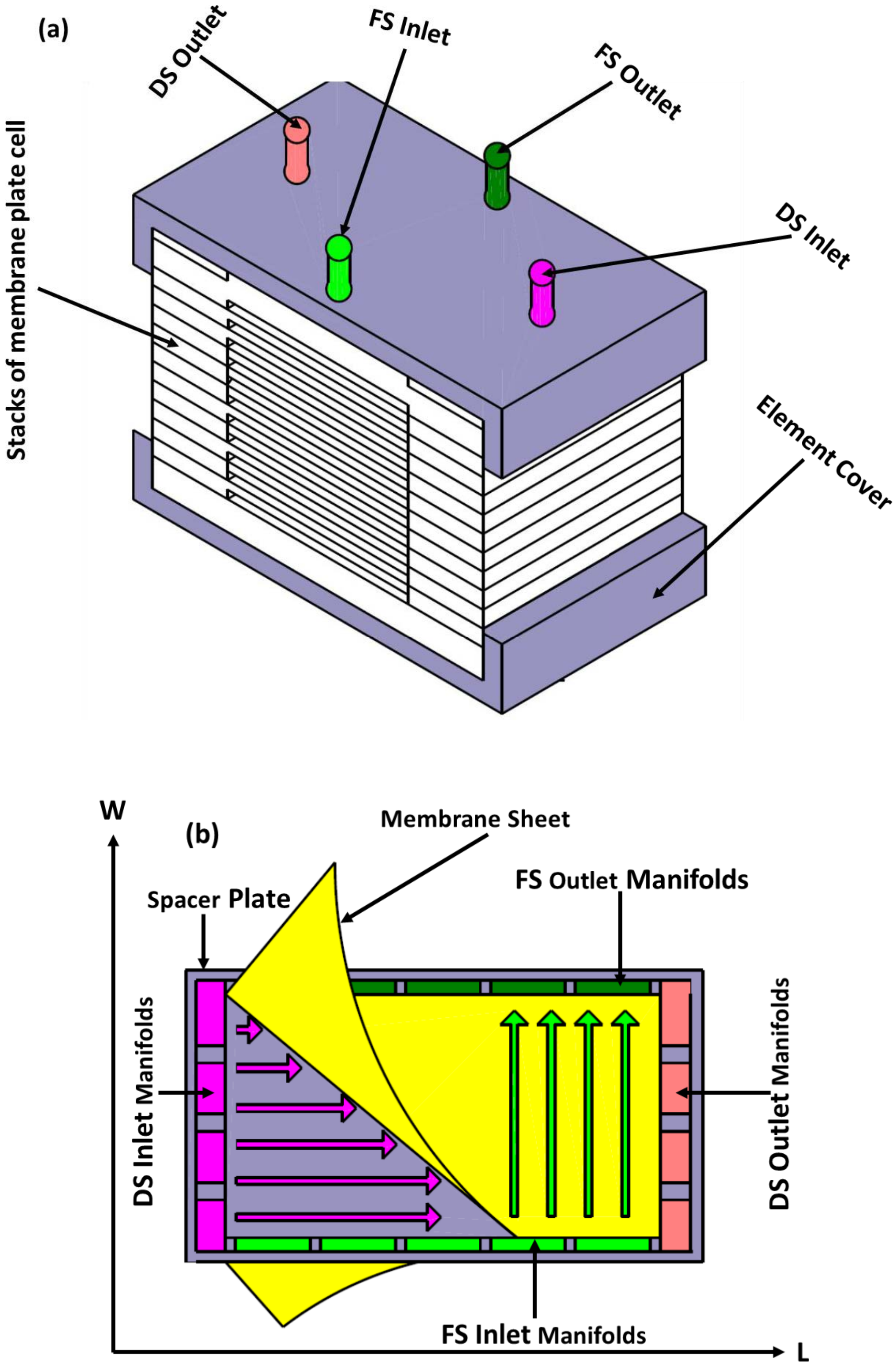


Figure 3.5. (a) Sectional view of a plate and frame FO membrane element and (b) membrane plate cell with its solution flow configurations.

3.6 Mathematical models and numerical simulation

This section outlines the major theoretical models used in this thesis to investigate the mass transport and fluid flow behavior. It also explains the numerical simulation procedures for the performance analysis of the membrane module.

3.6.1 Mathematical models

Mass transport models are employed to determine the water flux and reverse solute flux through the membrane resulted from the osmotic pressure gradient of two solutions separated by a semipermeable FO membrane. Therefore the mass transport through the FO membrane is a function of the membrane properties and the osmotic pressure difference of the solutions (Lee & Ghaffour 2019; Phuntsho et al. 2014). However, external concentration polarization (ECP) and internal concentration polarization (ICP) effects influence the osmotic pressure difference across the membrane. But ECP and ICP effects depend on the mass transfer coefficient and the membrane structural parameter. Considering the osmotic pressure of the solution, mass transfer coefficient, and membrane properties, water flux at any location through the membrane surface is given by (Shaffer et al. 2015; Song et al. 2018; Tiraferri et al. 2013),

$$J_w = A\left[\frac{\pi_D \exp\left(-\frac{J_w S}{D}\right) - \pi_F \exp\left(\frac{J_w}{k_F}\right)}{1+\left(\frac{B}{J_w}\right)\left(\exp\left(\frac{J_w}{k_F}\right) - \exp\left(-\frac{J_w S}{D}\right)\right)} + (P_F - P_D)\right] \tag{3.1}$$

here, π_D and π_F are the bulk osmotic pressure of the feed and draw solution, A and B are the water permeability and solute permeability coefficients, S is the structural parameter of the membrane which is a function of thickness, porosity, and tortuosity of the membrane and D is the diffusivity of the draw solution. Moreover, the applied hydraulic pressure in the feed and draw solution channel also influence the mass transfer. P_F and P_D are the hydraulic pressure in the feed and draw channel. Usually, the hydraulic pressures are considered negligible for the FO processes as these processes are osmotically driven and a very small pressure is needed to overcome the hydraulic resistance for the solution flow. Similarly, the reverse solute flux through the FO membrane at any location is expressed by,

$$J_s = B\left[\frac{C_D \exp\left(-\frac{J_w S}{D}\right) - C_F \exp\left(\frac{J_w}{k_F}\right)}{1 + \left(\frac{B}{J_w}\right)\left(\exp\left(\frac{J_w}{k_F}\right) - \exp\left(-\frac{J_w S}{D}\right)\right)}\right] \tag{3.2}$$

here, C_D and C_F are the bulk concentration of the draw and feed solution.

Due to the permeation of water and reverse solute diffusion through the membrane, the fluid flow behaviors, and solution concentration changes over the membrane surface along the path of the solution flow. The fluid mass conservation law is applied to determine the flow rate, velocity, and concentration. Changes in draw solution flowrate for a small change in membrane length or width is given by,

$$\frac{dQ_D}{dl \times dw} = J_w \tag{3.3}$$

here, Q_D is the draw solution flowrate through a very small section over the membrane, $dl,$ and dw are the length and width of the discrete section of the membrane. Feed solution flowrate along the length or width direction is expressed by,

$$\frac{dQ_F}{dl \times dw} = -J_w \tag{3.4}$$

here, Q_F is the feed solution flowrate. In addition, since the water permeates from the feed side to the draw side, water flux J_w shows an opposite sign. Similarly, the draw solute flowrates were calculated by,

$$\frac{dQ_{D,slt}}{dl \times dw} = -J_s \tag{3.5}$$

where, $Q_{D,slt}$ is the draw solute flowrate. Similarly, the feed solute flowrate was given by,

$$\frac{dQ_{F,slt}}{dl \times dw} = J_s \tag{3.6}$$

Considering the amount of solute in the solution the draw solution and feed solution concentration are presented by,

$$C_D = \frac{Q_{D,slt}}{Q_D} \tag{3.7}$$

$$C_F = \frac{Q_{D,slt}}{Q_D} \tag{3.8}$$

Fluid continuity theorem is applied to calculate the velocity. Draw solution velocity is calculated by,

$$U_D = \frac{Q_D}{dA} \tag{3.9}$$

here, U_D is the draw solution velocity. Feed solution velocity through any small section is expressed by,

$$U_F = \frac{Q_F}{dA} \tag{3.10}$$

3.6.2 Numerical simulation

Mass transport models for water flux is a highly non-linear and implicit equation. It has the unknown variable, water flux (J_w) on both sides of the equation and this variable cannot be separated. Therefore, this equation cannot be solved in a typical method. Therefore, it requires an iterative technique to solve this equation.

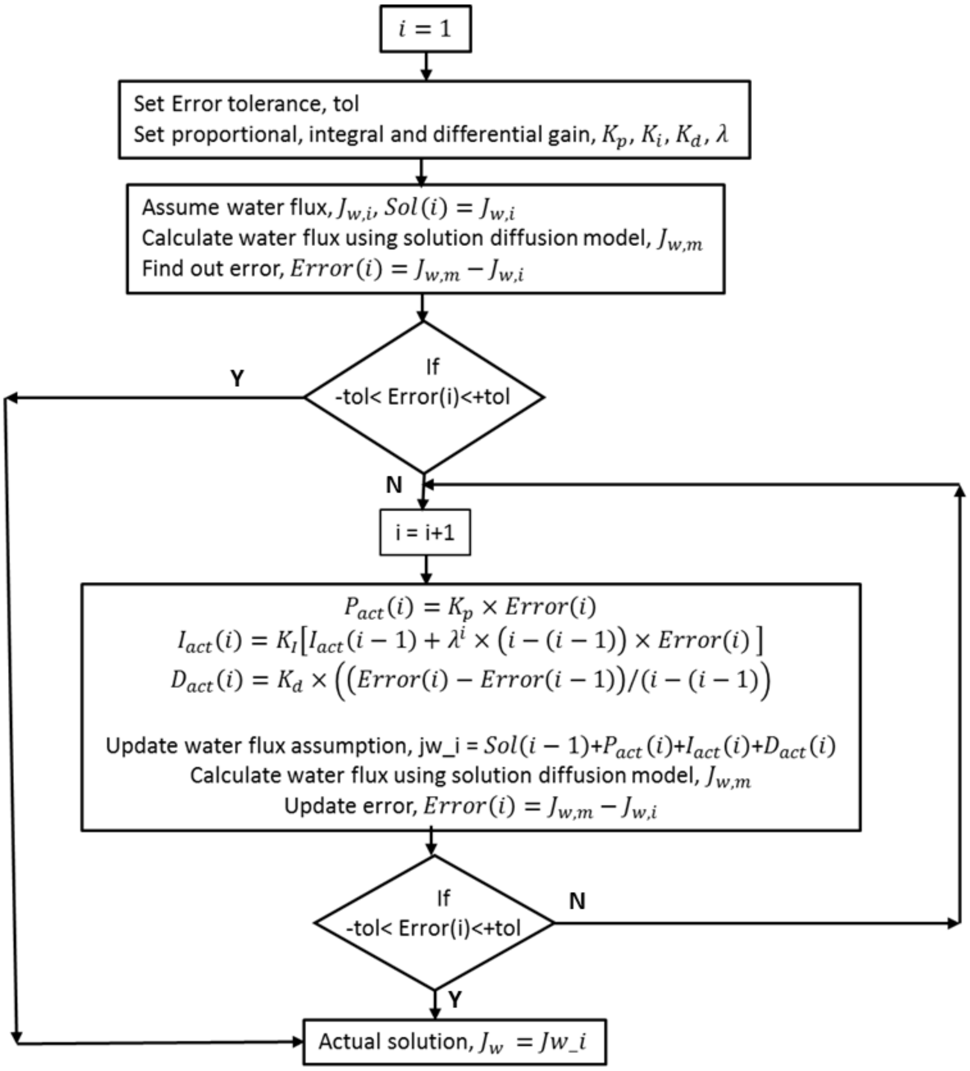
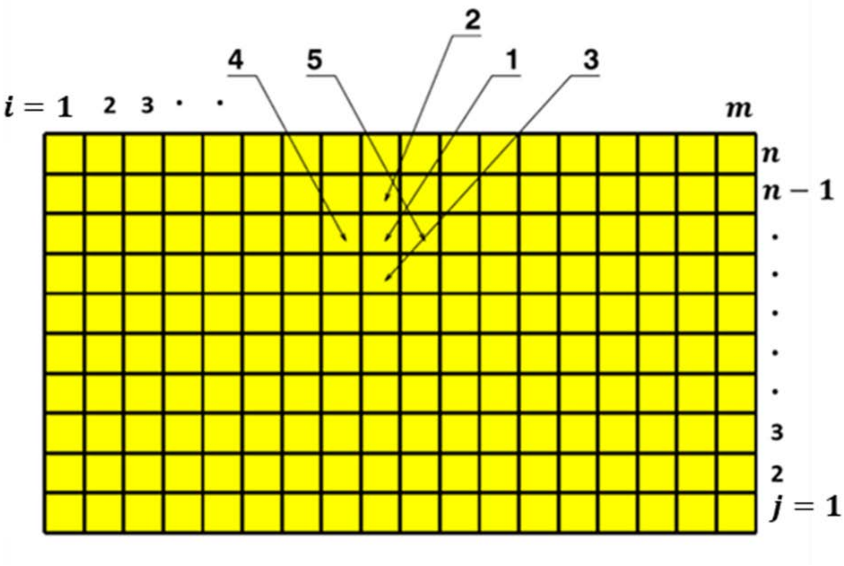


Figure 3.6 Algorithm flowchart for the solution of mass transport equation.

The algorithm flowchart for the solution of this equation is shown in Figure 3.6. It can be seen from the figure that the algorithm assumes a solution initially ($J_{w,i}$). Using this assumed value on the right-hand side of the transport equation, it calculates a new value of water flux ($J_{w,m}$), referred to as modeled water flux in this section. Then the assumed value is subtracted from the modeled value to find the error. This error is compared with a set value of error tolerance. In this study the error tolerance was 1e^{-20}. If the error value is less than the error tolerance the assumed flux value is considered as the actual solution of the model.

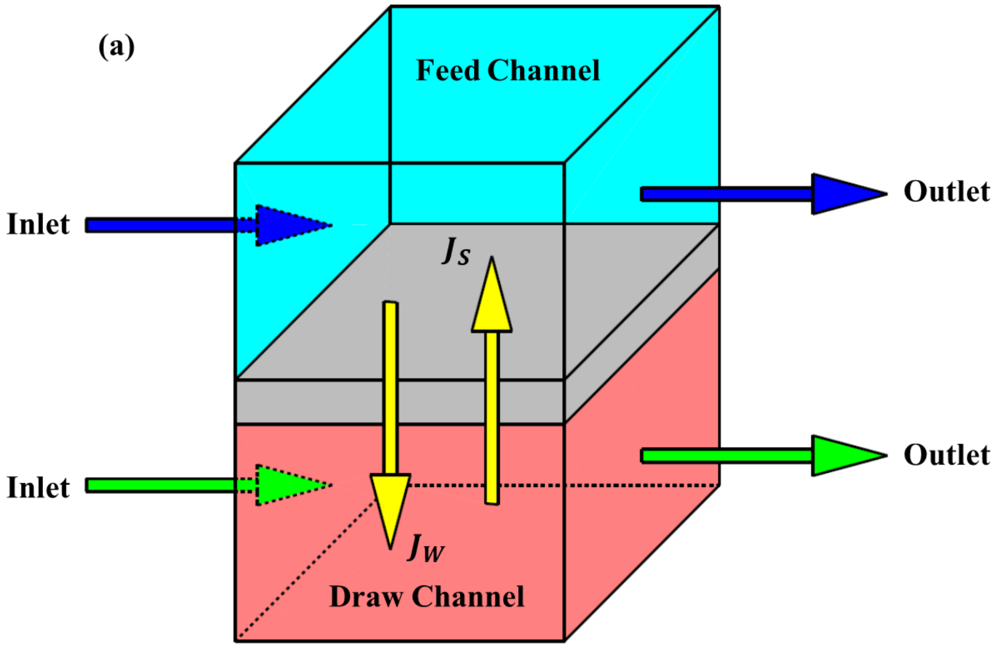
However, if the error is greater than the error tolerance a new technique was employed to minimize the error and find the solution of the mass transport model. PID (proportional-integral-derivative) control technique is widely used to minimize the error between the desired process variable and the actual process variable using a feedback loop. The same technique is adapted here to minimize the error between the assumed flux value and modeled flux value to an acceptable level. According to this technique, four constant parameters (such as proportional gain, integral gain, derivative gain, and stability factor) were defined. As shown in Figure 3.6, the new assumed value of water flux is calculated by using the PID control method. The error value is multiplied by the proportional gain (K_P), which ensures that the change in assumed flux value is proportional to the error. It means that if the error in the previous calculation is large, the change in assumed flux value will also be large. On the other hand, if the error is small the assumed flux value will also be changed proportionally from its previous values. However, only proportional control cannot completely minimize the error. As the error reduces to a very small value and then the assumed flux value does not change significantly. Consequently, a constant error is noticed. In order to avoid this problem, an integral control technique also employed. It changes the assumed flux value according to the error accumulated over the iterations and implemented by multiplying the integral gain (K_I) with the accumulated error. Finally, derivative control is used to minimize the rate of change of error. If there is a sudden increase or decrease in the error, derivative gain (K_D) is multiplied by the rate of change of error to update the assumed flux value. After adding the proportional, integral and derivative control effects, the previously assumed flux value is updated by this new assumed flux value ($J_{w,i}$). This new value is again used in the mass transport model to calculate the new modeled flux value $J_{w,m}$. If the error between the $J_{w,i}$ and $J_{w,m}$ is greater than the error tolerance, the next iteration starts which executes the same control technique described above. After several iterations when the error is less than the error tolerance, the algorithm returns the assumed flux value $J_{w,i}$ of that particular iteration as the actual solution of the mass transport model for water flux. Once the water flux is calculated, this value is used in equation (3.2) to calculate the reverse solute flux.

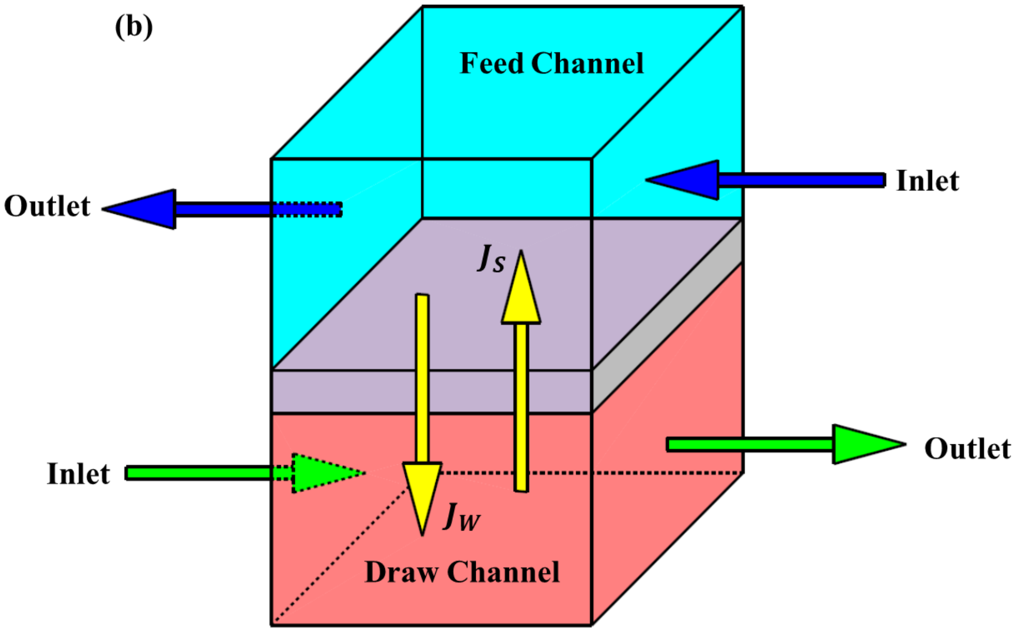


Discrete location of highlighted points:- 1: (i,j); 2: (i,j+1); 3: (i,j-1); 4: (i-1,j); 5: (i+1, j)

Figure 3.7. Discretization of the membrane surface for numerical simulation.

After the calculation of water flux (J_w) and reverse solute flux (J_s), the solution flowrate, velocity, and concentration over the membrane surface at different locations were calculated.





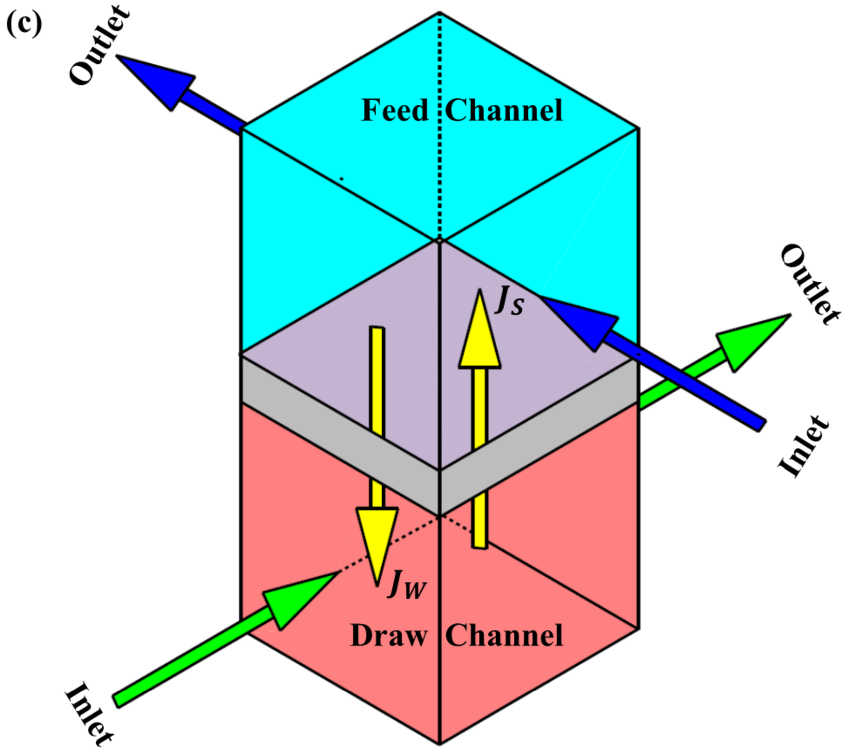


Figure 3.8. Different flow configuration in membrane modules, such as (a) co-current, (b) counter-current and (c) cross-current flow.

This thesis used a numerical method to solve the coupled differential models (equation 3.3 to 3.6) for draw solution and feed solution flowrate and solute flowrate which further gives the solution concentration and velocity. In order to apply the numerical technique, the membrane surface was discretized into very small segments as shown in Figure 3.7. This method employs the effects of water flux and solute flux through any discrete elements on the flowrate, velocity, and concentration of the adjacent elements considering the direction of the flow. Generally, three different flow orientations are found in the membrane modules, such as co-current, counter-current, and cross-current flow orientation as shown in Figure 3.8.

For the co-current flow where the draw solution and feed solution both flow in the length direction ($i = 1\ to\ m$) as shown in Figure 3.8(a), the models for the solution (equation 3.3 and 3.4), and solute (equation 3.5, 3.6) flowrates are solved employing the finite difference method. Draw solution and feed solution flowrates for the co-current flow configuration are expressed by,

$$Q_{DS}(i+1,j) = Q_{DS}(i,j) + J_w(i,j) \times (dl \times dw) \tag{3.11}$$

$$Q_{FS}(i+1,j) = Q_{FS}(i,j) - J_w(i,j) \times (dl \times dw) \tag{3.12}$$

Feed solution and draw solution flow in opposite directions for the counter-current flow configuration which is shown in Figure 3.8(b). Considering this flow configuration, the draw solution is given by,

$$Q_{DS}(i+1,j) = Q_{DS}(i,j) + J_w(i,j) \times (dl \times dw) \tag{3.13}$$

Similarly, the feed solution flowrate is given by,

$$Q_{FS}(i-1,j) = Q_{FS}(i,j) - J_w(i,j) \times (dl \times dw) \tag{3.14}$$

Cross-current flow is another type of flow configuration where the feed solution flows orthogonally or right angle to the draw solution as shown in Figure 3.8(c). Considering that the draw solution flows in the length direction ($i = 1\ to\ m$) and the feed solution flows in the width direction ($j = 1\ to\ m$), the draw solution flowrate model is proposed as,

$$Q_{DS}(i+1,j) = Q_{DS}(i,j) + J_w(i,j) \times (dl \times dw) \tag{3.15}$$

In the same way, the feed solution flowrate is given by,

$$Q_{FS}(i,j+1) = Q_{FS}(i,j) - J_w(i,j) \times (dl \times dw) \tag{3.16}$$

Likewise, the draw solute and feed solute flowrate are calculated for co-current, counter-current, and cross-current flow configurations which provide the solution velocity, concentration, water flux, and reverse solute flux at any location over the membrane surface. Considering the spatial distribution of solution properties and flow behaviors, module scale performance was estimated in terms of average water flux of the module, diluted draw solution production rate, final draw solution concentration, and recovery rate. However, since the module scale performance models are different for different modules, these models are described in the respective chapter of those modules.

3.7 Validation of theoretical models

Pilot scale experiments were conducted to measure the performance of different types of FO membrane modules at the same operating conditions of the theoretical simulations. The experimental results were then compared with the simulated module performance to assess the accuracy of the theoretical models employed in this study. Figure 3.9 shows the FO pilot scale experimental setup. It consists of a pressure vessel that housed the membrane modules, a feed solution pump, and a draw solution pump to supply the feed and draw solution flow through the modules. Seawater was considered as the draw solution for this work. Therefore, 0.6 M NaCl solution was prepared by using tap water to make it similar to the seawater (osmotic pressure of 28 bar at 0.6 M concentration). However, the FO system designed in this study was proposed to treat the MBR effluent (wastewater) by recovering freshwater from wastewater to dilute the seawater osmotically. For this particular study, the feed solution was prepared by using NaCl only to have the osmotic pressure of 0.99 bar (i.e. 0.02 M NaCl) without the presence of any organics considering the very low fouling potential of the secondary effluent.

Figure 3.9. FO pilot scale experimental setup

Due to the osmotic pressure gradient of the feed solution and draw solution water transfers from the feed solution tank of the setup to the draw solution tank. Water flux through the membrane was calculated by measuring the change in mass of the draw solution tank using equation (3.17),

$$J_w = \frac{\Delta m_{DS}}{\rho \times A_m \times t} \tag{3.17}$$

here, Δm_{DS} is the change of draw solution tank's mass, ρ is the density of the draw solution, A_m is the total membrane surface area of the module and t is the time of filtration.

A digital mass balance was used to measure the weight of the draw solution tank. This setup was also devised with some other sensors such as electric conductivity meters, pressure gauges, and flowmeters. Conductivity meters measure the inlet and outlet concentration of the solutions. Similarly, the flowmeters and pressure gauges measure the inlet and outlet flowrates and pressures of the feed and draw solutions. Finally, the datasets of the sensors were transmitted through a data acquisition system and recorded in a PC for further analysis.

3.8 Optimization algorithm development

Optimization of a full scale FO system is one of the main objectives of this thesis. A full scale FO system consists of an array of membrane modules to achieve the design targets such as diluted draw solution production capacity, recovery rate, and final draw solution concentration of the system. Usually, membrane modules are first connected serially and assembled in a pressure vessel to obtain the desired recovery rate and final draw solution concentration. However, if the total production capacity of the plant is not achieved by a single pressure vessel the pressure vessels are connected parallel to each other to meet the desired production capacity of the system. Usually, the previous module scale studies determined the optimum operating conditions based on the maximum recovery rate (Deshmukh et al. 2015; Mondal, Field & Wu 2017). However, to increase the recovery rate of a full scale FO plant the draw solution inlet flowrate to the system and the number of serially connected membrane modules can be increased. Although increasing the draw solution inlet flowrate increases the recovery rate of the system, it also increases the final concentration of the draw solution. On the other hand, the addition of more elements in the pressure vessel does not only increases the recovery rate of the system, but it also reduces the final draw solution concentration. Consequently, it increases the footprint of the system which contributed to the increase in production cost. Therefore, the optimization of the FO system should be based on the maximum recovery rate, minimum final draw solution concentration, and minimum number of elements instead of only considering the maximum recovery rate. Hence, this study proposed an overall performance index (considering the recovery rate, final draw solution concentration, and number of elements) to develop an optimization algorithm which determines the optimum operating conditions of a full scale FO system.

3.9 Software development

Finally, this study developed a FO system analysis (FOSA) software to design and optimize a full scale FO plant using various types of membrane modules. First of all, some algorithm flowcharts were designed to construct the framework of the software which were later coded using MATLAB programming language to develop the frontend and the backend layer of the software. Frontend layer or the graphical user interface (GUI) of the software interacts with the user to receive the input for the theoretical simulation and to display the results. On the other hand, simulations were carried out in the backend layer of the software, which was not visualized by the user. The FOSA software uses the mathematical models and optimization algorithm of the spiral wound, hollow fiber, and plate and frame module to determine the optimum membrane modular configuration and operating conditions of the full scale FO system.

CHAPTER 4

FORWARD OSMOSIS SYSTEM ANALYSIS FOR OPTIMUM DESIGN AND OPERATING CONDITIONS USING SPIRAL WOUND MODULE

4 Forward osmosis system analysis for optimum design and operating conditions using spiral wound module

4.1 Introduction

The sustainability of water and energy resources is endangered by the growing freshwater demand due to the increasing global population and economic development. Wastewater reuse and seawater desalination are two major alternatives to overcome this crisis (Anderson, Scarborough & Watson 2013; Valladares Linares et al. 2014). Currently, forward osmosis is being considered as a promising sustainable solution to this problem. It is a membrane based separation process that is only driven by the osmotic gradient between the highly concentrated draw solution (DS) and the relatively less concentrated feed solution (FS) (Cath, Childress & Elimelech 2006; Field & Wu 2018). In contrast, the conventional membrane desalination system, reverse osmosis (RO) requires to apply hydraulic pressure for the mass transport through the membrane. Therefore, it requires about 3-4 kWh energy to produce 1 m^3 of desalinated water from seawater (Ali et al. 2016; Mazlan, Peshev & Livingston 2016b). However, if the seawater is osmotically diluted by the FO process and used as the feed solution of the RO process it reduces the overall energy consumption of the overall desalination system (Im et al. 2020; Phuntsho et al. 2016; Seo et al. 2019). In addition, the lower operating pressure of FO process resulted in lower irreversible fouling effects (Siddiqui et al. 2018). As such significant research attention has been drawn to improve the performance of FO processes.

FO system performance primarily depends on the type of membrane module. Although there are three types of FO module (such as spiral wound, hollow fiber and plate and frame module) are commercially available, spiral wound module is the most widely available module due to its high packing density commercially (Attarde, Jain & Gupta 2016; Gu et al. 2011b). Numerous studies attempted to optimize the performance of the FO processes using spiral wound FO module. However, most of these studies experimentally measured the performance using a small membrane coupon using a lab scale experimental setup (Akther et al. 2020; Lee & Ghaffour 2019; Tiraferri et al. 2013). Outcomes of these studies are although pivotal to understand the fundamental mechanisms of the FO processes, these findings cannot be applied to estimate the performance of a full scale system as the osmotic gradient across the membrane changes due to the permeation of water from the feed side to the draw side of the full scale membrane module. Few studies experimentally investigated the performance of the pilot scale FO system (Im et al. 2016; Kim, Blandin, et al. 2017; Kim et al. 2015; Phuntsho et al. 2016). Since these experimental studies were conducted for a small range of operating conditions and a specific application, these studies cannot be applied for a wider range of operating conditions and other applications. Therefore, to design and optimize a full scale FO system, theoretical analysis of FO system is very important.

Recently, few studies theoretically optimized the performance of the full scale FO system. For example, the effects of different intrinsic properties of membrane on module scale FO process performance employing the co-current and counter-current configurations were simulated (Deshmukh et al. 2015). In another study, the optimum membrane area was determined by using the log mean temperature difference method at various operating conditions for co-current and counter-current flow configurations (Mondal, Field & Wu 2017). Phuntsho et al. simulated the performance of a full scale fertilizer drawn FO-NF hybrid system (Phuntsho et al. 2017). Benchik et al. numerically simulated and compared the performances of FO and AFO (Assisted FO) mass exchangers (Banchik et al. 2016).

However, the performance of FO systems was optimized by maximizing the recovery rate only. It was found that the higher recovery rate of the FO system can be achieved by lowering the dilution rate or increasing the final DS concentration (Mondal, Field & Wu 2017; Phuntsho et al. 2017; Phuntsho et al. 2016). But the optimum operating conditions should be determined based on both recovery rate and final draw solution concentration. Moreover, these studies considered arbitrary dimensions of the membrane modules which does not show a similar performance with the commercial modules. In addition, the solution flow was considered through the length direction of the membrane only. Although these configurations of flow are valid for plate and frame module, the solutions flow in both length and width directions of the membrane for a spiral wound module. In addition, the models and assumptions of these studies were not experimentally validated.

This study is aimed to design and optimize a full scale FO system employing a commercial spiral wound module. It employed the fluid mass balance equations coupled with the solution diffusion model to develop the mathematical models. The finite difference method was applied by considering the actual dimensions and flow configurations of a commercial TFC spiral wound module to numerically simulate the full scale FO system performance using the developed models. Analytically found performance was then compared with the experimental result to validate the models used for this study. Finally, a novel overall performance index was developed to optimize the design and operating conditions considering both recovery rate and concentration requirement.

4.2 Operating conditions and design parameter of FO system

Performance of a FO plant is defined in terms of production capacity, recovery rate and final DS concentration (Kim et al. 2018; Mondal, Field & Wu 2017; Phuntsho et al. 2017). Generally, production capacity of the system is measured as the volumetric production of diluted DS per day. Recovery rate is defined as the percentage ratio of water permeation rate to the inlet FS flowrate. An efficient FO system exhibits low final DS concentration along with high production capacity and recovery rate at a certain FS flowrate. However, better performance can be achieved by varying the operating conditions and design parameters. DS and FS inlet flowrates that are generally varied during operation for any applications are the two major operating conditions (Khayet et al. 2016; Kim et al. 2018; Xu et al. 2010). The other two most significant parameters are DS and FS inlet concentrations. The variations of flowrates also influence the concentrations and vice versa. Change in mass transfer coefficient due to the crossflow velocities defines the external concentration polarization that affects the concentration at solution-membrane interface (Phuntsho et al. 2014; Wang, Zhang, et al. 2016). Apart from these operating conditions, some design parameters such as membrane element type, the number of elements in pressure vessels, spacer thickness, flow configurations, etc. also influence the performance (Field et al. 2017; Jung et al. 2011; Kim, Blandin, et al. 2017; Kim et al. 2018). Increasing the number of elements causes increased water permeation, but due to the osmotic dilution the average water flux reduces. Consequently, the production cost of the system increases. On the other hand, the change of spacer thickness affects the solution velocity or external concentration polarization. Moreover, the membrane surface area and direction of flow change which also affects the performance. Therefore, to design an efficient system this study determined the optimum DS inlet flowrate and number of elements in a pressure vessel for a targeted recovery rate and final concentration using a specific type of element.

4.3 Theoretical analysis

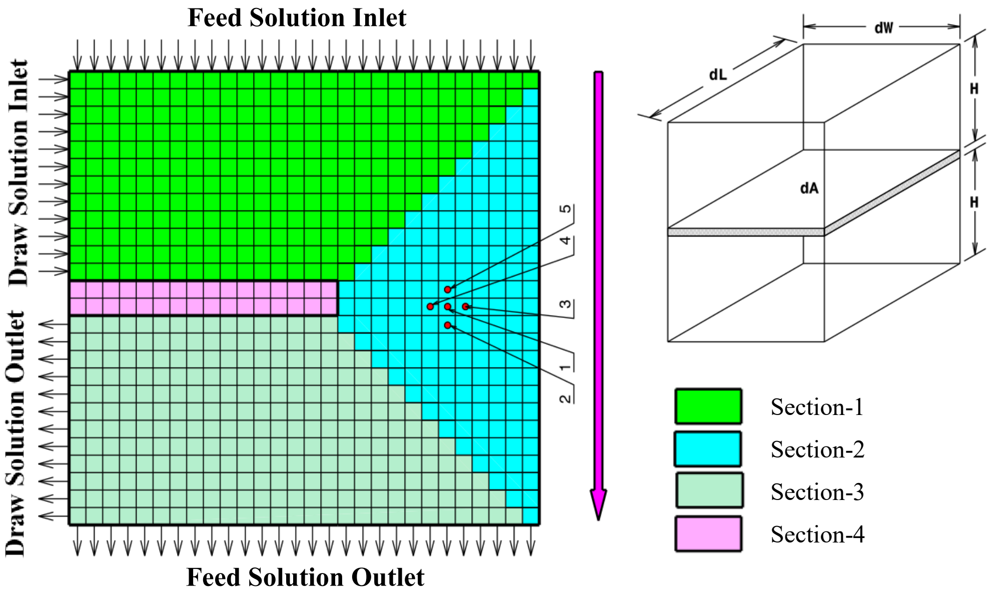
A complete theoretical framework was developed to determine the optimum operating conditions and design parameters of a FO system using spiral wound module. Solution diffusion model was incorporated with the mass balance equations to develop the mathematical models of a full-scale system. Numerical solution of these models can provide the local distributions of flux over the membrane surface, which in turn can estimate the system performance.

4.3.1 Mathematical models

The analytical framework begins with the calculation of thermodynamic properties of feed and draw solutions at different concentrations. Osmotic pressure of the solution was determined by using the modified Van't Hoff equation (Gu et al. 2011b; Phuntsho et al. 2014; Xue et al. 2016). Considering the membrane characteristics and concentration of the solution, the local water flux and reverse solute flux are determined by equation 3.1 and 3.2 presented in Chapter 3,

4.3.2 Numerical simulation

Total water permeation through a large membrane area cannot be calculated by applying the mathematical models for water flux and RSF considering the whole membrane surface is exposed to the same concentration and velocity. During the operation because of water flux and RSF, DS is diluted and FS is concentrated along the path of flow, which eventually reduces the solution concentration gradient.



Discrete location of highlighted points:- 1: (i,j); 2: (i+1,j); 3: (i,j+1); 4: (i,j-1); 5: (i-1, j)

Figure 4.1. Discretization of membrane area for iterative solution.

Therefore, the membrane area was discretised to sufficiently smaller areas ($\boldsymbol{l} \times \boldsymbol{w}$) as shown in Figure 4.1 and the models were applied iteratively to the discrete areas considering the effects of water flux and RSF of one discrete area on the next according to the directions of the flow. Mass balance equations were combined with the models to determine the local velocities and concentrations. It was assumed that the velocities and concentrations changed only in the dominant direction of the flow. Moreover, since the channels heights are too small and filled up with spacer, velocity and concentration boundary layers in the height direction were considered negligible. Employing the mass balance equation, the change in FS flowrate in the direction of their flow was expressed and solved numerically by the finite difference method to determine the local feed flowrate on the membrane surface as follows:

$$Q_F(i+1,j) = Q_F(i,j) + [-J_w(i,j) + J_s(i,j)](l \times w) \tag{4.1}$$

where, i and j are the indexes, which represent the location of the discrete area. $J_w(i,j)$ and $J_s(i,j)$ represent the volume of water and salt diffuses through the discrete area of the membrane.

Considering the dominant direction of flow, the DS flow path was divided into three sections as shown in Figure 4.1. At section-1, DS flowed in the forward direction and in a cross-current configuration with the FS flow path. At section-2, DS and FS flow paths are co-current. Finally at section-3, DS flowed in the reverse direction as compared to section-1, but the FS and DS flow path remain in cross-current mode. Thus, the DS flowrate for different sections is described as,

For section 1:

$$Q_D(i,j+1) = Q_D(i,j) + [J_w(i,j) - J_s(i,j)](l \times w) \tag{4.2}$$

For section 2:

$$Q_D(i+1,j) = Q_D(i,j) + [J_w(i,j) - J_s(i,j)](l \times w) \tag{4.3}$$

For section 3:

$$Q_D(i,j-1) = Q_D(i,j) + [J_w(i,j) - J_s(i,j)](l \times w) \tag{4.4}$$

Local velocity of the solutions can be estimated by equations (4.7) and (4.8), when the flowrate and the physical dimensions of the channel and spacer are known (Sim et al. 2015).

$$U_F(i,j) = \frac{Q_F(i,j)}{\varepsilon_{sp} h_{sp} w} \tag{4.5}$$

$$U_D(i,j) = \frac{Q_D(i,j)}{\varepsilon_{sp} h_{sp} l} \quad (4.6)$$

where, ε_{sp} is the voidage of the channel, h_{sp} is the height of the spacer. Feed spacers are generally diamond shaped net like structures with cylindrical filament. On the other hand, DS side spacers consist of a feed spacer like structure sandwiched between two very thin finely porous net type structures which work as permeate carrier.

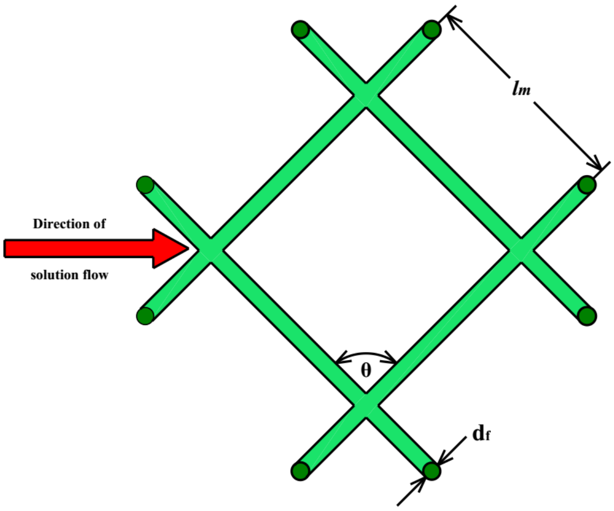


Figure 4.2. Physical dimensions of the spacer.

Since, the permeate carriers are very thin as compared to the feed spacer, the dimensions of draw side spacers were assumed similar to the feed side spacers for the calculation of channel voidage. However, the voidage can be estimated as (Da Costa, Fane & Wiley 1994),

$$\varepsilon_{sp} = 1 - \frac{\pi d_f^2}{2 l_m h_{sp} sin\theta} \quad (4.7)$$

where, d_f is the diameter of spacer filaments, l_m is the length of mesh, and θ is the angle as shown in Figure 4.2.

The amounts of solute in solution flows over the discrete area $(l \times w)$ are expressed as,

$$Q_F^{Slt}(i+1,j) = Q_F(i,j)C_F(i,j) + J_s(i,j)(l \times w) \tag{4.8}$$

For section 1:

$$Q_D^{Slt}(i,j+1) = Q_D(i,j)C_F(i,j) - J_s(i,j)(l \times w) \tag{4.9}$$

For section 2:

$$Q_D^{Slt}(i+1,j) = Q_D(i,j)C_F(i,j) - J_s(i,j)(l \times w) \tag{4.10}$$

For section 3:

$$Q_D^{Slt}(i,j-1) = Q_D(i,j)C_F(i,j) - J_s(i,j)(l \times w) \tag{4.11}$$

Local bulk concentration of solutions at different locations of the membrane channel was determined by calculating the amount of solute available in the unit amount of solution within the discrete area as shown in equations (4.14) and (4.15),

$$C_F^b(i,j) = \frac{Q_F^{Slt}(i,j)}{Q_F(i,j)} \tag{4.12}$$

$$C_D^b(i,j) = \frac{Q_D^{Slt}(i,j)}{Q_D(i,j)} \tag{4.13}$$

Since FS flows at right angle to the DS and the DS changes its direction of flow, the local water flux and RSF vary in both direction. Moreover, the other variables such as flowrate, velocity and concentration are dependent on the water flux and RSF. As a result, although these are one dimensional finite difference models, but all the variables are function of both the length and width of the membrane sheet. Therefore, the total water permeation from a module is estimated by a double integration of local water flux which is a function of both length and width as follows,

$$Q_w^{elm}(E) = N_{sheet} \int_{i=1}^{m} \int_{j=1}^{n} J_w(i,j)dldw \tag{4.14}$$

here, N_{sheet} is the number of membrane sheets available in one element, m $(m = Length/l)$ and n $(n = Width/w)$ represent the number of discrete areas in length and width direction. Average water flux of a membrane element is the ratio of total permeation to the membrane surface area of a module, which is given by,

$$J_w^{Avg}(E) = \frac{Q_w^{elm}(E)}{A_{elm}} \tag{4.15}$$

where, A_{elm} is the total membrane surface area. The amount of diluted DS produced after any element is expressed by summing all the local DS flowrates at the outlet of the membrane sheet, where $j = 1$,

$$Q_D^{Out}(E) = N_{sheet} \sum_{i=1}^{(m-p)/2} Q_D(i, 1) \tag{4.16}$$

here, p is the number of discrete areas along the width direction of the central glue line of a membrane sheet. Therefore, p was calculated as, p = width of glue line/l. Similarly, the concentrated FS outlet flowrate of the element is

$$Q_F^{Out}(E) = N_{sheet} \sum_{j=1}^{n} Q_F(m,j) \tag{4.17}$$

Final DS concentration after any element is given by,

$$C_D^{Out}(E) = \frac{N_{sheet} \sum_{i=1}^{(m-p)/2} Q_D^{Slt}(i,1)}{Q_D^{Out}(E)} \tag{4.18}$$

Outlet concentration of the FS from an element,

$$C_F^{Out}(E) = \frac{N_{sheet} \sum_{j=1}^{n} Q_F^{Slt}(m,j)}{Q_F^{Out}(E)} \tag{4.19}$$

Finally, the recovery rate after any element is expressed as the percentage ratio of total fresh water permeation to the FS inlet flowrate,

$$R(E) = \left[\frac{Q_D^{out}(E) - Q_D^{in}}{Q_F^{in}}\right] \times 100 \tag{4.20}$$

4.3.3 Optimization of design and operating conditions

This section introduces a novel optimization algorithm that can determine the optimum DS flowrate and the number of elements required to obtain a targeted final DS concentration and recovery rate at a certain FS flowrate for a specific FO application. Generally, the optimum operating condition is found based on either the maximum recovery rate or minimum final DS concentration. Previous studies however showed that a higher recovery rate can be obtained by applying higher DS inlet flowrate or concentration which results in a higher final DS concentration (Mondal, Field & Wu 2017; Phuntsho et al. 2017). So, the design or operating conditions that provide maximum recovery rate or minimum final concentration are not always the optimum conditions.

Therefore, a new performance parameter, overall performance rating (OPR) was proposed in this study, which combined the recovery rate and final DS concentration with the number of membrane elements. OPR can be defined as the percentage of the overall design target achieved per element. It was formulated by the multiplication of recovery fraction and concentration fraction. However, the recovery fraction is defined as the ratio of actual recovery rate achieved to the desired recovery rate and is given by,

$$RF(D,E) = \frac{R(D,E)}{R_D} \tag{4.21}$$

here, $R(D,E)$ and $RF(D,E)$ are the obtained recovery rate and recovery fraction respectively at a certain DS flowrate and the number of elements, whereas R_D is the targeted recovery rate. On the other hand, the concentration fraction is a ratio of targeted final concentration to the actual final concentration, which is given by,

$$CF(D,E) = \frac{C_D^F}{C_D(D,E)} \tag{4.22}$$

where, C_D^F is the desired final DS concentration, $C_D(D,E)$ and $CF(D,E)$ are the achieved final concentration and concentration fraction respectively. In order to prevent the overdesign, the maximum values of recovery fraction and concentration fraction were considered one, which means if $R(D,E) \geq R_D$, then $RF(D,E) = 1$. On the other hand, if $C_D(D,E) \leq C_D^F$, then $CF(D,E) = 1$. Combining the recovery fraction, the concentration fraction and the number of elements, OPR is expressed by,

$$OPR(D,E) = \left[\frac{RF(D,E) \times CF(D,E)}{N_E}\right] \times 100 \tag{4.23}$$

here, $OPR(D, E)$ is the value of overall performance rating for a certain DS flowrate and number of elements, whereas N_E is the number of elements required to obtain the recovery fraction and concentration fraction. The proposed optimization method was implemented by calculating OPR for several sets of operating conditions and selecting the operating conditions which provide the maximum value of OPR.

The main drawback of this method is that the OPR value of one set of operating conditions (where the recovery fraction is very high and the concentration fraction is very low) is similar with the OPR values of other sets of operating conditions (where the recovery fractions are very low and the concentration fractions are very high). As a result, this method sometimes finds the optimum operating conditions when the recovery rate is very high but the final DS concentration is higher than the targeted value. On the other hand, it may also find the optimum operating conditions where the final concentration is very low, but the recovery rate is lower than the targeted recovery rate, which is unacceptable. In order to overcome this limitation, search domain of operating conditions was constrained. Instead of searching the optimum conditions within the whole range, it only searches when both the actual recovery rate and the final concentration are in an acceptable proximity to the targeted recovery rate and concentration. This constrain was applied by $OPR(D, E) = 0$, when $RF(D, E) \leq (1 - Fl)$ or $CF(D, E) \leq (1 - Fl)$. Here, Fl represents the acceptable deviation from the target and is termed as flexibility of design.

The proposed optimization algorithm is described in Figure 4.3. The optimization process was started with the configuration of the design targets, such as the final concentration (C_D^F) and recovery rate (R_D) as well as the flexibility of design (Fl). Then the recovery rate and FS flowrate (Q_F^{in}) were configured as one target here.

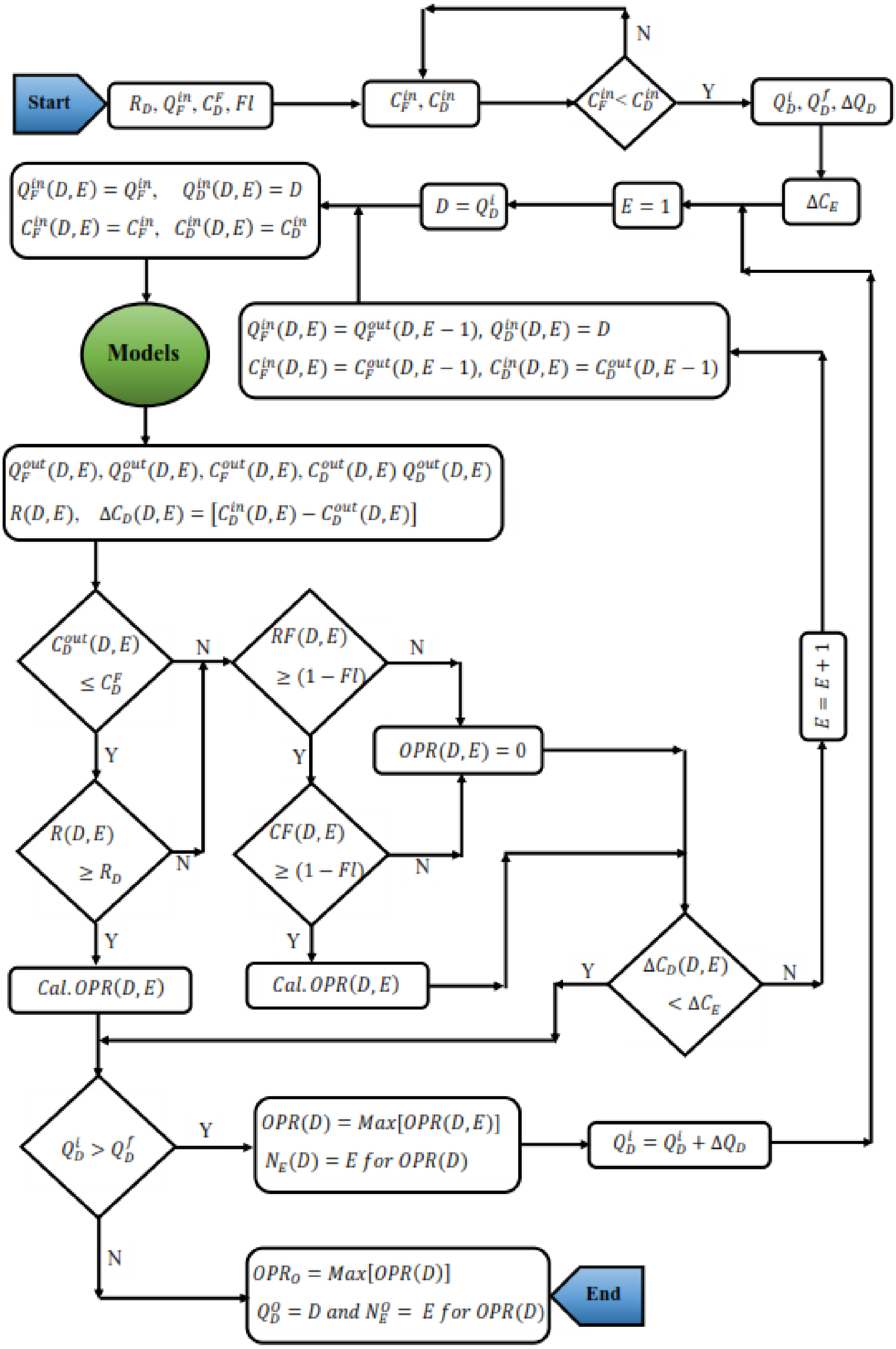


Figure 4.3. Optimization algorithm for FO system design and operation.

The FS and DS inlet concentration (C_F^{in} and C_D^{in}) were initialized. These operating conditions generally remain unchanged for a specific application. For example, FS and DS inlet concentrations are about 0.02 M and 0.6 M respectively for seawater dilution using wastewater. In addition to this minimum dilution (ΔC_E) required to add more elements in a pressure vessel was configured. It means that if reduction in DS concentration due to the addition of an element in a pressure vessel is less than ΔC_E, no more element is added. Finally the DS inlet flowrate was incremented from an initial value (Q_D^i) to a final value (Q_D^f) by increasing the flowrate a small amount (ΔQ_D) at each iteration. For every DS flowrate, the number of elements per pressure vessel was increased and OPR of the FO system was investigated and compared to determine the optimum DS flowrate and number of elements per pressure vessel. The number of element (E) was initialized with 1 and then the inlet operating conditions for the first element were set for the simulation using the mathematical models presented.

The simulation results provided the FS and DS outlet flowrate (Q_F^{out} and Q_D^{out}) and concentration (C_F^{out} and C_D^{out}). Recovery rate and osmotic dilution by the element was calculated. Dilution by the addition of this element was calculated by,

$$\Delta C_D(D,E) = \left[C_D^{in}(D,E) - C_D^{out}(D,E)\right] \tag{4.24}$$

The outlet DS concentration and recovery rate were then compared with the targeted final concentration and recovery target. If the target was not achieved (i.e. either the outlet concentration was higher than the final concentration target or the recovery rate was less than the recovery target), the recovery fraction and concentration fraction were calculated and checked whether these fractions were within the acceptable range. If any of these fractions were out of the acceptable limit OPR was set as 0, otherwise OPR was calculated using equation 4.25. Then the dilution by the addition of this element ($\Delta C_D(D, E)$) was compared with the minimum required dilution for adding more element (ΔC_E). If the dilution was higher than the required minimum dilution, the second element was added and the outlet conditions of the first element were considered as the inlet conditions for the second element. Then the same processes were repeated. Due to the addition of another element, the recovery rate increased and the final concentration decreased. However, if the concentration or recovery rate still did not reach the target, the recovery and concentration fractions were calculated again. Otherwise, if these were in the acceptable range, OPR was calculated and the dilution was compared. If it was less than the minimum required dilution, the current DS inlet flowrate was checked whether it was the last value of the DS flowrate range. If the targets were reached, OPR was calculated and the DS flowrate was compared with the last value of the range. If the current flowrate was less than the last flowrate of the range, the highest OPR for this flowrate, $OPR(D)$ was found.

The optimum number of elements for this flowrate, $N_E(D)$ was the corresponding number of elements for the $OPR(D)$. Then the DS inlet flowrate was incremented and the same process was continued. In contrast, if the current DS flowrate was the last value of the range, the $OPR(D)$ values for all flowrates were compared and the maximum value (OPR_O) was selected. DS flowrate (Q_D^O) and the number of elements (N_E^O) corresponding to the maximum OPR (OPR_O) were the optimum operating conditions and design parameters for the desired final concentration and recovery rate.

4.4 Results and discussion

4.4.1 Validation of experimental and simulated results

In order to validate the outcomes of the software, the simulation results of this software were compared with the findings of our previous experimental study (Kim et al. 2018). A commercial 8040 spiral wound thin film composite FO membrane element was used for pilot-scale FO experiments. Sodium chloride (0.6 M concentration) solution was used as DS which is similar to that of typical seawater. The application of FO systems in this study was dilution of seawater using wastewater effluent of membrane bioreactor (MBR). Wastewater has an osmotic pressure of around 0.99 bar. Therefore, FS was prepared by using NaCl only to have the same osmotic pressure (i.e. 0.02 M NaCl) without the presence of other components. The FS and DS outlet of the first element was used as the inlet of the second element and so on. These experiments were conducted up to 4 elements. The inlet and outlet flowrate and concentration of the modules were recorded using electromagnetic flowmeters and conductivity meters connected to a PC through a data acquisition system. The inlet flowrates of FS and DS for the first element were 40 L/min and 4.7 L/min respectively. The same membrane element properties and operating conditions (as shown Table A1.1 of the appendix) were considered for this study to compare the analytical findings with the experimental results.

Figure 4.4 compares the analytically estimated solution inlet concentrations and flowrates for increasing number of elements in a pressure vessel with the experimentally measured flowrates and concentrations. It can be seen from the figure that the analytical results from the system analysis software are in a very good agreement with the experimental results.

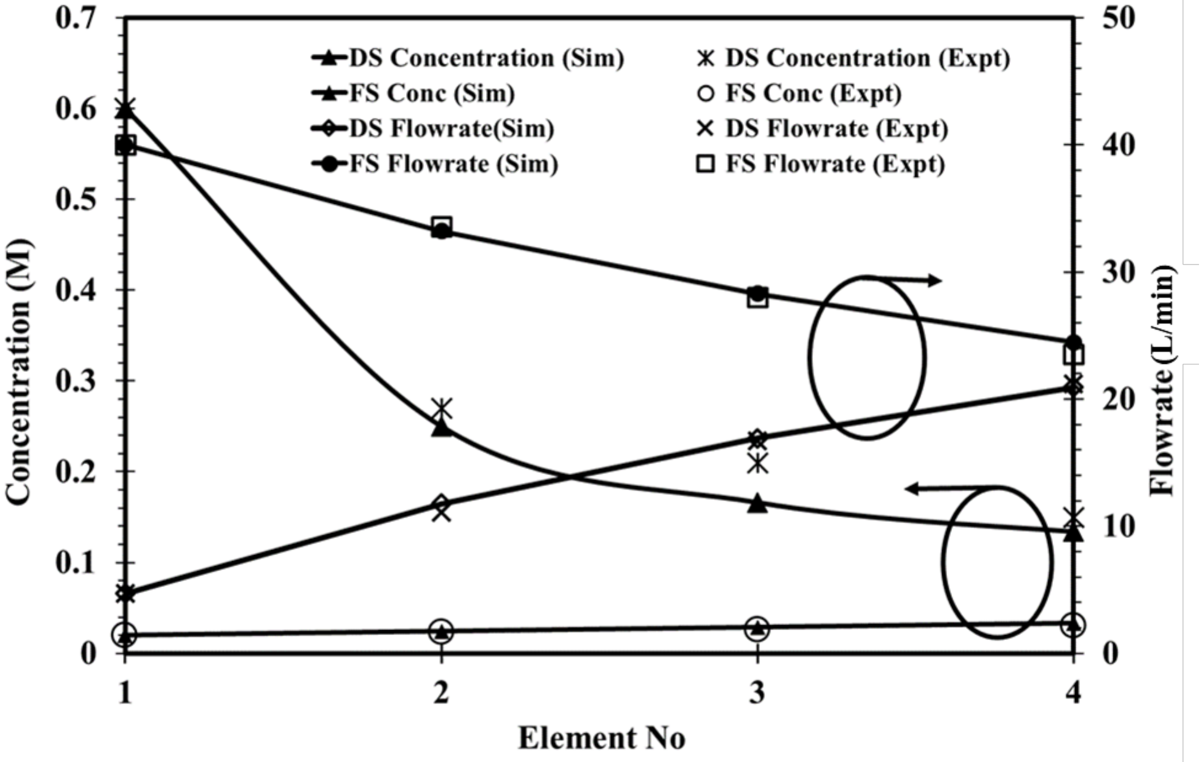


Figure 4.4. Comparison of the concentrations along the element.

It shows that due to the permeation of fresh water from FS to DS, the flowrate of DS increased, whereas the FS flowrate decreased. In contrast, the permeation of water caused dilution of DS but increased the concentration of FS. According to this study, inlet DS flowrate increased from 4.7 L/min to 21 L/min for the fourth element in a pressure vessel, with only 2.8% average deviation from the experimental results. Similarly the FS flowrate decreased from 40 L/min to 20.5 L/min for the same number of element. Average deviation from the experimental results was only 2%. On the other hand DS concentration decreased from 0.6 M to 0.13 M and FS concentration increased from 0.02 M to 0.03 M. The average difference of theoretical FS concentration with the experimental study was found to be 5.25%. As compared to the other parameters, simulated DS concentration showed higher deviation (about 12%) from the experimental study.

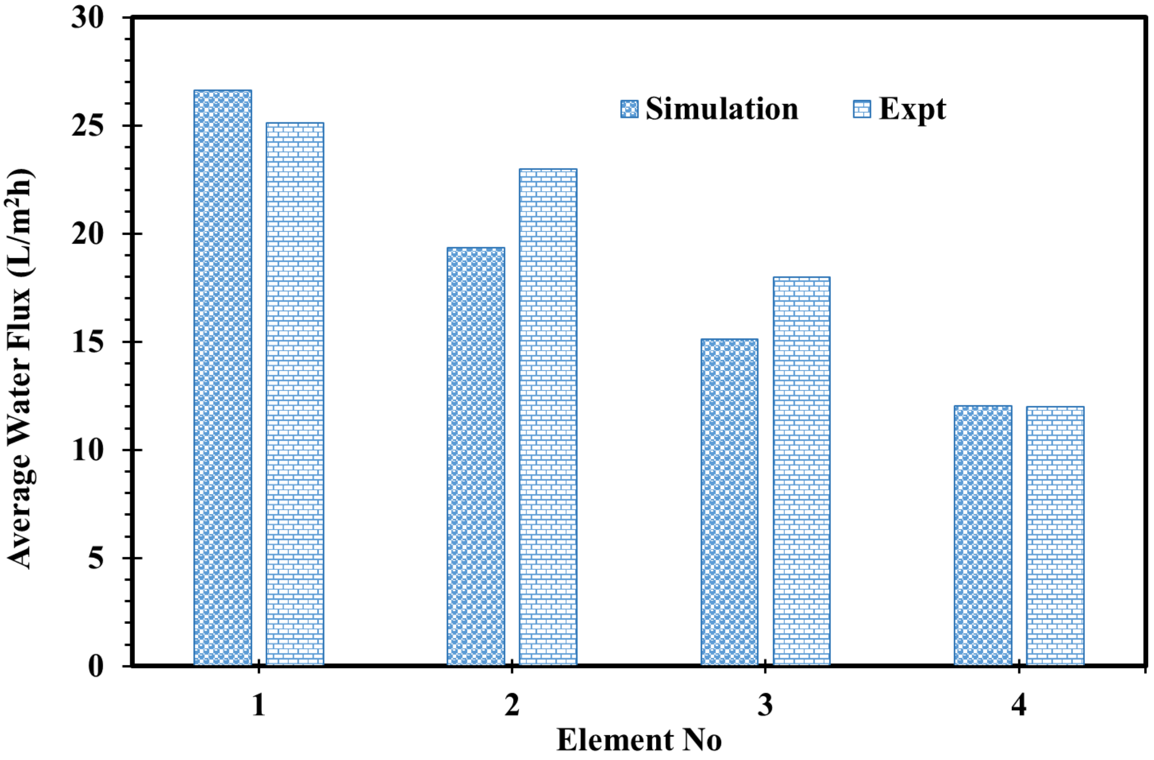


Figure 4.5. Comparison of the average water flux from the elements.

Table 4.1 Calculation of optimum performance rating (OPR).

No. of Element	**1**	**2**	**3**	**4**	**5**	**6**
Recovery Rate (%)	17.68	30.53	40.57	48.56	54.94	60.02
RF	0	0	0	0.97	1	1
Final Concentration (M)	0.25	0.17	0.13	0.12	0.10	0.097
CF	0	0	0	1	1	1
OPR	0	0	0	**24.28**	20	16.67

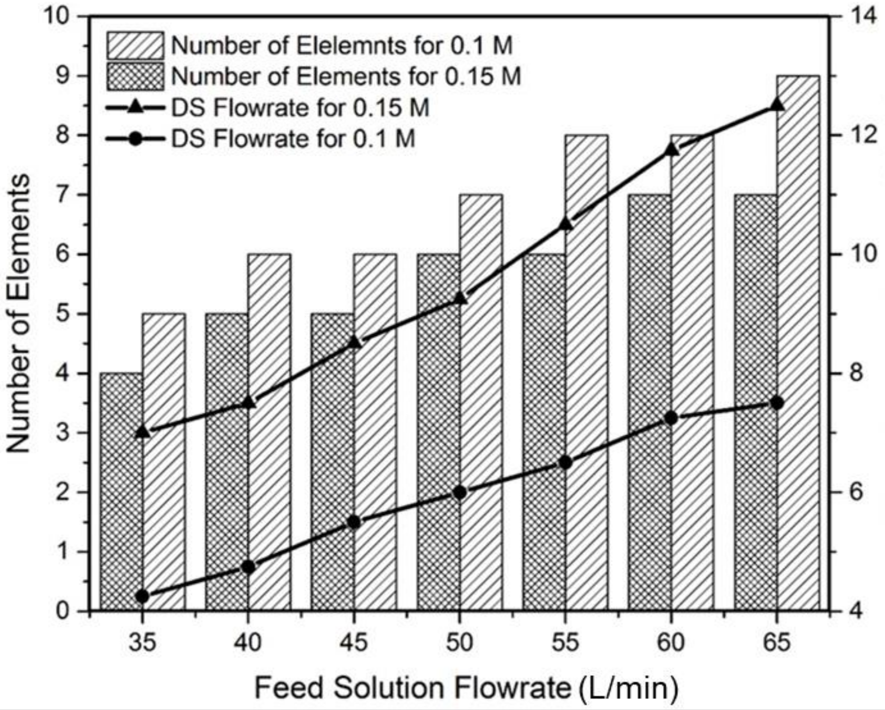
However the reason for this deviation was an experimental error in the measurement of inlet DS concentration for the third element, because although the inlet concentrations for other elements matched with the experimental value satisfactorily, only the inlet DS concentration of the third elements deviated unexpectedly. Figure 4.5 further demonstrates the average water flux from the elements. The average water flux for the first element was 26.5 L/m^{2}h, which reduced to 12 L/m^{2}h for the fourth element due to the reduction in concentration gradient along the solution flow path. It can be seen from the figure that the experimental results displayed a similar decreasing trend, although the water flux for the second and third element indicated a little higher value. This difference in analytical and experimental results was because of the additional pressure applied unintentionally due to the operation of pumps. As a result of this additional pressure, slightly higher water flux was noticed for the experimental investigation as compared to this theoretical system analysis. However, the average deviation from the experimental results for water flux was $\approx$ 7%. In addition to this for the same operating conditions with a design objective of 50% recovery rate and 0.15 M final DS concentration, the optimum number of elements in the pressure vessel was calculated using the proposed algorithm in Section 4.4.3, when the flexibility in design was 5%.

Table 4.1 explains the optimization algorithm. The recovery rate and final concentrations were simulated and listed in the table. Since the recovery rates for the first, second and third element were >5% less than the desired recovery rate (i.e. out of the OPR search domain), RF was set as zero. On the other hand, as the recovery rate for the fifth and sixth element was higher than the targeted recovery rate, RF was set to its maximum value 1. Similarly, the value of CF was calculated and furnished in Table 4.1. Considering the recovery fraction, concentration fraction and number of elements, OPR was calculated and listed in the table. It can be seen that for the discussed operating conditions and design objectives, the maximum of OPR (24.28% objective/element) was achieved when 4 elements were used in a pressure vessel. In comparison, the experimental study also shows that the optimum number of elements in a pressure vessel was 4 (Kim et al. 2018). The above discussion shows that the findings of the developed FO system analysis software satisfied the experimental results with a very close tolerance of deviation, which is about 5% in average. Thus, the software can be employed to design and optimize the full-scale FO system for different applications and operating conditions.

4.4.2 Parametric optimization

Figure 4.6 shows the optimum DS flowrate and the number of elements in a pressure vessel for different design objectives, i.e. for different desired recovery rates at various FS flowrates and final DS concentrations. It can be seen from the figure that the lower the final DS concentration, the lower the optimum inlet DS flowrate to the system. According to Figure 4.6(a) for 0.15 M final DS concentration and 60% recovery rate, when the inlet FS flowrate increased from 35 L/min to 65 L/min, the optimum DS inlet flowrate increased from 7 L/min to 12.2 L/min. On the other hand, if the required final concentration was 0.1 M for the same recovery rate and FS flowrate range, the optimum DS flowrate was lower. It varied from 1.25 L/min to 7.75 L/min. As the higher inlet DS flowrate means that the greater amounts of solute were present in the solution, it resisted the dilution of the solution. Therefore, for a lower final concentration and the same inlet DS concentration, low DS flowrate was used. But at the less DS concentration, the recovery rate reduced. Therefore, for a same recovery rate when the final concentration was lower, more elements were required. Figure 4.6(a) also shows that for 0.15 M final concentration and 60% recovery rate at 35 L/min and 65 L/min FS flowrate, 4 and 6 elements were required in a pressure vessel respectively, whereas for the same design objectives the required number of elements were 5 and 9, when the final concentration were 0.1 M. In addition to this at 60% recovery rate with 35 to 65 L/min inlet FS flowrate, 0.05 M final DS concentration cannot be achieved. Similarly, Figure 4.6(b) and Figure 4.6(c) show the optimum operating conditions for 50% and 40% recovery rate. These figures also show the similar trend of DS flowrate and the number of elements. It can be seen from the figures that for the high final concentration (0.15 M) although the overall trend of the optimum DS flowrate was increasing when the FS flowrate increased, but for few other FS flowrates, the DS flowrate decreased at a regular interval. Since more permeation was required for the same recovery rate at higher FS flowrate, the proposed optimization algorithm increased the number of elements and varied the DS flowrate within a range to find the optimum flowrate. Consequently, at a certain interval the optimum DS flowrate decreased a little bit when the number of element increased, but increased again with the increase in FS flowrate.

(a)



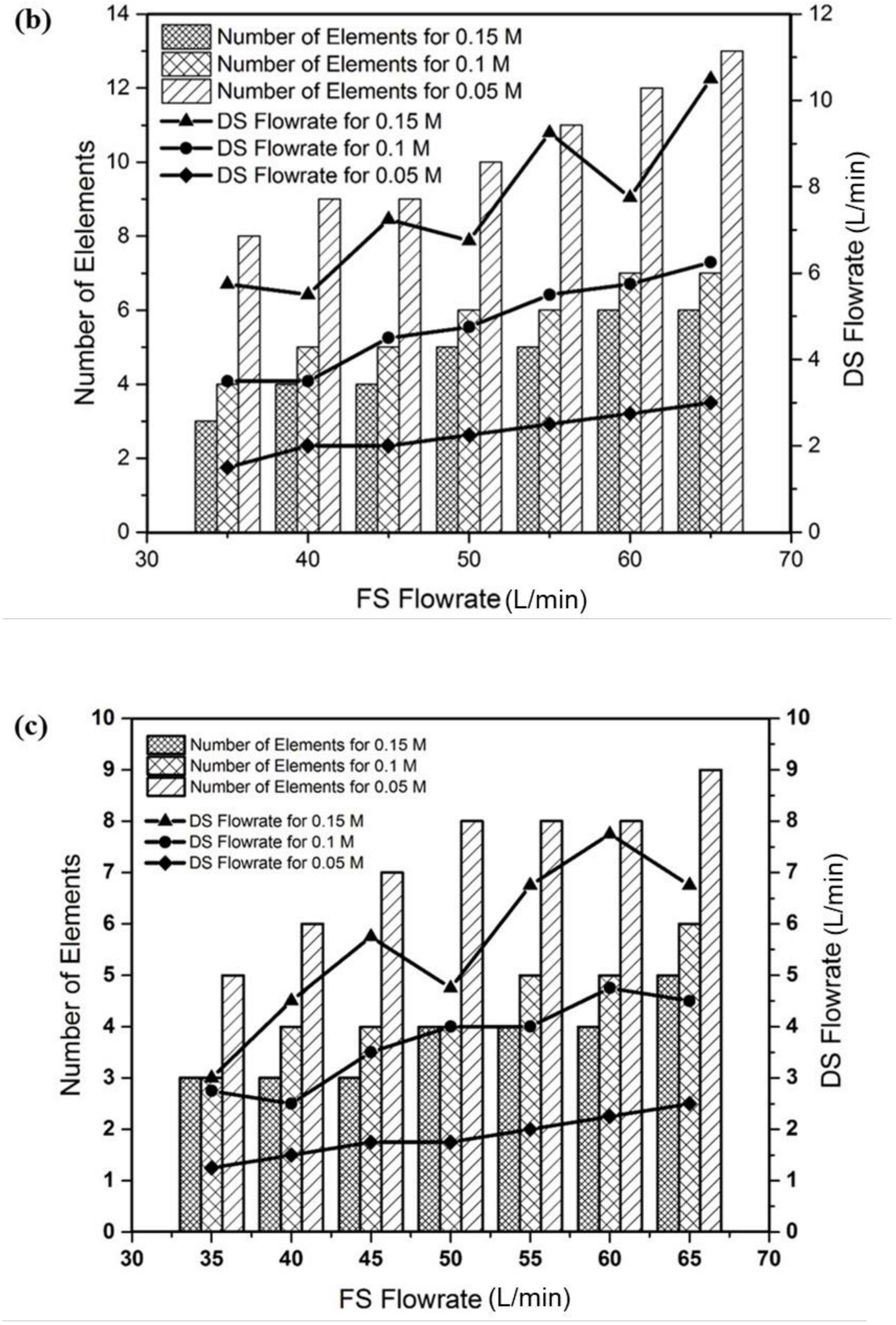


Figure 4.6. Optimum number of elements in a pressure vessel and DS flowrate for (a)

60%, (b) 50% and (c) 40% recovery rate at different FS flowrate and final DS concentration.

However, for the low final concentration, the optimum DS flowrate varied within a small range, therefore the reduction in the optimum DS flowrate was not observed for the 0.1 M and 0.05 M final concentrations.

Finally, average flowrate ratios (i.e. the average values of all ratios of FS flowrate to the DS flowrate) for each recovery rate and final concentration were calculated and compared in Figure 4.7. It shows that for a same DS final concentration the average flow ratio decreased when the recovery rate increased. In comparison, for a same recovery rate the flow ratio decreased, if the final concentration decreased. It can be seen that average flow ratios for 40%, 50% and 60% recovery rate at 0.15 M final DS concentration were 9, 7 and 5 respectively. For the same recovery rates at 0.1 M concentration, FS inlet flowrate was 14, 10 and 8 times higher than the DS flowrate.

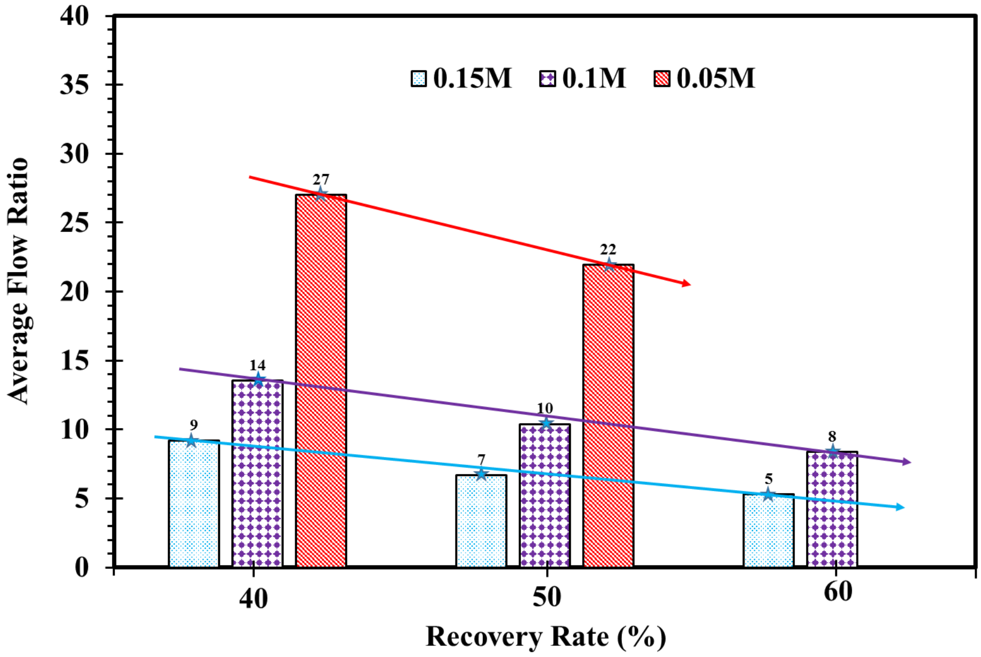


Figure 4.7. Average flowrate ratios for different recovery rates.

Finally at 0.05 M final concentration for 40% and 50% recovery rate the average flow ratios were 27 and 22. Since 60% recovery rate was not obtained at 0.05 M final concentration, Figure 4.7 does not show the average flow ratio for this recovery rate and final concentration.

4.5 Conclusions

The current work modeled and simulated the performance of the FO system for various operating conditions and design parameters. A novel overall performance parameter was presented to estimate the performance of the system considering both the recovery rate and the final DS concentration. This overall performance parameter was used to compare performance at various operating conditions. Operating conditions corresponding to the maximum value of this parameter were the optimum conditions. Accordingly, the main findings of this analytical work are listed below:

- An average ≈ 5% difference between the theoretical and experimental findings validate the models and assumptions used for this study.
- Full-scale FO system design parameters by the proposed system analysis software for 100000 m^3 of diluted DS production capacity per day at 0.15 M final concentration are as follows when the recovery rate is 50%, FS and DS inlet flowrates are 40 L/min and 4.7 L/min:

  Number of elements in a pressure vessel: 4

  Total number of pressure vessel: 2879

  Total number of elements: 11516

CHAPTER 5

FORWARD OSMOSIS SYSTEM DESIGN AND SIMULATION FOR REAL SCALE APPLICATIONS USING A HOLLOW FIBER MEMBRANE MODULE

5 Forward osmosis system design and simulation for real scale applications using a hollow fiber membrane module

5.1 Introduction

Rapidly growing global population and economic development have intensified the water scarcity problem to a new level. Reusing wastewater or desalinating seawater are two of the most sustainable approaches to overcoming this challenge (Choi et al. 2017; Seo et al. 2019; Wang et al. 2018). Superior product quality and less chemical requirement established the membrane filtration processes as an effective method to produce freshwater from these unconventional sources of water. However, the high energy consumption to apply hydraulic pressure and the fouling propensity limited the efficiency of these processes to a great extent (Akther et al. 2020; Chen et al. 2014; Lee et al. 2019). In contrast, FO processes have emerged as an alternative membrane filtration method. It uses the osmotic pressure difference of two different solutions (feed and draw solution) separated by a semipermeable membrane as a driving force for mass transport through the membrane (Ali et al. 2018; Cath, Childress & Elimelech 2006). Moreover, the absence of hydraulic pressure in the system makes it less vulnerable to the fouling effects. As such this process is efficient for its standalone or hybrid applications where the regeneration of DS is not required (Chekli et al. 2016; Im et al. 2020). Therefore FO has obtained significant attention for its widespread application in water filtration.

Numerous studies aimed to optimize the FO process using small membrane in laboratory-scale setup (Wang, Järvelä, et al. 2016; Zhao et al. 2016; Zhao et al. 2014). Outcomes of these studies are very useful to understand the fundamentals of FO processes. However, the real-scale applications of FO processes that use large membrane modules instead of small token sized membranes are more complicated compared to the laboratory scale studies. Spatial variation in DS and FS concentration over a large membrane surface area of the module causes a significantly different process performance. Moreover, the pressure drop and external concentration polarization effects show more influence on the full scale system's performance (Attarde, Jain & Gupta 2016; Kim & Park 2011). Therefore, full scale studies are essential to design and optimize FO plant for real scale applications. Currently, the commercially available membrane modules for FO systems are spiral wound, hollow fiber, and plate and frame type. Most FO studies so far focused on analyzing the performance of the spiral wound module due to its widespread commercial availability (Im et al. 2016; Kim, Blandin, et al. 2017; Kim et al. 2015). However, recently hollow fiber FO module has drawn significant research attention due to its very high packing density and simple hydrodynamics for spacer-less membrane channel (Chou et al. 2010; Lim et al. 2019; Ren & McCutcheon 2018; Tran et al. 2019).

FO experimental studies using hollow fiber modules were mostly conducted employing commercial module and for some specific applications (Corzo et al. 2017; Nikbakht Fini et al. 2020). Therefore these studies are not applicable for other hollow fiber modules and applications. On the other hand, some theoretical studies optimized the hollow fiber module design for FO process. For example, Xiao et al., optimized the flow configuration, solution inlet concentration, flowrate, and membrane properties for module average FO efficiency (Xiao et al. 2012). In another study, effects of applied hydraulic pressure, DS concentration, and the structural parameter of the hollow fiber membrane on FO performance were experimentally measured and the results were theoretically validated (Shibuya et al. 2016). 2D CFD simulation has also been conducted to investigate the effects of membrane water permeability, support layer thickness, module length, and DS concentration on module performance (Ren et al. 2020). However, these studies did not provide any information about the membrane modular arrangement and operating conditions of an FO plant for real scale applications. Attarde et al. estimated the energy consumption of an FO-RO system using hollow fiber membrane module (Attarde et al. 2017). It was found that for RO recovery rate of 50%, 25% energy saving was achieved by FO-RO hybrid system at given operating conditions. However, the scope of the study was limited to the energy consumption of the system neglecting the cost of the membrane modules, and the effects of operating conditions. In another study, optimum membrane length of a hollow fiber FO module was determined for seawater desalination using an FO-RO hybrid process, where RO process was considered as the draw solution regeneration process (Altaee et al. 2019). Energy consumption of the FO-RO process was calculated based on the theoretical minimum energy consumption, which is different from the actual energy consumption. Another modeling and simulation study optimized the module length and operating conditions considering the maximum pure water permeate flow rate and minimum energy consumption (Teklu, Gautam & Subbiah 2020). However, the permeate flowrate of an FO process at any operating condition can be increased, if the dilution of the process is kept low. Therefore, the optimization of the FO process should consider the final volume and concentration of the DS instead of considering the net permeate flowrate. In addition, all of these studies employed the mass transport model of the flat sheet membrane to evaluate the performance of hollow fiber membrane, although these two membranes are geometrically very different.

This study aims to develop system design models for the simulation and optimization of a full scale FO plant using a hollow fiber membrane module. Mathematical models were developed to estimate the performance of the hollow fiber FO module and the information required for the real scale design. Theoretical results were then validated by comparing with our pilot scale experimental data using an available commercial hollow fiber module. The validated models were then applied for the design of a 1,000 m^3/day FO plant to produce osmotically diluted DS of 0.25 M, 0.2 M, and 0.15 M concentrations using 0.6 M NaCl solution as DS (~seawater) and 0.02 M NaCl solution FS (~MBR effluent). Appropriate membrane modular arrangement was selected from the module scale performance and the applicable ranges of operating conditions for the optimization were decided according to the manufacturer's operation guideline. Required number of membrane modules and specific energy consumption to obtain the desired production capacity and the final DS concentration were theoretically evaluated at various operating conditions. Finally, the optimum design and operating conditions were determined based on the cost incurred for the membrane module and energy consumption.

5.2 Operating conditions for design and optimization

A full scale FO system consists of an array of membrane modules connected either in serial or parallel or in a combination of serial and parallel configuration. Generally, the membrane modules are connected serially and assembled in a pressure vessel to obtain a desired production capacity and final DS concentration. Since the total production capacity of the FO plant cannot be obtained from a single pressure vessel, numerous pressure vessels are added in parallel connection. However, Operating conditions of the FO system play a key role in designing full scale plant. DS and FS inlet flowrate and concentration are four major operating conditions of the system. For the optimum design of the FO system, the operating conditions are varied in an applicable range to obtain the best possible performance. However, for most of the FO applications the feed and draw solutions are predefined such as seawater, RO brine, MBR effluent etc. So the DS and FS inlet concentrations remain the same and cannot be varied to optimize for these applications. Therefore, this study varied the DS and FS inlet flowrate to design and optimize an FO plant using a commercial hollow fiber module for the desired production capacity and final DS concentration.

5.3 Theoretical backgrounds

Theoretical models presented in this section to analyze the performance of hollow fiber module were developed from the rigor of the membrane mass transport and fluid mass conservation fundamentals. These models considered the operating conditions of the FO system and the fluid flow behaviors in the shell and bore sides of the module. Element scale performances were employed to determine the information required to design a plant with the desired production capacity and final DS concentration.

5.3.1 Mass transport models for hollow fiber membrane

Mass transport models for a flat sheet FO membrane are well established. But the flat sheet and hollow fiber membranes are geometrically very different. Sivertsen et al. developed mass transport models of the hollow fiber membrane for PRO process considering the actual geometric configuration of the membrane (Sivertsen et al. 2013). This section modified the models for the outer selective FO hollow fiber membrane. Figure 5.1, describes the geometric structures of the hollow fiber membrane. It shows the inner radius(r_i), outer radius(r_o), the thickness (t_s) of the membrane support layer and the shell radius of one membrane(r_{shell}). The selective layer of the membrane is so thin that its thickness is considered negligible. It also shows the distribution of concentration over the membrane.

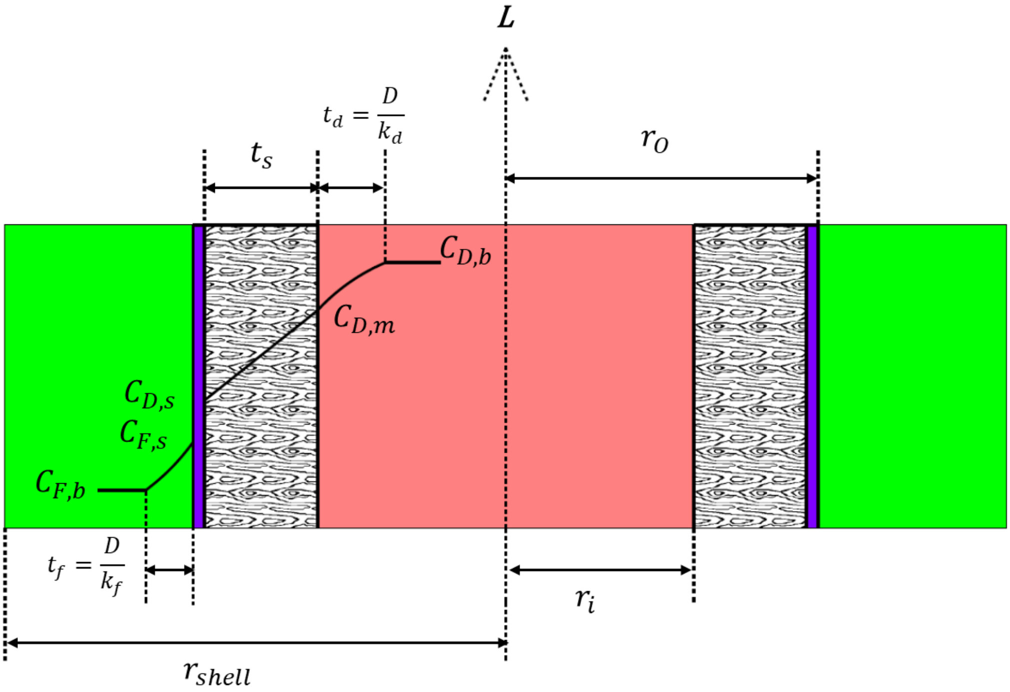


Figure 5.1. Geometric description and concentration profile of the outer selective FO hollow fiber membrane.

FS concentration on the selective layer considering the external concentration polarization effect is given by,

$$C_{F,s} = \left(C_{F,b} - \frac{J_s}{J_w}\right)\left[\frac{r_0 + D/k_f}{r_0}\right]^{\left(\frac{J_w r_0}{D}\right)} + \frac{J_s}{J_w} \tag{5.1}$$

here, $C_{F,b}$ is the bulk concentration of the FS, D is the diffusivity coefficient of the solution, k_f is the mass transfer coefficient of the flow, r_0 is the outer diameter of the membrane, J_w and J_s are the pure water and reverse solution of the membrane. On the bore side of the membrane due to the permeation of water dilutive ECP occurs, where the concentration of the draw solution reduces from $C_{D,b}$ to $C_{D,m}$. DECP of the hollow fiber membrane is quantified by,

$$C_{D,m} = \left(C_{D,b} - \frac{J_s}{J_w}\right)\left(\frac{r_i - D/k_d}{r_i}\right)^{\frac{J_w r_0}{D}} + \frac{J_s}{J_w} \tag{5.2}$$

here, r_i is the inner diameter of the membrane and k_d is the mass transfer coefficient of the draw side. ICP occurs in the porous substrate of the membrane where the DS concentration drops from $C_{D,m}$ to $C_{D,s}$ which is expressed as,

$$C_{D,s} = \left(C_{D,b} - \frac{J_s}{J_w}\right)\left(\frac{r_i - D/k_d}{r_i}\right)^{\frac{J_w r_0}{D}} \left(\frac{r_i}{r_0}\right)^{\frac{J_w r_0 \tau}{D\emptyset}} + \frac{J_s}{J_w} \tag{5.3}$$

here, $\emptyset$ and τ are porosity and tortuosity of the membrane. However, these parameters are generally not measured directly and reported in the literature, whereas structural parameter, S is a more common term to characterize the FO membrane. The structural parameter is defined as $S = (t_s\tau)/\emptyset$, where t_s is the thickness of the membrane support layer. Thickness of the support layer can be written as $r_0 - r_i$. Replacing $\emptyset$ and τ in equation (5.3),

$$C_{D,s} = \left(C_{D,b} - \frac{J_s}{J_w}\right)\left(\frac{r_i - D/k_d}{r_i}\right)^{\frac{J_w r_0}{D}} \left(\frac{r_i}{r_0}\right)^{\frac{J_w r_0 S}{D(r_0 - r_i)}} + \frac{J_s}{J_w} \tag{5.4}$$

Therefore the effective concentration gradient across the selective layer of the outer selective hollow fiber membrane can be expressed by,

$$\Delta C_s = \left(C_{D,b} - \frac{J_s}{J_w}\right)\left(\frac{r_i - D/k_d}{r_i}\right)^{\frac{J_w r_0}{D}} \left(\frac{r_i}{r_0}\right)^{\frac{J_w r_0 S}{D(r_0 - r_i)}} - \left(C_{F,b} - \frac{J_s}{J_w}\right)\left(\frac{r_0 + D/k_f}{r_0}\right)^{\frac{J_w r_0}{D}} \tag{5.5}$$

Using the concentration gradient across the selective layer, the actual osmotic pressure gradient across the selective layer ($\Delta\pi_s$) can also be calculated (Phuntsho et al. 2014). Then the water flux is evaluated by the solution diffusion equation as follows,

$$J_w = A[\Delta\pi_s + (P_F - P_D)] \tag{5.6}$$

here, A is the water permeability coefficient of the membrane, P_F and P_D are the hydraulic pressure applied in the feed and draw channel. Similarly the reverse solute flux, J_s is given by,

$$J_s = B\left[\left(C_{D,b} - \frac{J_s}{J_w}\right)\left(\frac{r_i - D/k_d}{r_i}\right)^{\frac{J_w r_0}{D}}\left(\frac{r_i}{r_0}\right)^{\frac{J_w r_0 S}{D(r_0 - r_i)}} - \left(C_{F,b} - \frac{J_s}{J_w}\right)\left(\frac{r_0 + D/k_f}{r_0}\right)^{\frac{J_w r_0}{D}}\right] \tag{5.7}$$

where B is the solute permeability coefficient of the membrane. A, B, and S values of the CTA hollow fiber membrane were considered as 0.27 L/m^{2}hbar, 0.035 L/m^{2}h and 1024 respectively (Shibuya et al. 2016).

5.3.2 Module scale performance analysis

Water and solute transport models presented in the previous section are incorporated with the fluid mass conservation theory to analyze the module scale performance of the hollow fiber FO module. Further, these models were employed to design a real scale FO plant for a desired production capacity and final DS concentration. For the module scale modeling, a uniform distribution of fibers with equal distance between adjacent fibers was assumed. In order to determine the characteristic dimensions of the module, number of membrane fibers were calculated by,

$$N_{fbr} = \frac{A_{ms}}{\pi d_0 l_{fbr}} \tag{5.8}$$

here, A_{ms} is the total membrane surface area of the module, d_0 represents the outer diameter of the fiber and l_{fbr} is the length of the fiber. This information is available in the manufacture’s datasheet and listed in Table A2.1 of the Appendix section. Shell area for each fiber is calculated by,

$$A_{shel} = \frac{A_{m,in}}{N_{fbr}} = \frac{\pi {D_{m,in}}^2}{4N_{fbr}} \tag{5.9}$$

where, $A_{m,in}$ is the cross-sectional area of the module considering the inner diameter of the module, $D_{m,in}$. Therefore the diameter of the shell zone for each fiber is given by,

$$d_{shell} = \sqrt{\frac{4A_{shell}}{\pi}} \tag{5.10}$$

Void fraction, ε is another important characteristic dimension of the module which is calculated by,

$$\varepsilon = 1 - \frac{A_{fbr}}{A_{shell}} = 1 - \frac{{d_0}^2}{{d_{shell}}^2} \tag{5.11}$$

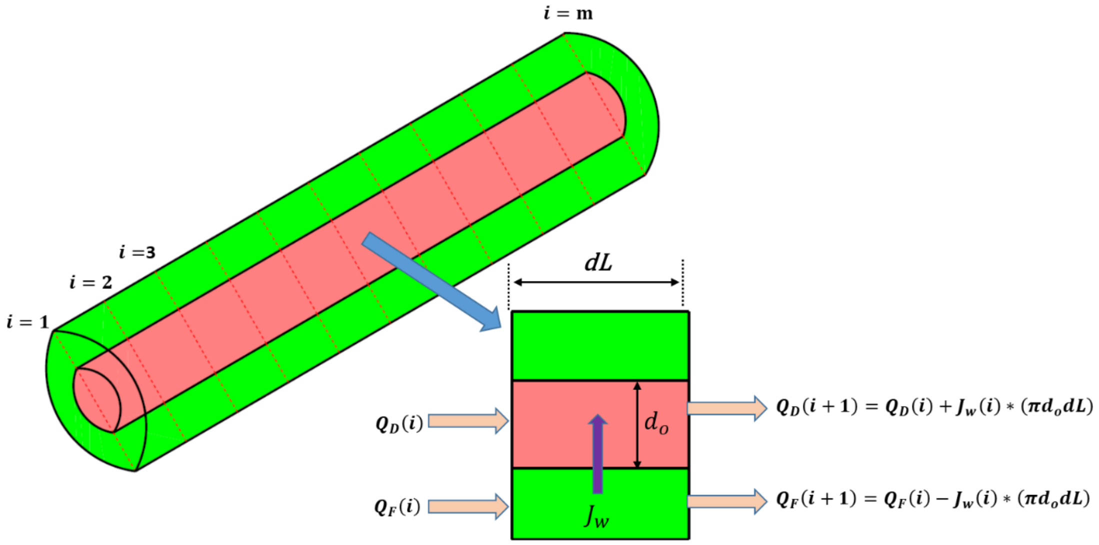


Figure 5.2. Numerical simulation method for hollow fiber membrane.

Further, both feed and draw solution flows longitudinally through the shell and bore sides of the module respectively. Although the radial flow of solutions may occur at the inlet and outlet of the module, considering the sufficiently long membrane length, radial flow can be neglected for the analysis. Since the radial flow is negligible, each fiber of the module performs almost equally in the module. Therefore, this study analyzed the performance of one fiber and determined the module scale performance by multiplying it with the total number of fibers. But calculating the performance of the fiber considering the same feed and draw solution concentration for the entire fiber length is not accurate. Since the permeation of water and solute changes the concentration and velocity of the solution at various locations of the fiber, water flux, and reverse solute flux of the hollow fiber membrane is required to be calculated at very small sections of the membrane. In addition, the effects of the permeation form each section should be taken into account for the neighboring sections by applying fluid conservation theory. Figure 5.2 schematically depicts the changes in solution flow and flux permeation through the discretized sections of the fiber and the surrounding shell zone. Length of each discrete section is dl. Local values of DS flowrate at any small section is given by,

$$Q_{DS}(i+1) = Q_{DS}(i) + J_w(i) \times (\pi d_0 \times dl) \tag{5.12}$$

here, Q_{DS} is the DS flowrate, J_w is the water flux through the membrane and d_0 refers to the outer diameter of the fiber. Moreover, i is an index that indicates the location of the discrete cell. Similarly, the FS flowrate is given by,

$$Q_{FS}(i+1) = Q_{FS}(i) - J_w(i) \times (\pi d_0 \times dl) \tag{5.13}$$

Since the permeation of water flux changes the DS flowrate, it also changes the DS and FS velocity. DS velocity at any location is given by,

$$U_{DS}(i) = \frac{4Q_{DS}(i)}{\pi {d_i}^2} \tag{5.14}$$

d_i is the inner diameter of the fiber. Considering the shell area, FS velocity is presented by,

$$U_{FS}(i) = \frac{4Q_{FS}(i)}{\pi\left({d_{shell}}^2 - {d_0}^2\right)} \tag{5.15}$$

here, d_{shell} is the diameter of the shell zone for each fiber. Apart from the solution flowrate and velocity, the solute flowrates through the shell and bore sides of the fiber were also estimated. Draw solute flowarte is mathematically expressed by,

$$Q_{DS,slt}(i) = Q_{DS}(i)C_{DS}(i) - J_s(i) \times (\pi d_0 \times dl) \tag{5.16}$$

here, C_{DS} is the concentration of the DS. Similarly, the feed solute flowrate is given by,

$$Q_{FS,slt}(i) = Q_{FS}(i)C_{FS}(i) + J_s(i) \times (\pi d_0 \times dl) \tag{5.17}$$

Considering the solute flowrate and solution flowrate, the concentrations of the solutions were calculated. DS concentrations at different discrete locations are estimated by,

$$C_{DS}(i) = \frac{Q_{DS,slt}(i)}{Q_{DS}(i)} \tag{5.18}$$

Similarly, the FS concentration is given by,

$$C_{FS}(i) = \frac{Q_{FS,slt}(i)}{Q_{FS}(i)} \tag{5.19}$$

In addition to the concentration and velocity, hydraulic pressure applied by the solution flow also influences the mass transport through the membrane. DS pressures at different discrete locations were determined by Hagen–Poiseuille equation as below (Attarde et al. 2017; Sivertsen et al. 2013),

$$P_{DS}(i+1) = P_{DS}(i) - \frac{32\mu U_{DS}(i)}{{d_i}^2} dl \tag{5.20}$$

FS pressure on the shell side is calculated by employing a modified Ergun equation (Shibuya et al. 2016).

$$P_{FS}(i+1) = P_{FS}(i) - \left[\frac{K_1(1-\varepsilon)^2\mu U_{FS}(i)}{\varepsilon^3(1.5d_0)^2} + \frac{K_2(1-\varepsilon)\rho {U_{FS}}^2(i)}{\varepsilon^3(1.5d_0)^2}\right] dl \quad (5.21)$$

here K_1 and K_2 are the two empirical constants. In this study, the values of these constants were estimated to be 150 and 45 for the proposed hollow fiber by fitting the model values with the experimental pressure drop values (Teklu, Gautam & Subbiah 2020). Determination of K_1 and K_2 are discussed more detailed in Table A2.2 of the Appendix of this dissertation.

Total water permeation of the module is expressed by,

$$Q_{net,perm} = N_{fibre} \sum_{i=1}^{m} J_w(i) \times (\pi d_0 \times dl) \quad (5.22)$$

Here, m is the total number of the discrete sections on the hollow fiber. Average flux of the module is then calculated by,

$$J_{w,avg} = \frac{Q_{net,perm}}{A_m} \quad (5.23)$$

Total production of the diluted DS from the module is given by,

$$Q_{DS,out} = N_{fibre} \sum_{i=1}^{m} Q_{DS}(i) \quad (5.24)$$

Specific energy consumption of the system was also computed by the following expression,

$$SEC = \frac{Q_{Fs,in} \times N_M \times P_{FS,in} + Q_{DS,in} \times N_M \times P_{DS,in}}{Q_{DS,out} \times N_M} \quad (5.25)$$

here, $Q_{Fs,in}$ and $Q_{Ds,in}$ are the FS and DS inlet flowrates to each module, $P_{FS,in}$ and $P_{DS,in}$ stand for the FS and DS inlet pressure respectively, $Q_{DS,out}$ is the produced diluted DS from each module and N_M is the number of modules connected in parallel configuration.

5.4 Results and discussion

The theoretical models developed were first applied to predict the performance of the system at different operating conditions considering the FS and DS flowrates, concentrations, inlet pressure, and pressure drop in the membrane. These simulation results were then compared to actual results obtained from pilot-scale experimental results for model validation. Finally, the module scale performances were employed to design and optimize a 1,000 m^3/day FO plant to produce 0.15 M, 0.2 M and 0.25 M diluted seawater. Optimum DS and FS inlet flowrate and the number of modules required for the plant were calculated by comparing the cost of membrane modules and the energy consumption of the system.

5.4.1 Validation of theoretical models

To evaluate the reliability of the proposed models for the hollow fiber FO membrane module, the simulation results were first validated by comparing with the experimental results obtained from the pilot scale experiment using the same module and operating condition. For this benchmark experiment, the module scale performances were measured in terms of average water flux and outlet DS concentration as a function of different DS inlet concentrations. NaCl solution and DI water were used as DS and FS for these experiments. Since the bore diameter of the hollow fiber membrane (85 μm) and the void fraction (0.45) of the module were too small, experiments were conducted at a relatively low feed and draw solution inlet flowrate to avoid high solution inlet pressure to the module. The FS and DS flowrate to each module (175 μm fibers, 31.5 m^2 effective membrane area) for the experiments were 0.7 L/min and 0.35 L/min respectively.

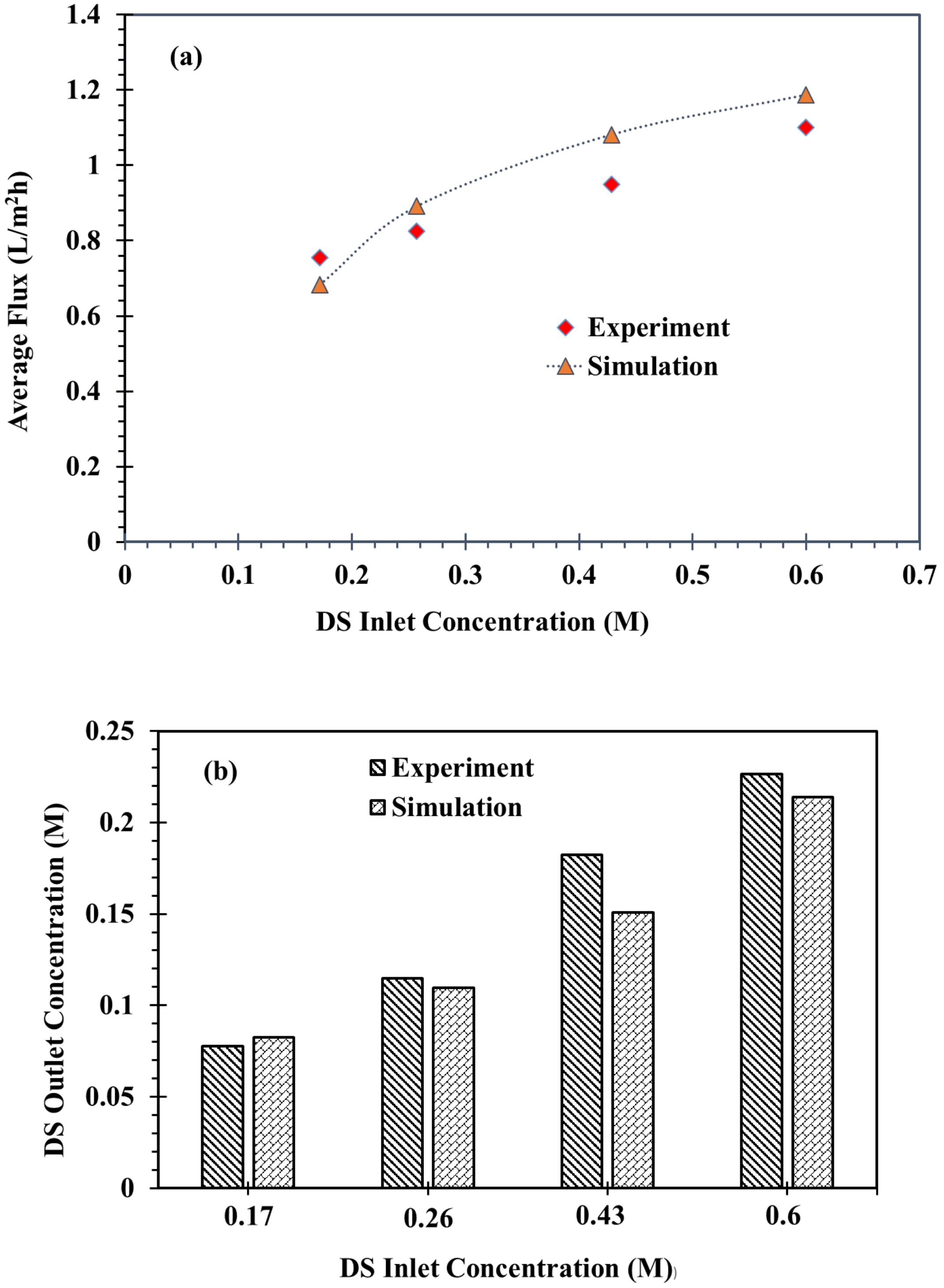


Figure 5.3. Theoretical model validation by comparing experimental and simulation results of (a) the average water flux of the module and (b) the DS outlet concentration as a function of DS inlet concentration. NaCl solution and DI water were used as FS and DS. Inlet flowrates of FS and DS were 0.7 L/min and 0.35 L/min respectively.

Commercial hollow fiber CTA FO membrane module consists of 175 μm diameter fibers with a membrane area of 31.5 m^2).

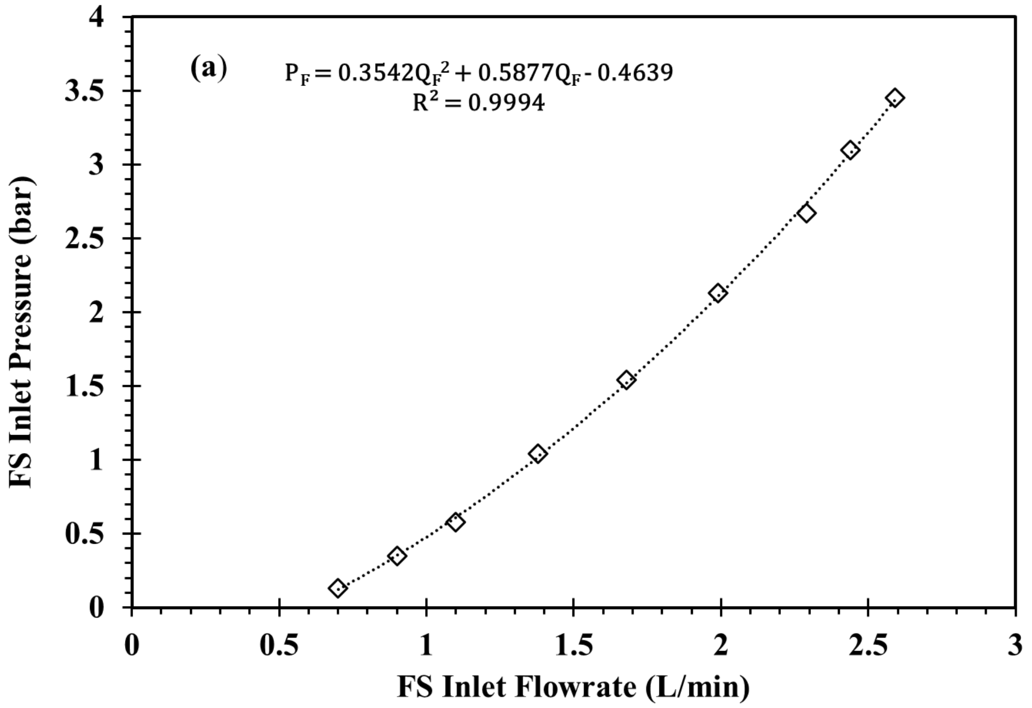
Figure 5.3(a) shows the experimental and simulated average water fluxes of the hollow fiber membrane module as a function of DS inlet concentration at fixed FS and DS inlet flowrates. Experimentally it was found that the average flux increased from 0.75 L/m^2h to 1.10 L/m^2h when the DS inlet concentration increased from 0.17 M to 0.6 M and these results were closely similar to the simulated results water fluxes of 0.68 L/m^2h to 1.18 L/m^2h obtained under the same operating conditions. The DS outlet concentration is an important performance parameter of the FO module which was measured to validate the models. Figure 5.3(b) compared the experimental and theoretical concentration values of the DS outlet for the same operating conditions mentioned above. As shown in the figure, the theoretical and the experimental results were very closely similar in values with an average deviation of only about 10% between them which validates the accuracy of the models used. The DS outlet concentration increased from 0.07 M to 0.23 M, whereas the numerical simulation found that the outlet concentration increased from 0.08 M to 0.21 M when the inlet concentration increased from 0.17 M to 0.6 M. The simulated water fluxes were slightly higher than experimental results which also explains slightly lower final diluted DS concentrations. However, this small differences between the experimental and simulated values can be attributed to both experimental error and the non-ideality of the assumptions made in the models while considering the axial flow of the solutions and the equal distribution of FS and DS flowrate through each fiber and shell regions of the module.

5.4.2 Selection of operating conditions and modular arrangement

Design of a large scale FO plant for real applications involves the calculation of the number of membrane elements, their optimum arrangement, and optimization of the operating conditions to obtain the desired production capacity and final DS concentration. Since this study focused on designing and optimizing an FO system to osmotically dilute seawater, it considered 0.6 M NaCl solution as the DS of the system, which is comparable to seawater in terms of osmotic pressure (27 bar). Similarly, 0.02 M NaCl solution was assumed as the FS considering its osmotic pressure which is almost similar to the secondary (MBR) effluent (0.99 bar). Moreover, the fouling potential of MBR effluent is very low, the fouling effect of the feed solution was not accounted for the simplicity in the models (Ali et al. 2018; Kim et al. 2018). Moreover, the performance of the hollow fiber module is estimated at a reasonable range of operating conditions considering the operational guidelines of the manufacturer. In this study, maximum FS inlet flowrate was selected based on the maximum allowable shell side inlet pressure of the module. As per the manufacturer's manual of the Toyobo CTA hollow fiber FO module, a maximum of 4.9 bar pressure can be applied at the shell side of the module. A second-degree polynomial function has been proposed to estimate the FS inlet pressure at various FS inlet flowrate by curve fitting method employing previously published experimental data (Teklu, Gautam & Subbiah 2020) as shown in Figure 5.4(a). The function is given by,

$$P_F = 0.3542{Q_F}^2 + 0.5877Q_F - 0.4639 \tag{5.26}$$

here P_F and Q_F are the FS pressure (bar) and FS flowrate (L/min) at the inlet of the module.



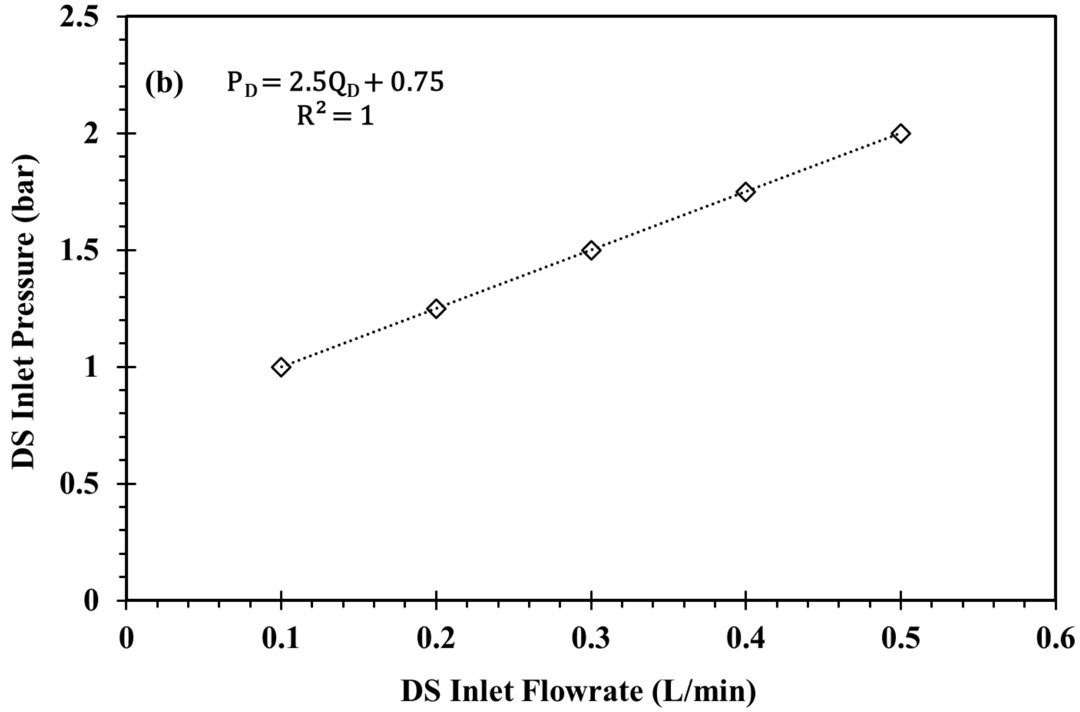


Figure 5.4. Relation between (a) the shell (Feed) and (b) bore side inlet flowrate and pressure.

Employing the polynomial function, it was found that 3.15 L/min inlet flowrate is the maximum FS flowrate that can be applied on the shell side as it causes 4.9 bar pressure at the inlet of the module. Therefore, 3 L/min is considered as the maximum FS inlet flowrate for this analysis.

Table 5.1 Operating conditions and design objectives considered for the simulation

Operating Conditions		**Design Objectives**	
Feed solution	MBR effluent	Production Capacity	1,000 m^3/day
FS Inlet Conc.	0.02 M	Final DS Conc. (FDC)	0.25 – 0.15 M
FS Inlet Flowrate	2 – 3 L/min		
Superficial velocty	0.016 - 0.025 m/s		
Draw solution	Seawater		
DS Inlet Conc.	0.6 M		
DS Inlet Flowrate	0.25 –1 L/min		
Superficial velocty	0.007 - 0.029 m/s		

Similarly, the relation between the DS inlet flowrate and the bore side inlet pressure was formulated by using the technical data provided in the manual of the module which is shown in Figure 5.4(b). The correlation is expressed as (TOYOBO 2016),

$$P_D = 2.5Q_D + 0.75 \qquad (5.27)$$

where, P_D and Q_D are the DS inlet pressure (bar) and flowrate (L/min) of the module. In order to achieve sufficient dilution of the DS and to maintain a positive pressure difference between the feed and draw solution side of the membrane, DS inlet flowrate is considered smaller than the FS flowrate. In this study, the maximum DS inlet flowrate is considered as 1 L/min which is half of the minimum FS inlet flowrate, 2 L/min, which ensures about 50% recovery rate at these conditions. In addition, the minimum DS inlet flowrate was considered as 0.25 L/min, because at a very low DS inlet flowrate the whole membrane surface area is not used. Therefore, the range of inlet flowrate for the FS was 2 L/min to 3 L/min. In comparison, the DS flowrate varied from 0.25 L/min to 1 L/min as shown in Table 5.1. Therefore, considering rhe dimensions of the membrane module the superficial velocity of the FS was varied from 0.016 to 0.025 ms/s. On the other hand the velocity of DS was varied from 0.007 to 0.029 m/s.

Apart from the selection of proper operating conditions, it is also very important to decide the suitable modular arrangement of the membrane modules. Generally, for large scale FO system using flat sheet membranes, few spiral wound membrane elements/modules are connected serially and housed in a pressure vessel. The number of modules per pressure vessel depends on the required final DS concentration and production capacity of the plant. Then several pressure vessels are connected parallel to each other to attain the total production capacity of the plant. However, compared to some other membrane modules (such as spiral wound membrane module), the current commercial hollow fiber module is operated at a significantly lower feed and draw solution flowrate, which leads to a high reduction in osmotic pressure gradient after the first module.

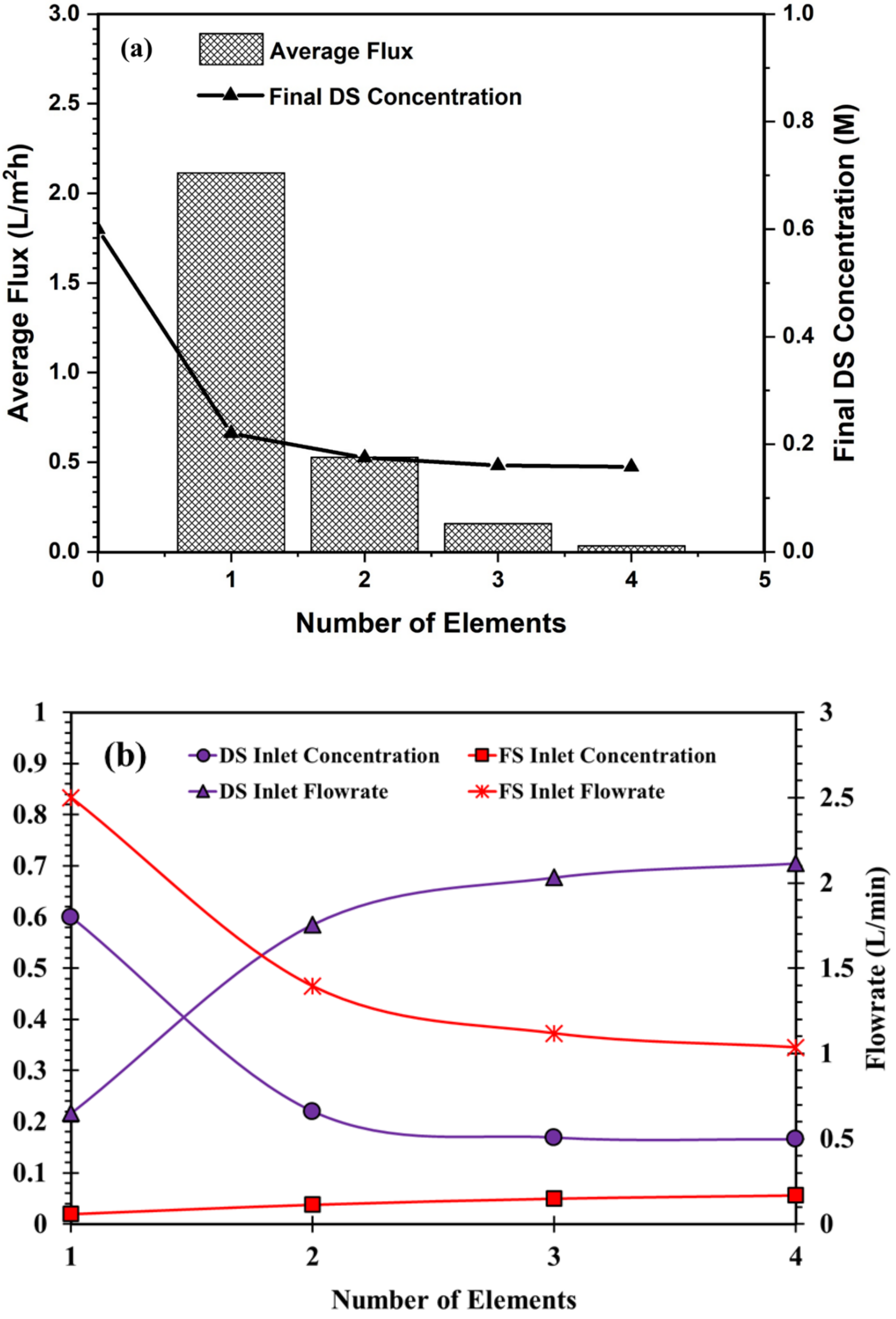


Figure 5.5. Serially connected hollow fiber membrane module's (a) performance at 2.5 L/min and 0.65 L/min FS and DS inlet flowrate. (b) Inlet conditions for each element.

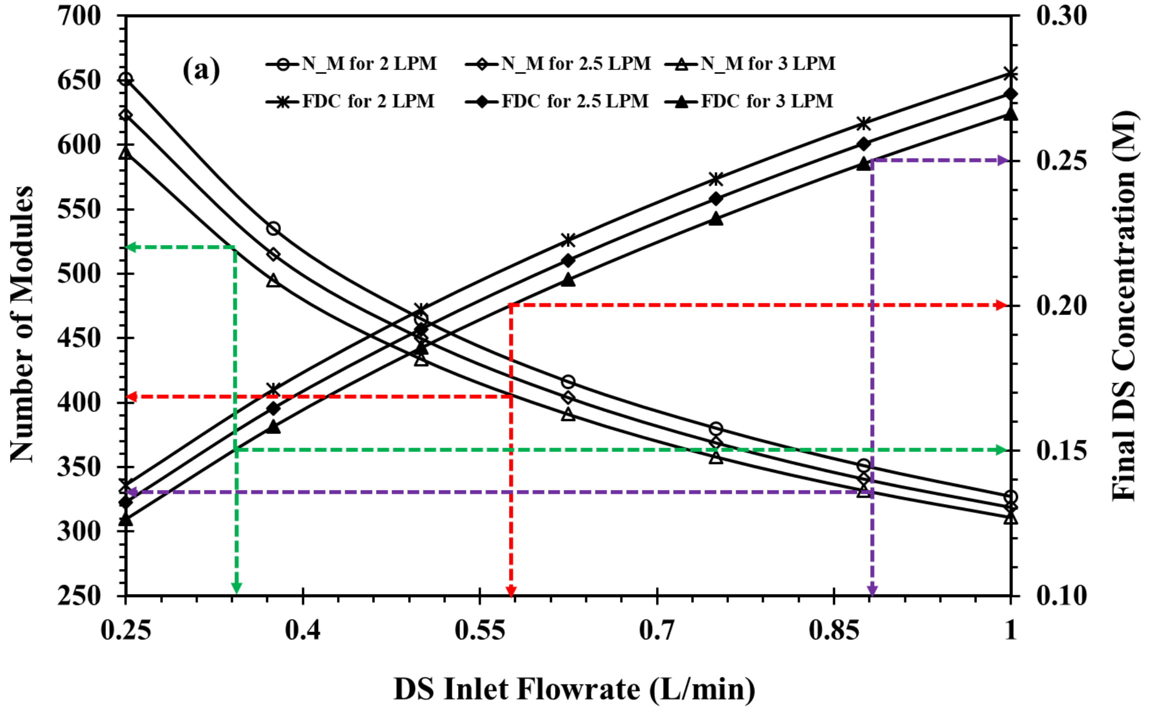
Therefore, whether a serial connection of the membrane modules was essential or not was first verified by estimating the performances of up to 4 serially connected hollow fiber modules. Figure 5.5(a) shows the performance of the elements in terms of average water flux and final DS concentration when the FS and DS inlet flowrate to the first module were 2.5 L/min and 0.65 L/min respectively. The average water flux of the module drops from 2.10 L/m^{2}h for single module to 0.53 L/m^{2}h for two modules in series. The addition of subsequent modules in series, the water flux was further reduced very significantly. In addition, the final DS concentration achieved by the single element module was 0.22 M. However any addition of membrane elements in series did not contribute significantly to the dilution of the final DS concentrations.

Figure 5.5(b) describes the FS and DS conditions at the inlet of each membrane element for the serially connected module. As a result of water permeation through the membrane, DS flowrate increased from 0.65 L/min at the inlet of the first element to 1.76 L/min at the inlet of the second element in the series module. On the other hand, FS inlet flowrate reduced from 2.5 L/min for the inlet of the first element to 1.39 L/min for inlet of the second element. Due to this change in flowrates, concentration difference between the DS and FS reduced from 0.58 M at the inlet of the first element to 0.18 M at the inlet of the second element in a series module. Besides, the DS pressure was found greater than the FS pressure which resists the water transport from feed to draw side (according to equation 5.26 and 5.27). Consequently, the combined effects of lower osmotic pressure gradient and higher DS pressure contributed to a significantly lower average flux and dilution by the second and the following modules. Therefore, it is clear that connecting the module serially will not only deteriorate the performance of the system but also increase the footprint of the system and energy consumption due to higher pressure drop across the modules. Therefore, this study considers only parallel configuration. Section

5.3 and 5.4 will discuss the estimation of the minimum number of elements and DS inlet flowrate per module required to design a 1,000 m^3/day FO plant.

5.4.3 Required number of modules

As described in the previous section, parallel connection of a single-element module of the Toyobo hollow fiber membrane module is considered for the FO plant in this study. According to the parallel configuration, the total number of module (N_M) is obtained by diving the production capacity (λ) of the plant with the DS outlet flowrate ($Q_{D,out}$) of each hollow fiber module, which can be expressed as, $N_M = \lambda / Q_{D,out}$. Moreover, the final DS concentration of the plant in the parallel configuration is equal to the DS outlet concentration of each element. In this section, feed and draw solution inlet flowrate refers to the flowrate for each module connected in the system.



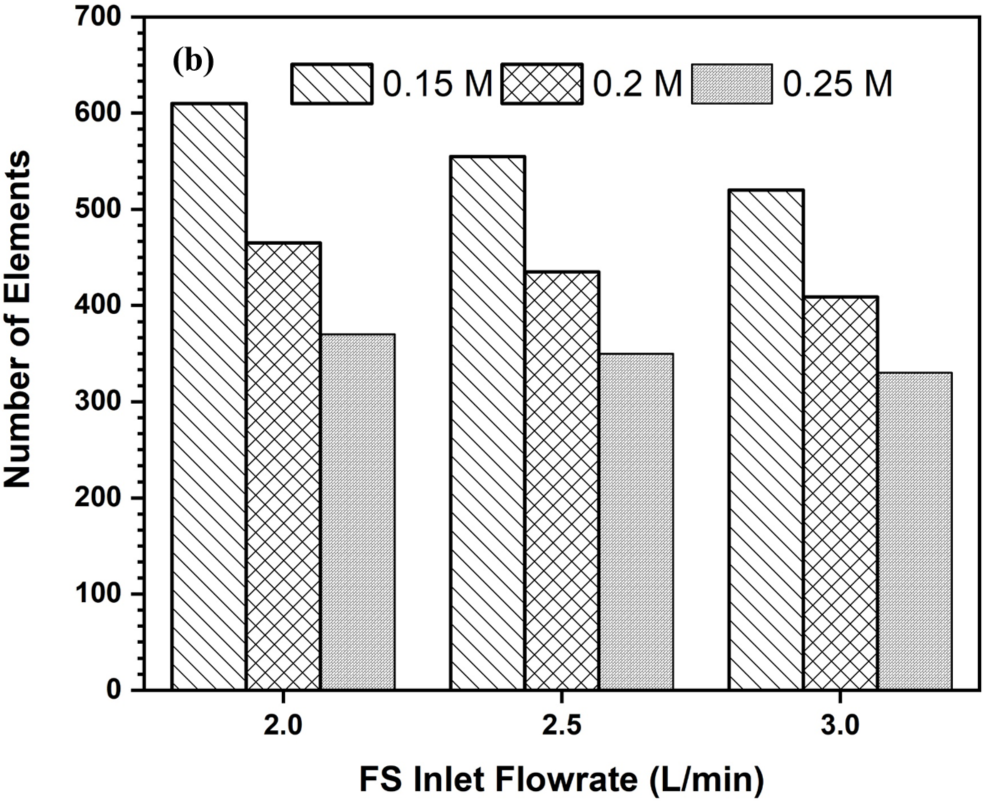


Figure 5.6. (a) Required number of membrane modules and final DS concentration at various DS inlet flowrate per module at 2 L/min, 2.5 L/min, and 3 L/min FS inlet flowrate for a 1000 m^3/day FO plant. Violet, red and green arrow indicated the number of elements and minimum DS inlet flowrate corresponding to the 0.25 M, 0.2 M, and 0.15 M final DS concentration at 3 L/min FS inlet flowrate. (b) Number of elements required to achieve 0.15 M, 0.2 M, and 0.25 M final DS concentration of the plant. FS and DS inlet flowrates stand for flowrate to each module

Figure 5.6(a) and (b) exhibit the required number of modules and final DS concentration of a 1000 m^3/day FO system at various DS inlet flowrate per module for 2 L/min, 2.5 L/min, and 3 L/min FS inlet flowrate to each module. At any specific FS inlet flowrate, the number of element decreased with the increasing DS inlet flowrate and this is because, higher DS inlet flowrate results in a lower dilution of the DS as shown in the same Figure along secondary y-axis, it maintains higher osmotic pressure difference or driving force across the membrane. Consequently, the average flux of the module is higher, which contributed to the reduction in number of modules required for the FO plant. For instance, the number of modules was reduced from 651 to 327 when the DS inlet flowrate per module increased from 0.25 L/min to 1 L/min, and the FS inlet flowrate remains the same at 2 L/min as shown in Figure 5.6(a). This reduction in the number of membrane modules, however, comes at a cost in the form of correspondingly increased final DS concentration of the FO system as the DS inlet flowrate increases. It was found that for the employed hollow fiber module the DS outlet concentration increased from 0.14 M to 0.28 M for the same range of DS inlet flowrate. Further, the increase in the FS inlet flowrate for any DS inlet flowrate contributed to the reduction in both number of modules and final DS concentration. As the increasing FS flowrate reduces the external concentration polarization effect and increases the FS pressure, the mass transport is enhanced. Consequently, the increased water flux at higher FS flowrate resulted in lower DS outlet concentration and the required number of modules. It can be seen from Figure 5.6(a) that at 0.63 L/min DS inlet flowrate a total of 416, 404, and 391 hollow fiber modules are required to produce 1000 m^3/day at when the FS inlet flowrates were 2 L/min, 2.5 L/min, and 3 L/min respectively. On the other hand, final DS concentration slightly reduced from 0.22 M to 0.20 M for the same variation in the FS inlet flowrate. Therefore, comparing the effects of DS inlet flowrate and FS inlet flowrate on the number of modules and final DS concentration using the hollow fiber module used in this study, DS flowrates found to have more influence on the FO system performance. However, Figure 5.6(a) although show the number of elements required for a 1000 m^3/day plant and its final DS concentration, but these figures do not provide the information about the number of modules required to design the 1000 m^3/day FO system at desired final DS concentration (0.25 M, 0.2 M and 0.15 M in this study).

Finally, the minimum number of modules required to design the FO plant for different desired final DS concentration was determined from the trends of Figure 5.6(a) and presented in Figure 5.6(b). Violet arrow on the figure shows the required number of elements and DS inlet flowrate corresponding to the desired 0.25 M final DS concentration at 3 L/min FS inlet flowrate. On the other hand, red and green arrows show the parameters at 0.2 M and 0.15 M final concentration. As shown in Figure 5.6(b), at any FS flowrate for the lower final DS concentration requirement more modules were needed to be connected. For example, at 2.5 L/min FS inlet flowrate, the number of modules increased from 350 to 555 when the desired final concentration lowered from 0.25 M to 0.15 M. It was also found that less number of modules are required to design an FO plant for a certain final DS concentration when the plant is operated at a higher FS inlet flowrate. At 0.15 M final concentration, total 605, 555, and 510 modules are required when the FS inlet flowrates are 2 L/min, 2.5 L/min, and 3 L/min respectively.

5.4.4 DS inlet flowrate per module

The DS inlet flowrate per module has a significant impact on the FO system performance. Therefore, to design an optimum system it is essential to estimate the DS inlet flowrate for each module that ensures the desired final DS concentration and production capacity of the plant. DS flowrates required to achieve 0.25 M, 0.20 M, and 0.15 M final concentration at 2 L/min, 2.5 L/min, and 3 L/min FS inlet flowrate per module were determined from the trends of final DS concentration curves of Figure 5.6(a) and plotted in Figure 5.7. According to the figure, at any FS flowrate, for higher dilution requirement, the system is required to be operated at a lower DS inlet flowrate. For 2.5 L/min FS inlet flowrate, the DS inlet flowrate required to obtain 0.25 M, 0.2 M, and 0.15 M final concentration were found to be 0.8 L/min, 0.54 L/min, and 0.29 L/min respectively.

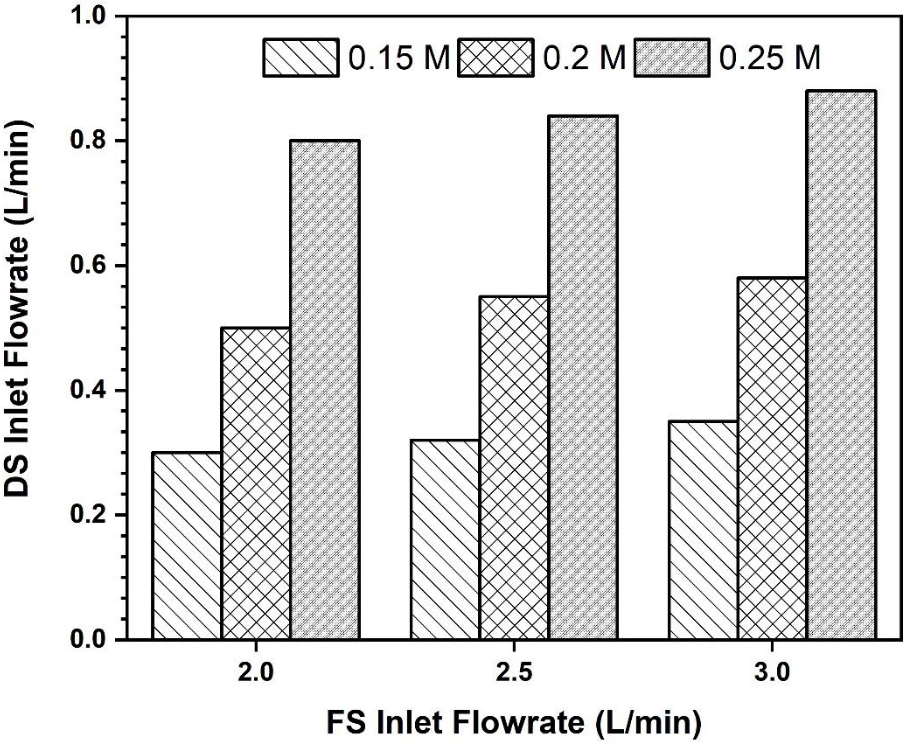


Figure 5.7. DS inlet flowrate for 0.25 M, 0.2 M and 0.15 M final concentration for 2 L/min, 2.5 L/min and 3 L/min FS inlet flowrate. FS and DS inlet flowrate means the flowrate for each module.

In addition, the DS inlet flowrate slightly increased at higher FS flowrate for the same dilution requirement. It can be found that for 0.2 M desired concentration, the DS flowrate increased from 0.5 L/min to 0.58 L/min when the FS inlet flowrate increased from 2 L/min to 3 L/min.

5.4.5 Specific energy consumption of the system

Although it was found from Sections 5.3 and 3.4 that the increase in DS and FS inlet flowrate reduces the required number of membrane modules, the higher flowrate leads to higher energy consumption by the system as the energy consumption is directly proportional to the flowrates.

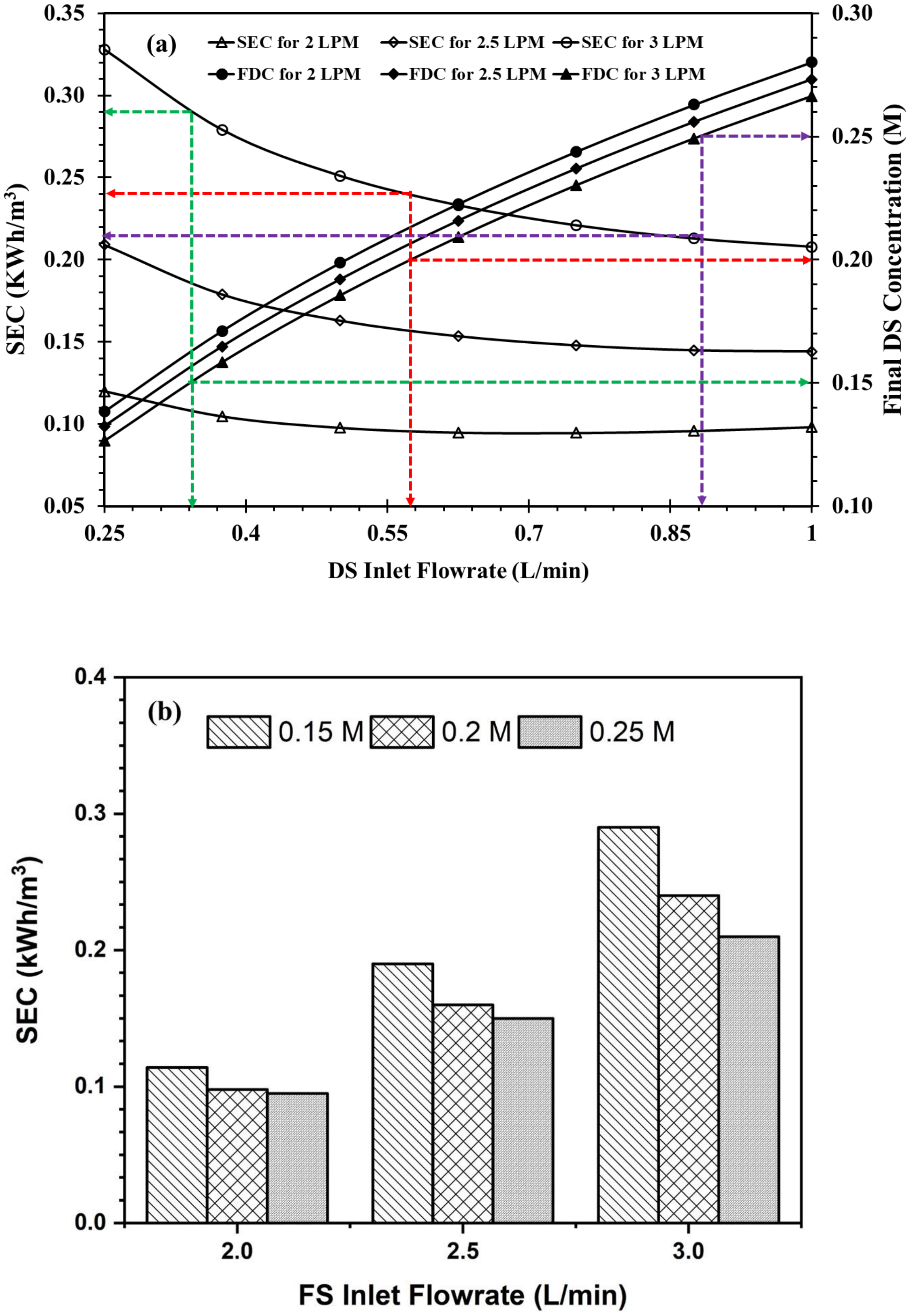


Figure 5.8. (a) Specific energy consumption and final DS concentration at various DS Inlet flowrate. (b) Specific energy consumption for 0.25 M, 0.2 M and 0.15 M final

concentration at various FS inlet flowrate. FS and DS inlet flowrate here represent the flowrate to each module.

Therefore, the specific energy consumption of the FO system for different operating conditions using the proposed hollow fiber module was also calculated to determine the optimum design. Figure 5.8(a) show the specific energy consumption for 2 L/min, 2.5 L/min, and 3 L/min FS flowrate at various DS inlet flowrate. These figures also show the final DS concentration at various operating conditions to determine the specific energy consumption corresponding to the 0.25 M, 0.2 M, and 0.15 M final diluted DS concentrations. However, it was found from the figures that the increase in DS inlet flowrate per module at any FS flowrate per module reduces the specific energy consumption of the system. Since the higher DS flowrate per module reduces the number of modules significantly, therefore although the flowrate per module increases total draw and feed solution flowrate reduces. Consequently, the specific energy consumption of the system reduces at higher DS flowrate per module.

For example, at 2 L/min FS inlet flowrate per module, the specific energy consumption of the system slightly reduced from 0.12 kWh/m^3 to 0.10 kWh/m^3 when the DS flowrate per module increased from 0.25 L/min to 1 L/min as shown in Figure 5.8(a). Despite the effects of DS flowrate on specific energy consumption at lower FS flowrate was very small, Figure 5.8(a) also showed that at higher FS flowrate, specific energy consumption is more influenced by the DS flowrate. At 2.5 L/min FS flowrate per module, specific energy consumption was reduced from 0.21 kWh/m^3 to 0.14 kWh/m^3 for the same range of DS flowrate per module. In comparison, the energy consumption decreased from 0.33 kWh/m^3 to 0.21 kWh/m^3 for 3 L/min FS flowrate per module. On the other hand, if DS flowrate per module remains the same, the specific energy consumption of the system increases with the increase in FS flowrate per module. Since the increase in FS flowrate per module marginally reduced the number of modules, the total FS flowrate increases with the increase in FS inlet flowrate per module. As a result the specific energy consumption also increases with the increase in FS inlet flowrate per module. For example, at 2 L/min, 2.5 L/min, and 3 L/min FS inlet flowrate per module the specific energy consumptions were 0.09 kWh/m^3, 0.15 kWh/m^{3,} and 0.23 kWh/m^3 when the DS flowrate per module was 0.63 L/min.

5.4.6 Optimum operating conditions

Section 5.3 and 5.5 explained the effects of FS inlet flowrate per module on the specific energy consumption and the total number of modules required to design an FO system using the Toyobo CTA hollow fiber module. A trade-off was found between the specific energy consumption and the number of modules. As the FS inlet flowrate per module increases the number of modules reduces. But the specific energy consumption increases with the increase in FS flowrate per module because of higher flowrate and increased pressure drop. Therefore to find out the optimum FS inlet flowrate per module, a simple cost analysis has been conducted by considering the costs of membrane modules and energy consumption. Although the total cost of production includes numerous other cost components (such as land cost, building cost, piping cost, maintenance cost, etc.), for the simplicity, only the cost of energy for pumping the solutions through the modules and the cost of membrane modules were considered. For the analysis, cost of hollow fiber FO membrane module and unit power consumption (kWh) were considered as US$ 1,000 and US$ 0.25 respectively (Kim, Phuntsho, et al. 2017). In addition, the module life was assumed to be 5 years (Valladares Linares et al. 2016).

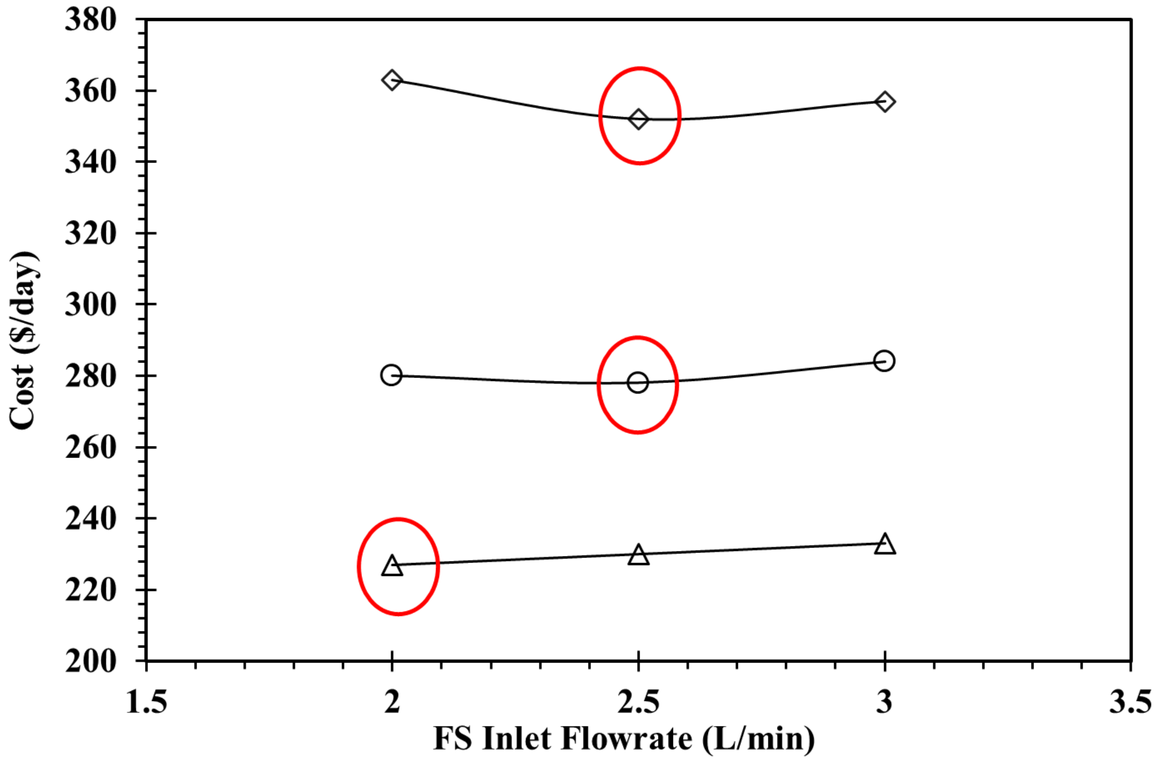


Figure 5.9. Comparison of the total cost of diluted DS production at different FS flowrate per module.

Employing these assumptions the cost of membrane element (C_M), the cost of energy (C_E) and the total total cost of production per day for a 1,000 m^3/day plant at different FS flowrates were plotted in Figure 5.9. Cost of membrane per day ($/day) is defined as,

$$C_M = \frac{C_{mod} \times N_M}{(L_{mod} \times 365)} \tag{5.28}$$

here, C_{mod} is the cost of module ($), N_M is the total number of membrane module required for the plant and L_{mod} stands for the life of the module (years). Similarly, the cost of energy consumption is calculated by,

$$C_E = SEC \times \lambda \times 24 \tag{5.29}$$

where SEC is the specific energy consumption (kWh/m^3) and λ represents the production capacity (m3/day). The cost analysis shows that the cost of membrane elements per day for 0.25 M desired final DS concentration reduced from US$ 203 to US$180 when the FS flowrate per module increased from 2 L/min to 3 L/min. However, for the same operating conditions, the cost of energy consumption per day increased from US$ 24 to US$ 53. Thus the total cost of production per day for 0.25 M final concentration at 2 L/min, 2.5 L/min, and 3 L/min FS inlet flowrate per module were US$ 227, US$230, and US$ 233. So the optimum FS flowrate per module for 0.25 M was considered as 2 L/min. Similarly, for 0.2 M and 0.15 M final DS concentration, 2.5 L/min FS inlet flowrate per module was found to be the optimum operating conditions. Since the higher FS flowrate reduces the final DS concentration of the system (as discussed in section 5.2), for the lower final concentration the optimum FS flowrate was found higher. Therefore, the DS inlet flowrate and the number of modules corresponding to the optimum FS inlet flowrate were considered as the optimum operating conditions in this study. Accordingly, the optimum DS flowrate per module is 0.8 L/min, 0.55 L/min and 0.32 L/min for 0.25 M, 0.2 M and 0.15 M respectively. Moreover, the optimum number of modules for the same final concentrations were found to be 370, 435, and 555.

5.5 Conclusions

This study developed mathematical models for the system design and optimization of a full scale FO plant employing a hollow fiber membrane module. Accuracy of the developed theoretical models was validated with the pilot scale experimental data using a CTA hollow fiber FO module (HPC3205) manufactured by Toyobo Co. Ltd. Membrane modular arrangement and the ranges of operating conditions for the module were determined for the optimization. Finally, a design of a 1000 m^3/day FO plant was simulated and optimized by using the hollow fiber module to produce 0.25 M, 0.20 M and 0.15 M diluted seawater. Accordingly, the major outcomes of this study are listed below:

1. Less than 10% difference between simulation and experimental data supports the accuracy of the developed models.
2. For the commercial hollow fiber membrane element used for this study, single element membrane module was adequate to achieve the target final DS dilution and the gain from any additional element in series configuration was insignificant.
3. For a 1000 m^3/day FO plant using the CTA hollow fiber module:
    i. optimum FS inlet flowrate for 0.25 M, 0.2 M, and 0.15 M final DS concentration were found to be 2 L/min, 2.5 L/min, and 2.5 L/min respectively.
    ii. optimum number of modules were 370, 435, and 555 for 0.25 M, 0.20 M, and 0.15 M final concentration.
    iii. minimum DS inlet flowrates for the final concentrations were 0.8 L/min, 0.55 L/min, and 0.32 L/min.

CHAPTER 6

DESIGN AND OPTIMIZATION OF A FULL SCALE FORWARD OSMOSIS SYSTEM USING A PLATE AND FRAME MODULE

6 Design and optimization of a full scale forward osmosis system using a plate and frame module

6.1 Introduction

Ensuring access to clean water for all is one of the sustainable development goals set by the United Nations (Ait-Kadi 2016; Programme 2003). Membrane based technologies play a key role to achieve this goal by desalinating seawater and recovering freshwater from impaired water sources. However, the conventional membrane technologies (RO/NF) are pressure driven and susceptible to irreversible fouling that makes these processes cost intensive and unaffordable (Im et al. 2020; Siddiqui et al. 2018; Suwaileh et al. 2020). Therefore, an alternative technology is required for affordable water filtration.

Forward osmosis (FO) process has gained significant research attention as a promising alternative to pressure-driven membrane processes (Cath, Childress & Elimelech 2006; Johnson et al. 2018; Lambrechts & Sheldon 2019). As this process is osmotically driven, minimal hydraulic pressure is required which results in lower operational and maintenance costs and easily cleanable reversible fouling. Owing to the low pressure operation and less fouling propensity, FO processes have been widely studied for numerous applications such as wastewater treatment, desalination, diluted fertilizer production, and so on (Ibrar et al. 2020; Kook et al. 2018; Phuntsho et al. 2016; Volpin et al. 2018). Most of these studies primarily focused on investigating the mass transport performance of the FO membrane through lab-scale experimental setup which is required to understand the fundamentals of the process and the membrane performances (Akther et al. 2020; Lee & Ghaffour 2019; Tiraferri et al. 2013). Apart from the membrane characteristic, the performance of an FO plant for real scale applications is significantly dependent on its design and operating conditions (Kim et al. 2018; Phuntsho et al. 2017). Therefore, full scale FO system design and optimization are very important to develop an efficient and affordable membrane based water filtration system.

Few pilot scale experimental studies investigated the full scale FO system performance using different types of membrane elements (Im et al. 2016; Kim, Blandin, et al. 2017; Kim et al. 2015; Phuntsho et al. 2016). Although the spiral wound (SW), hollow fiber (HF) and plate and frame (PF) type FO membrane elements are commercially available, these studies mainly focused on analyzing the performance of SW element considering its relatively high packing density. However, compared to the winding flow configuration of SW element, feed solution (FS) and draw solution (DS) flow through the rectangular membrane channels of a PF element in a single direction. Consequently, the fouling effects and pressure drops are lower for this element due to the simpler configuration (Lee & Kim 2018). Therefore, the PF type membrane elements are worth studying for full scale FO system design. Song, M. et al., experimentally investigated the performance of a commercial PF type membrane element (PFO100, manufactured by Porifera Inc., USA) at various operating conditions (Song et al. 2018). In another pilot scale experimental study, the same PF elements were used to investigate the performance of the elements in terms of water flux and recovery rate as a function of membrane area, concentrations, and flowrates of the solutions (Lee & Kim 2018). However, these experimental results either showed a single element's performance at various operating conditions or few serially connected elements' performances at a very narrow range of operating conditions, as conducting experiments for many elements and numerous operating conditions are not feasible. Therefore, a theoretical study is required to design and optimize an FO plant over a wider range of operating conditions using PF type membrane element.

A limited number of theoretical studies investigated the FO performance using plate and frame type membrane elements. Most of these studies considered either co-current or counter-current flow for their analysis but the actual flow configuration of a commercial PF element is orthogonal/cross current (Banchik et al. 2016; Deshmukh et al. 2015; Mondal, Field & Wu 2017). Some of these studies explored the effects of various operating conditions on the performance of a single element, which did not provide information about the membrane modular arrangement to design a full scale FO plant (Gu et al. 2011a). Few of these studies estimated the total membrane area and the required number of membrane elements for a desired recovery rate (RR) at a specific solution flowrate (Deshmukh et al. 2015; Lee & Kim 2018). However, these studies did not consider the final DS concentration (FDC) as a design objective, although the purpose of an FO plant is to dilute the DS to a certain final concentration. Moreover, the optimum membrane area was obtained only by maximizing the RR only without considering the FDC requirement of the FO plant. Hence, a theoretical model is needed to evaluate the performance of a PF type element considering the actual flow configuration. An optimization algorithm should also be used to determine the optimum membrane modular arrangement to obtain the desired RR and FDC of an FO plant.

This main objective of this study is to develop theoretical models for plate and frame type membrane element to design and optimize a full scale FO system. Solution diffusion models and fluid mass conservation laws were coupled to develop the mathematical models. Performance of the element was numerically simulated considering the actual flow orientation and physical dimension. Simulation results were compared with the experimentally measured element scale performance for the validation of the models. An overall performance index presented in our previous study (Ali et al. 2018) was used to optimize the FO system design by determining the minimum DS inlet flowrate and optimum membrane modular arrangement for the desired system output. Moreover, this study included the models and the optimization algorithm for the PF element with an FO system analysis software (which was developed to optimize the FO plant using spiral wound element in our previous study (Ali et al. 2018)) and reported the optimum membrane modular configurations and DS inlet flowrates for a 1,000 m^3/day FO plant. Finally, a parametric study has been conducted to investigate the effects of different operating conditions and design objectives on the full scale design using the PF membrane element.

6.2 Operating parameters for full scale design and optimization

Operating conditions of an FO process significantly influence the performance of the system. Inlet concentration and flowrates of the draw and feed solution are the most important operating conditions of the system (Gu et al. 2011a; Kim et al. 2018; Phuntsho et al. 2014; Xue et al. 2016). However, the inlet concentrations generally do not change for a specific application (where the DS and FS are predefined). Therefore, this study analyzed the effects of flowrates on the full scale FO plant performance using a commercial PF type membrane element. Production capacity, final DS concentration (FDC), and recovery rate (RR) are considered as the most significant performance parameters of the system (Awad et al. 2019; Phuntsho et al. 2017). Production capacity is defined as the volumetric quantity of diluted draw solution (which is the product of the system) produced by the system in a day. In contrast, the final concentration of the draw solution indicates the quality of the product, which is the concentration of the draw solution outlet of the system after the dilution. Furthermore, the RR is another important parameter to design a system. It specifies the percentage of FS inlet flowrate permeates to the DS side across the membrane. Any FO system is designed to obtain the desired production capacity, FDC, and the overall RR of the system. Therefore, these performance parameters are termed as design objectives in this study.

The design objectives are usually attained by varying the operating conditions (draw solution and feed solution inlet flowrate) of the system (Ali et al. 2018). Despite these operating conditions improve some of the performance parameters, but also diminish some other performance parameters. For example, increasing the DS inlet flowrate improves the RR of the system but also increases the FDC (Phuntsho et al. 2017). Therefore, it is essential to optimize the operating conditions to design and operate a full scale system. Besides the operating conditions, some design parameters also influence the performance of the system. Membrane type, spacer plate type, flow orientations in the membrane channel are some major design parameters that affect the performance. However, these parameters cannot be varied while designing an FO system using a commercial membrane module as these modules have their own membrane type, spacer plate design, and flow configurations. But the PF type membrane elements can be connected serially in a module to increases the RR and to reduce the FDC. Even though increasing the number of elements in a module improves both quantity (RR) and quality (FDC) of the product, but more dilution of the DS resulted in a lower average flux of the module which eventually increases footprint of the system. Therefore, to design and optimize of a FO plant using a PF type element, DS inlet flowrate and the number of elements in a membrane module are optimized in the current study.

6.3 Theoretical investigation

Comprehensive mathematical models were developed to design and optimize an FO system using PF type membrane elements for real scale applications. These models were formulated by combining the membrane mass transport models and fluid mass conservation laws considering the actual flow directions and physical dimensions of the membrane elements. Numerical analysis techniques were applied to simulate the system scale FO performance. Finally, an optimization algorithm was employed to optimize the membrane modular configurations and operating conditions of the system.

6.3.1 Mass transport models

Water from feed solution transports through the semipermeable FO membrane to the draw solution as a result of the osmotic pressure difference between these two solutions. Therefore the mass transport of FO processes are functions of membrane properties and the osmotic pressure gradient (Lee & Ghaffour 2019; Phuntsho et al. 2014). However, the osmotic pressure difference across the membrane is affected by the internal concentration polarization (ICP) and external concentration polarization (ECP) effects which depend on the membrane support layer structural parameter and the feed channel mass transfer coefficient. Considering the local mass transfer coefficients in the feed channel and membrane properties, water flux and reverse salt flux at any location over the membrane surface are calculated by equation 3.1 and 3.2 discussed in Chapter 3,

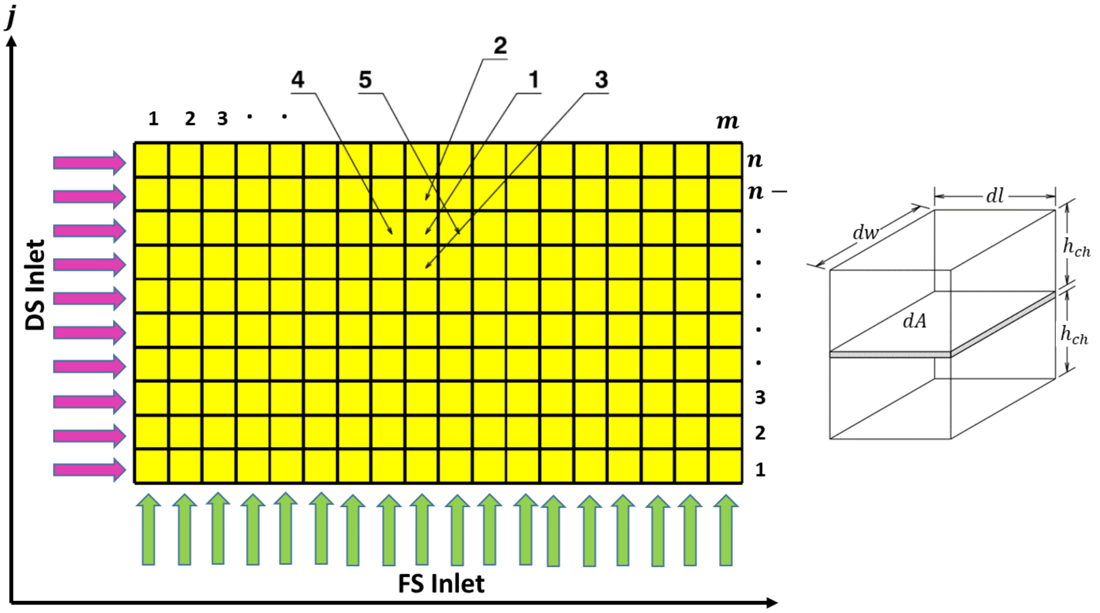
6.3.2 Numerical performance analysis

Performance of each PF type membrane elements connected serially in a module was theoretically investigated to design a full scale FO plant. However, the PF elements have several membrane plate cells stack on each other, where the membrane cells consist of a feed channel and a draw channel formed by the membrane sheets and the spacer plates. This study assumed that the performance of each membrane sheet is identical and therefore it analyzed the performance of one membrane sheet and determined the element scale performance from it. But the performance of the membrane sheet cannot be directly estimated by using the inlet conditions of the feed and draw solution. Due to the permeation of water and salt through the membrane, the spatial solution velocity, and the concentration in the membrane channels are significantly different from its inlet conditions. As a result, the local water flux and reverse salt flux over the whole membrane area also vary according to the direction of solution flow. Therefore to precisely estimate the performance of the membrane element, the whole membrane surface was divided into very small $(dl \times dw)$ discrete sections as shown in Figure 6.1. Solution diffusion models were employed to calculate the flux values at the inlet of the feed and draw channel using the known values of the DS and FS inlet velocity and concentration. Fluid mass conservation law was then applied to update the flow velocities and concentrations of the adjacent membrane elements considering these flux values and the direction of solution flows. Similarly, the local velocity, concentration, water flux, and reverse salt flux were iteratively calculated at each discrete membrane section shown in Figure 6.1.

According to the description of the PF type membrane element (in section 6.2), DS flows along the length direction and FS flows through the width direction over the membrane surface. Numerical models were developed to determine the local values of flowrate, velocity, and concentration assuming that these parameters only change in the dominant direction of the flow. Thus, DS flowrate at any location considering the water flux and flow direction is expressed by,

$$Q_{DS}(i+1,j) = Q_{DS}(i,j) + J_w(i,j) \times (dl \times dw) \tag{6.1}$$

here, i and j represent the indices for the location of the discrete membrane sections as shown in Figure 6.1. i indicates the distance of any section from the channel inlet in the length direction, whereas j mentions the distance in the width direction. $J_w(i,j)$ is the water flux that permeates from the FS to DS through the discrete section at a location (i,j). $dl \times dw$ is the area of the section. The amount of salt permeated through the section was so small compared to the volume of DS that in this study it was considered negligible.



Discrete location of highlighted points:- 1: (i,j); 2: (i,j+1); 3: (i,j-1); 4: (i-1,j); 5: (i+1, j)

Figure 6.1. Discretised membrane sheet for numerical simulation.

In comparison, the FS flowrate at any location that flows orthogonally with the DS is given by,

$$Q_{FS}(i,j+1) = Q_{FS}(i,j) - J_w(i,j) \times (dl \times dw) \tag{6.2}$$

FS and DS flow velocities at any discrete location over the membrane surface were calculated by applying the fluid continuity theorem as follows,

$$U_{FS}(i,j) = \frac{Q_{FS}(i,j)}{\varepsilon h_{ch} dl} \tag{6.3}$$

$$U_{DS}(i,j) = \frac{Q_{DS}(i,j)}{\varepsilon h_{ch} dw} \tag{6.4}$$

here, $U_{FS}(i,j)$ and $U_{DS}(i,j)$ are the local feed solution and draw solution velocity at any section (i,j). $\varepsilon h_{ch} dl$ stands for the effective cross-sectional area of the FS flow over the discrete section where ε is the porosity of the channel, h_{ch} is the channel height and dl is the length of the element as shown in Figure 6.1. Physical dimensions of the membrane channel are listed in Table A3.1 of the Appendix. Similarly, $\varepsilon h_{ch} dw$ is the actual crossectional area for the DS flow through the section. Mass transfer coefficient of the FS flow was calculated by using the velocity of the feed solution and the dimension of the channel (Phuntsho et al. 2014).

Apart from the fluid velocity, local FS and DS concentrations were also calculated to determine the local flux values. Solution concentration in this study was calculated by estimating the amount of solute flows through any specific location within a specific interval of time. The flowrate of feed solute at any location is estimated by,

$$Q_{FS,slt}(i,j+1) = Q_{FS}(i,j)C_{FS}(i,j) + J_s(i,j) \times (dl \times dw) \tag{6.5}$$

here, $C_{FS}(i,j)$ is the FS concentration and $J_s(i,j)$ represents the reverse salt flux that permeates from DS to the FS across the discrete location (i,j). Similarly, the draw solute flowrate was calculated by,

$$Q_{DS,slt}(i+1,j) = Q_{DS}(i,j)C_{DS}(i,j) - J_s(i,j) \times (dl \times dw) \tag{6.6}$$

where, $C_{DS}(i,j)$ is the draw solution concentration at a specific location (i,j). Considering the flowrates of solution and solute, FS and DS concentration at any location were given by,

$$C_{FS}(i,j) = \frac{Q_{FS,slt}(i,j)}{Q_{FS}(i,j)} \tag{6.7}$$

$$C_{DS}(i,j) = \frac{Q_{DS,slt}(i,j)}{Q_{DS}(i,j)} \tag{6.8}$$

Since the FS and DS flow orthogonally to each other in the PF type membrane element, the solution concentration and velocity change in both length and width direction. Consequently, the water flux and reverse salt flux also vary in both directions at any location. Therefore, the net permeate flowrate of any PF element connected in a module is expressed by adding all permeate flowrates at each discrete location,

$$Q_{net,perm} = N_{sheet} \sum_{i=1}^{m} \sum_{j=1}^{n} J_w(i,j) dl dw \tag{6.9}$$

where, $N_{sheet}$is the total number of membrane sheets in a plate and frame type membrane element, m and n are the number of discrete sections in the length and width direction. Since the flux is not uniform over the membrane surface, performance of the membrane element is expressed in terms of average flux, which is given by,

$$J_{w,avg} = \frac{Q_{net,perm}}{A_m} \tag{6.10}$$

A_m represents the total surface area of a membrane element here. Diluted DS is the product of an FO system. Therefore, the performance of the element is also reported in terms of draw solution outlet flowrate, which is expressed by,

$$Q_{DS,out} = N_{sheet} \sum_{j=1}^{n} Q_{DS}(m,j) \tag{6.11}$$

Diluted draw solution from a membrane element is collected from the draw channel outlet (where i = m) and is calculated by adding the draw solution flowrates through each discrete section in the width direction (j = 1 to n) as shown in Figure 6.1. Similarly, the feed solution outlet of a plate and frame type membrane element was calculated by,

$$Q_{FS,out} = N_{sheet} \sum_{i=1}^{m} Q_{Fs}(i,n) \tag{6.12}$$

However, the draw solution outlet concentration represents the quality of the product from the FO system, which is given by,

$$C_{DS,out} = \frac{N_{sheet} \sum_{j=1}^{n} Q_{DS,slt}(m,j)}{Q_{DS,out}} \tag{6.13}$$

Draw solution outlet concentration was calculated by estimating the ratio of the total amount draw solute and draw solution leaves membrane element (where i = m) in the same period. FS outlet concentration was also calculated similarly,

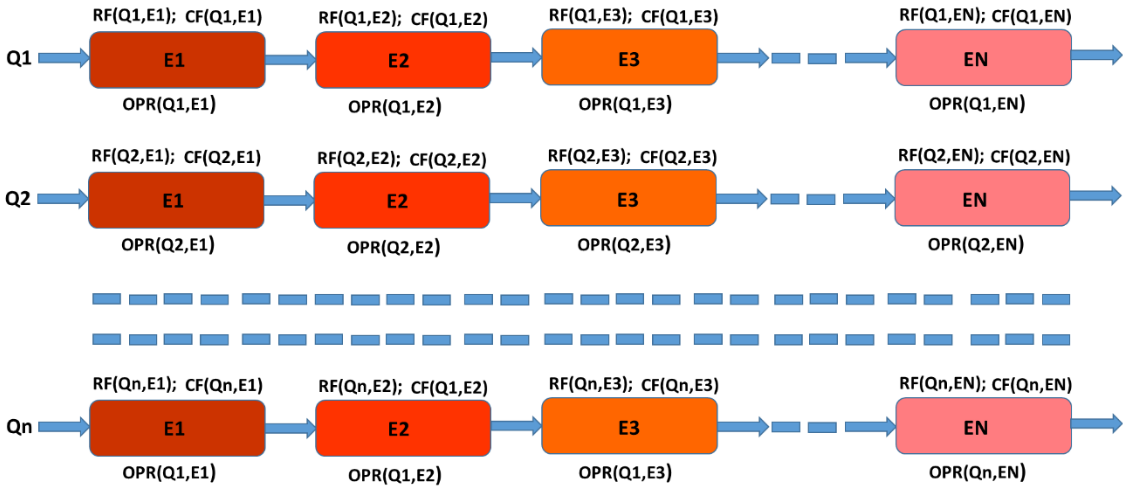
$$C_{FS,out} = \frac{N_{sheet} \sum_{i=1}^{m} Q_{FS,salt}(i,n)}{Q_{FS,out}} \tag{6.14}$$

Finally, the net water permeation rate of the system as a percentage of feed solution inlet flowrate is denoted as recovery rate, which is given by,

$$RR = \left[\frac{Q_{net,perm}}{Q_{FS,in}}\right] \times 100 \tag{6.15}$$

6.3.3 Optimization of FO system

This section summaries the optimization procedure employed in this study to design a full scale FO plant using plate and frame type membrane module. In order to design an FO plant for a desired RR (at any specified FS inlet flowrate) and FDC, DS inlet flowrate and the number of elements connected in a module can be varied. However, increasing the DS inlet flowrate increases the RR but it also increases the final concentration of the draw solution.



Values of Q and E corresponding to the maximum OPR value are the optimum conditions

Figure 6.2. Optimization procedure employed for full FO system design and operation. Here Q is the draw solution inlet flowrate and E is the number of elements serially connected in a plate and frame type membrane module to obtain the desired performance. For each draw solution inlet flow rate (Q) and the number of elements (E), overall performance parameter OPR is calculated. Operating conditions for the maximum OPR is considered as optimum**.**

On the other hand, by adding more elements in the module RR of the system increases as well as the FDC reduces, but it increases the footprint of the system. Therefore, this study employed an optimization algorithm that used an overall performance parameter considering the RR, FDC, and number of elements per module. The optimization method is schematically depicted in Figure 6.2. This optimization method begins by selecting a range of DS inlet flowrate. The flowrate was increased from the first value of the range Q_1 to the last value of the range Q_n with a very small increment of ΔQ. For each draw solution inlet flowrate, the number of elements in a module was increased from 1. Performance of the FO system after each element was theoretically evaluated in terms RR and DS outlet concentration. More elements were added (from 1 to N elements) until both desired RR and FDC were achieved. However, for every condition, overall performance parameter OPR was calculated. OPR is the percentage of overall design target (RR and FDC) obtained by adding each element. It can be expressed as,

$$OPR(Q,E) = \left[\frac{RF(Q,E) \times CF(Q,E)}{N_E}\right] \times 100 \tag{6.16}$$

here, $RF(Q,E)$ and $CF(Q,E)$ are the recovery fraction and concentration fraction for any DS flowrate and elements. N_E is the number of elements connected in a module. However, the recovery fraction is given by,

$$RF(Q,E) = \frac{RR(Q,E)}{RR_D} \tag{6.17}$$

where, $RR(Q,E)$ is the actual RR achieved after any specific number of element (E) using any DS inlet flowrate (Q). RR_D is the desired RR. Similarly, the concentration fraction is expressed by,

$$CF(Q,E) = \frac{C_{D,F}}{C_D(Q,E)} \tag{6.18}$$

here, $C_{D,F}$ is the desired final concentration and $C_D(Q,E)$ is the actual draw solution outlet concentration after any number of element (E) for any DS inlet flowrate (Q). The maximum value of recovery fraction (RF) and concentration fraction (CF) were assumed 1 (one) in this study to stop the optimization process from overdesigning the system. However, since the higher values of both RF and CF contribute to improving the FO system performance, these two fractions are needed to be maximized. In contrast, for a smaller footprint of the system the number of elements (N_E) is to be minimized. Therefore, the OPR value increases with the increase in RF and CF, but it decreases with the increase in the number of elements according to equation (6.18). OPR values for all conditions [OPR(Q1,E1) to OPR(Qn,EN)] were saved and compared. DS inlet flowrate and the number of PF elements per module which showed the maximum OPR value were considered as the optimum design and operating conditions.

In this optimization method, to prevent the algorithm from selecting an optimum condition that provides very low RR or final concentration, the search domain is constrained. The constrain is applied by assuming,

$OPR(Q,E) = 0$, when $RF(Q,E) \leq (1 - fl/100)$ or $CF(Q,E) \leq (1 - fl/100)$.

here, fl is the acceptable percentage deviation of actual RR or FDC from the desired value which is termed as flexibility. For example, if 5% flexibility is allowed OPR is 0 for any RF or CF value lower than 0.95.

6.4 Results and discussion

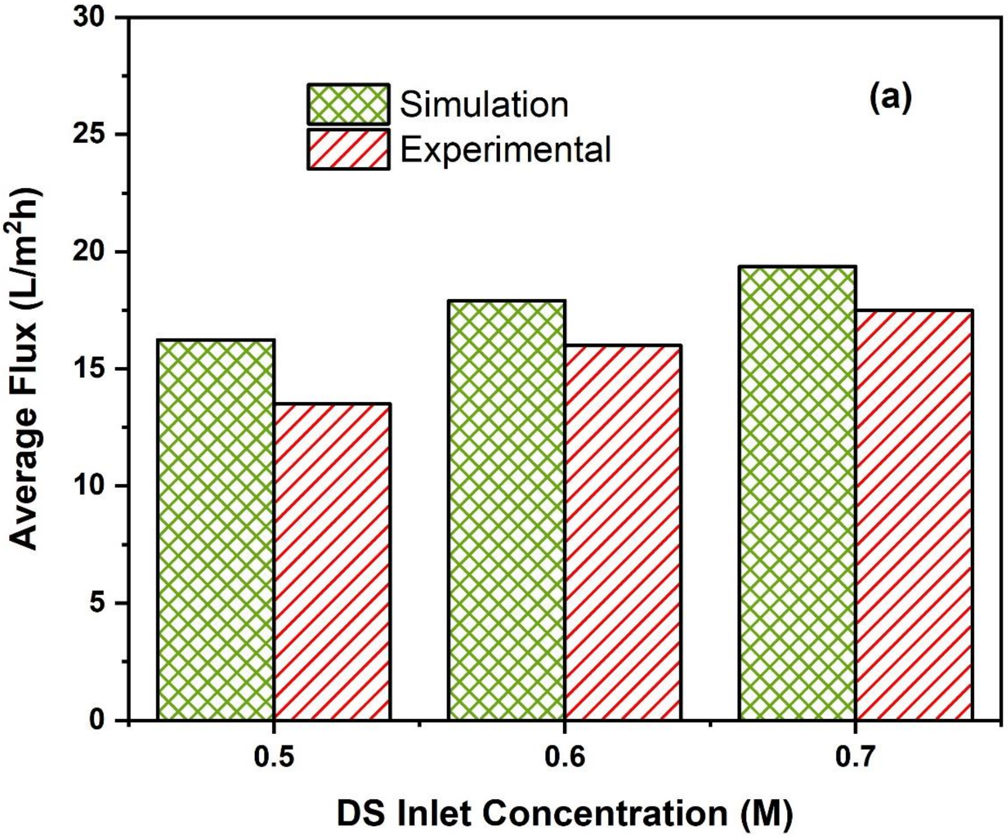
Performance of a full scale FO plant with PF type membrane module was numerically simulated and optimized based on different operating conditions. The simulation results were validated by comparing with the published experimental results from the PF membrane elements. Parameters used for the numerical simulation are furnished in Table 6.1. Moreover, the element scale performance was employed to optimize the FO plant for different recovery rates (at different feed solution inlet flowrates) and final DS concentrations. Finally, the effects of different design objectives and operating conditions on FO plant design using the plate frame elements were analyzed by the software.

Table 6.1 Operating conditions and design objectives considered for the simulation

Operating Conditions		**Design Objectives**	
Feed solution	0.02 M NaCl	Production Capacity	1,000 m^3/day
FS Osmotic pressure	0.99 bar (Similar to MBR effluent)	Recovery rate (RR)	40 – 60%
FS Inlet Flowrate	10 – 40 L/min	Final DS Concentration (FDC)	0.25 – 0.15 M NaCl
Superficial Velocity	0.03 - 0.11 m/s		
Draw solution	0.6 M NaCl		
DS Osmotic pressure	27.8 bar (Similar to seawater)		
DS Inlet Flowrate	0.5 – 20 L/min		
Superficial Velocity	0.008 – 0.04 m/s		

6.4.1 Validation of the mathematical models

The mathematical models developed in this study were validated with the published performance results of a commercial plate and frame type membrane element (PFO 100, manufactured by Porifera Inc., USA) reported in the literature. Performance of the PF element was theoretically measured as a function of DS inlet concentration when the DS and FS inlet flowrate were 18 L/min and 8 L/min as shown in Figure 6.3(a). DI water was considered as the FS for the validation. According to the simulation results, the average water flux of the element increased from 16 L/m^{2}h to 19 L/m^{2}h, when the draw solution inlet concentration was increased from 0.5 M to 0.7 M. In comparison, a pilot scale experimental study using the same PF membrane element found that at the same operating conditions the average flux of the elements varied from 14 L/m^{2}h to 17.5 L/m^{2}h, when the draw solution concentration increased from 0.5 M to 0.7 M (Song et al. 2018). Both experimental and theoretical results showed almost similar trends of flux behavior as the DS inlet concentrations were varied.



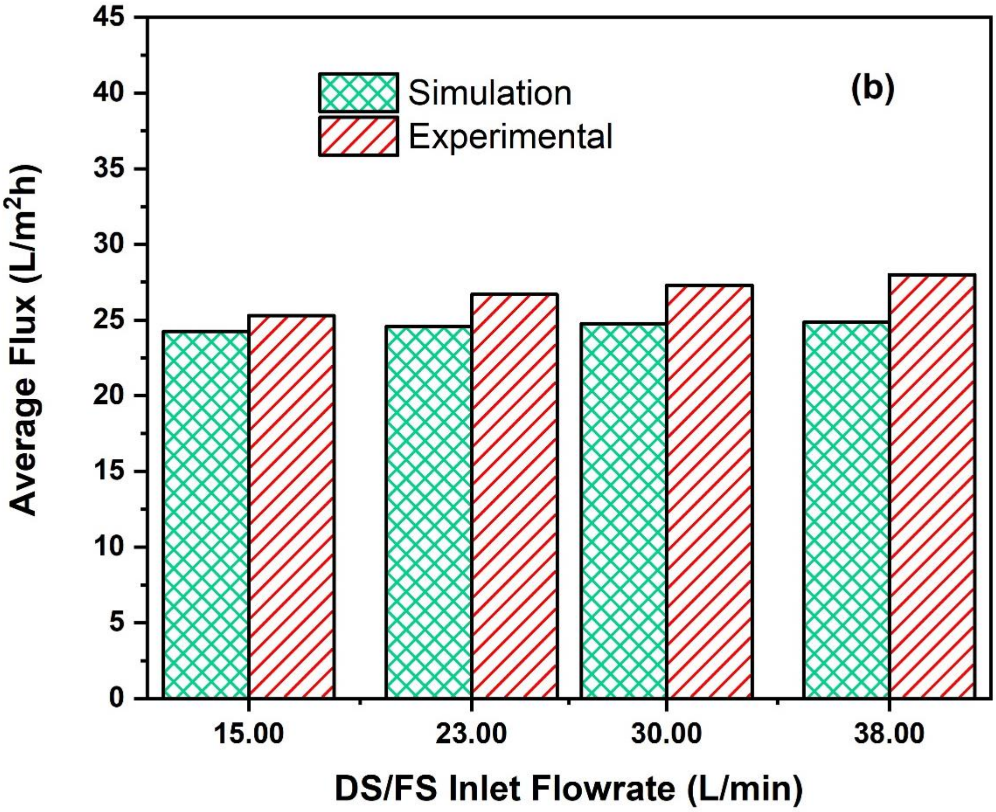


Figure 6.3. Comparison of the simulated element scale performance with the experimental results at (a) different DS inlet concentrations with 8 L/min DS and 18 L/min FS flow rates and, (b) at different inlet DS or FS flowrates (but same flow rates for 1 M NaCl as DS and DI water as FS).

In addition, an average 11% difference between the experimental and simulation results exhibited a very good agreement between them. Apart from the DS inlet concentration, the membrane element's performance was also compared as a function of DS and FS inlet flowrates, when their inlet concentrations remained unchanged at 1 M and 0 M, respectively. Simulation results at these operating conditions were compared with the manufacturer provided experimentally measured performance data as shown in Figure 6.3(b) (Benton & Bakajin 2017). Theoretical results showed that when the DS and FS flowrates increased from 15 L/min to 38 L/min, the average water flux increased only slightly from 24 L/min to 25 L/min. Similarly, the experimental water fluxes for the same operating conditions also increased only marginally from 25 L/min to 28 L/min. This time, however, the experimental water fluxes were slightly higher than the theoretical water fluxes in contrast to the observations in Figure 6.3(a) although still within a similar average deviation of 9% indicating the reliability of the simulation model.

6.4.2 Optimization of full scale FO system

This section covers the optimization of a full scale FO plant using the PF type membrane element to design a FO plant for the osmotic dilution of seawater. Hence, this study considered 0.6 M NaCl solution as DS which has a similar osmotic pressure to the seawater. Similarly, 0.02 M NaCl was chosen as FS in this study as it has the same osmotic pressure of secondary MBR effluent. Figure 6.4 explains the optimization of membrane modular configurations of an FO plant which is operated at 50% RR, 30 L/min FS inlet flowrate, and produces osmotically diluted seawater at 0.2 M. The FDC and RR after each element connected serially in a module are shown in Figure 6.4 for 4 L/min, 5 L/min, and 6 L/min DS inlet flowrates. Considering the flexibility of 5%, for any RR less than 47.5% or any FDC greater than 0.21 M, the recovery fraction (RF) and the concentration fraction (CF) was considered 0. In addition, when the FDC was lower than the 0.2 M or the RR was greater than 50%, CF and RF were considered 1 to protect the overdesign of the system. Figure 6.4 does not show the performance values after the 9th element for 5 L/min and 6 L/min DS inlet flowrate as both desired RR and FDC were reached by the 8th element for these flowrates.

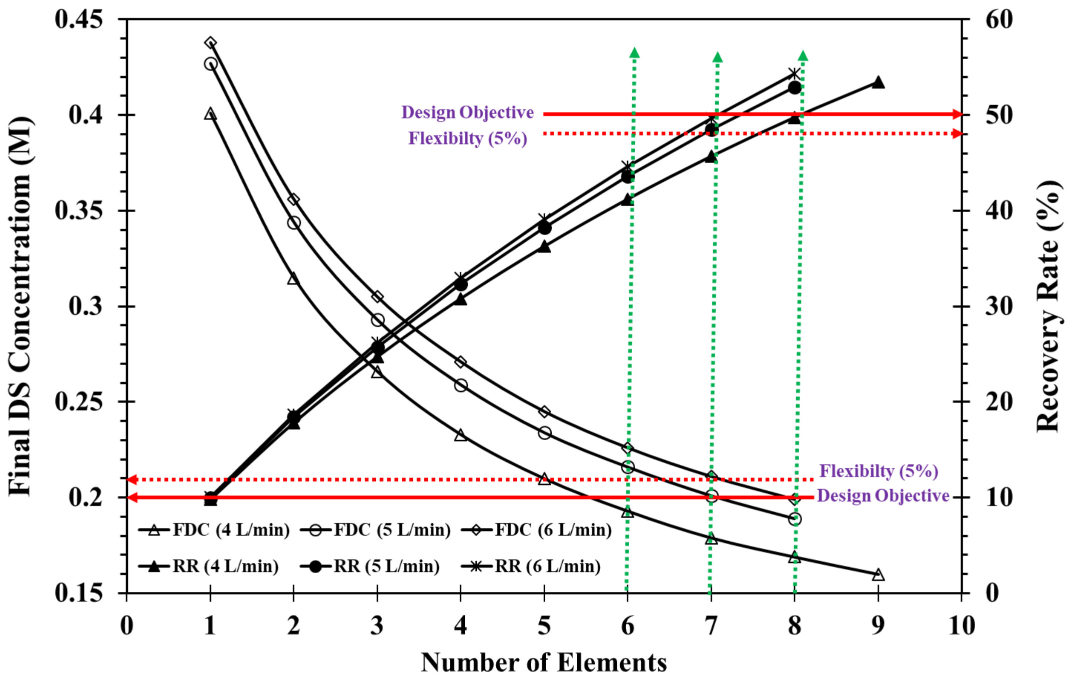


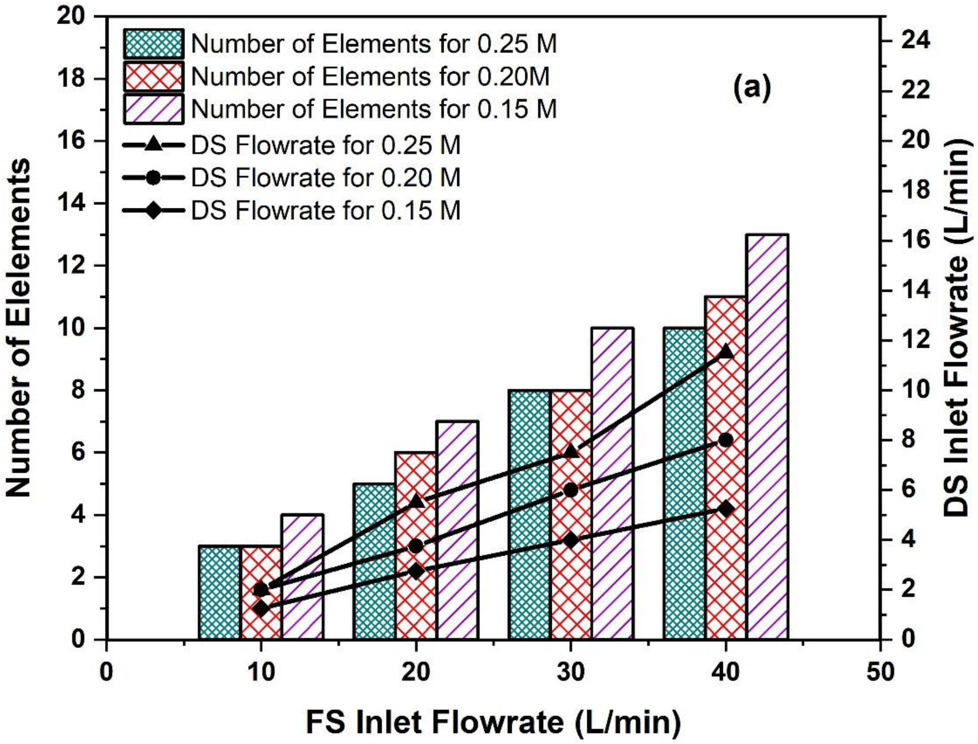
Figure 6.4. Optimization of design and operating conditions. DS inlet flowrate was varied from 4 to 6 L/min to determine the recovery fraction (RF), concentration (CF), and overall performance rating (OPR) for different number of elements per module.

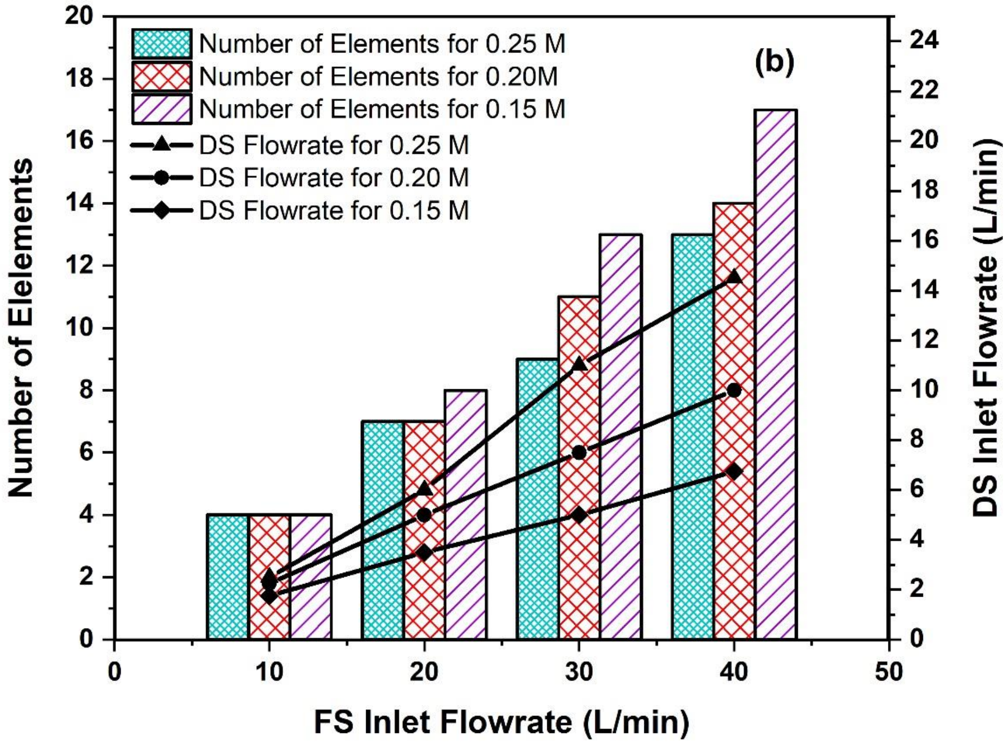
Recovery fraction, concentration fraction, and OPR were calculated for all DS inlet flowrates and the number of elements. It can be seen from the figure that at 4 L/min DS inlet flowrate, RR of the system exceeds 47.5% after the 8th element. RR for the 8th element and 9th elements are 49.76% and 53.45%. Therefore, RF for the 8th element and 9th elements were found to be 0.99 and 1. On the other hand, FDC of the system drops below 0.21 M after the 5th element when the DS inlet flowrate is 4 L/min. FDC for the 5th, 6th, 7th, 8th, and 9th elements are 0.21 M, 0.193 M, 0.179 M, 0.169 M, and 0.16 M respectively. CF for these elements were 0.95, 1, 1, 1, and 1 respectively. Considering RF, CF and the number of elements, OPR for 8th and 9th elements were 12.38 and 11.11. Therefore, maximum OPR for 4 L/min DS flowrate was found to be 12.38 when 8 elements were used. Similarly, for 5 L/min DS inlet flowrate, the highest OPR was 13.72

and the corresponding number of elements was 7. In contrast, the optimum number of elements and OPR were 8 and 12.50, respectively for 6 L/min DS flowrate. So, the maximum OPR value of 13.72 was observed for 5 L/min DS inlet flowrate with 7 elements per module. Therefore, to design a FO plant with 50% RR (30 L/min FS inlet flowrate) and to produce diluted seawater at 0.2 M final concentration, 5 L/min DS inlet flowrate and 7 PF membrane elements (PFO100) per module are the optimum design and operating conditions. However, to design a real scale plant for a certain production capacity the total number of membrane elements and modules are also needed to be calculated which is discussed in the next section.

6.4.3 Effects of design objectives and operating conditions on full scale design

This section explains the effects of design objectives and operating conditions on the design of an FO plant using PF type membrane element for real scale application. It shows the changes in optimum DS inlet flowrate, number of elements per module with the changes in desired FDC, and RR at various FS inlet flowrates. Figure 6.5(a) describes the optimum conditions for 40% RR, when the FS inlet flowrate is increased from 10 to 40 L/min and the FDC varied from 0.25 M to 0.15 M. It can be seen from the figure that for the same RR at any FDC, the FO plant requires larger DS flowrate and more PF elements when the system was operated at a higher FS inlet flowrate. Since more water permeation is required at higher FS inlet flowrate for the same RR, higher DS inlet flowrate is required to increase the net permeation. However, higher DS inlet flowrate increases the FDC also. As a result, to improve the water permeation and dilution of DS more elements are required to be connected in a module.


(a)
Number of Elements for 0.25 M
Number of Elements for 0.20M
Number of Elements for 0.15 M
DS Flowrate for 0.25 M
DS Flowrate for 0.20 M
DS Flowrate for 0.15 M
Number of Elelements
DS Inlet Flowrate (L/min)
FS Inlet Flowrate (L/min)
0
2
4
6
8
10
12
14
16
18
20
22
24
0
10
20
30
40
50


(b)
Number of Elements for 0.25 M
Number of Elements for 0.20M
Number of Elements for 0.15 M
DS Flowrate for 0.25 M
DS Flowrate for 0.20 M
DS Flowrate for 0.15 M
Number of Elelements
DS Inlet Flowrate (L/min)
FS Inlet Flowrate (L/min)
0
2
4
6
8
10
12
14
16
18
20
22
24
0
10
20
30
40
50

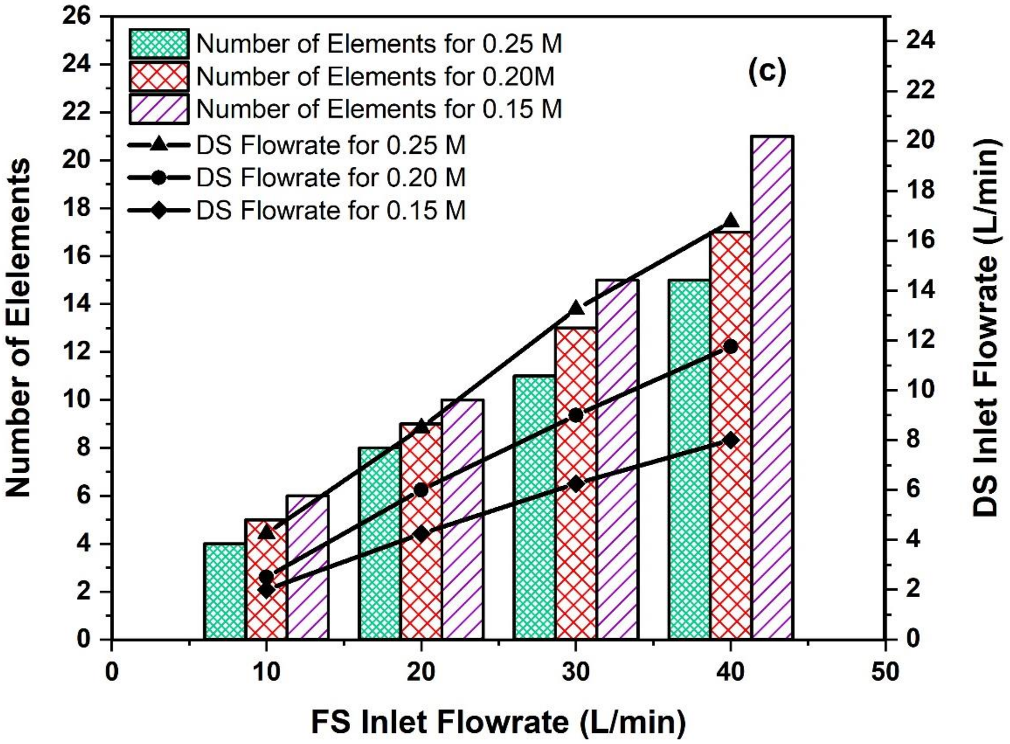


Figure 6.5. Full scale FO plant design and operating conditions for (a) 40%, (b) 50% and (c) 60% RR at different FDC.

For example, to recover 40% of freshwater at 0.25 M final concentration using the PFO100 element, the required DS inlet flowrate increased from 2 L/min to 11.5 L/min, when FS inlet flowrate is varied from 10 L/min to 40 L/min. Further, to obtain a more diluted DS from the FO system at the same RR and any FS inlet flowrate it was found that the system is required to operate at a lower DS inlet flowrate but more elements were required to be connected. It was found that for 40% RR and 20 L/min FS inlet flowrate to produce diluted seawater at 0.25 M, 0.2 M, and 0.15 M, the required DS inlet flowrates were 5.5 L/min, 3.75 L/min, and 2.75 L/min respectively. Since the lower DS inlet flowrate contributes to produce a more diluted solution, the DS flowrate was lowered when the less FDC was required. However, reducing the DS inlet flowrate also reduces the water permeation because it reduces the inlet draw solute mass flow rate (Phuntsho et al. 2014). Therefore to maintain the required RR more elements were connected in the module. For the same RR and FS inlet flowrate, the number of elements per module was increased from 5 to 7, when the FDC requirement changed from 0.25 M to 0.15 M. Likewise, Figure 6.5(b) and Figure 6.5(c) also show the optimum parameters for 50% and 60% RR. These figures show that for the same FDC and FS inlet flowrate, to increase the RR of the system both DS inlet flowrate and number of elements are needed to be increased. As an example, for 20 L/min FS inlet flowrate and target FDC of 0.2 M, to improve the RR of the system from 40% to 60% the optimum DS inlet flowrate has to be increased from 3.75 L/min to 6 L/min and number of elements to be increased from 6 to 9.

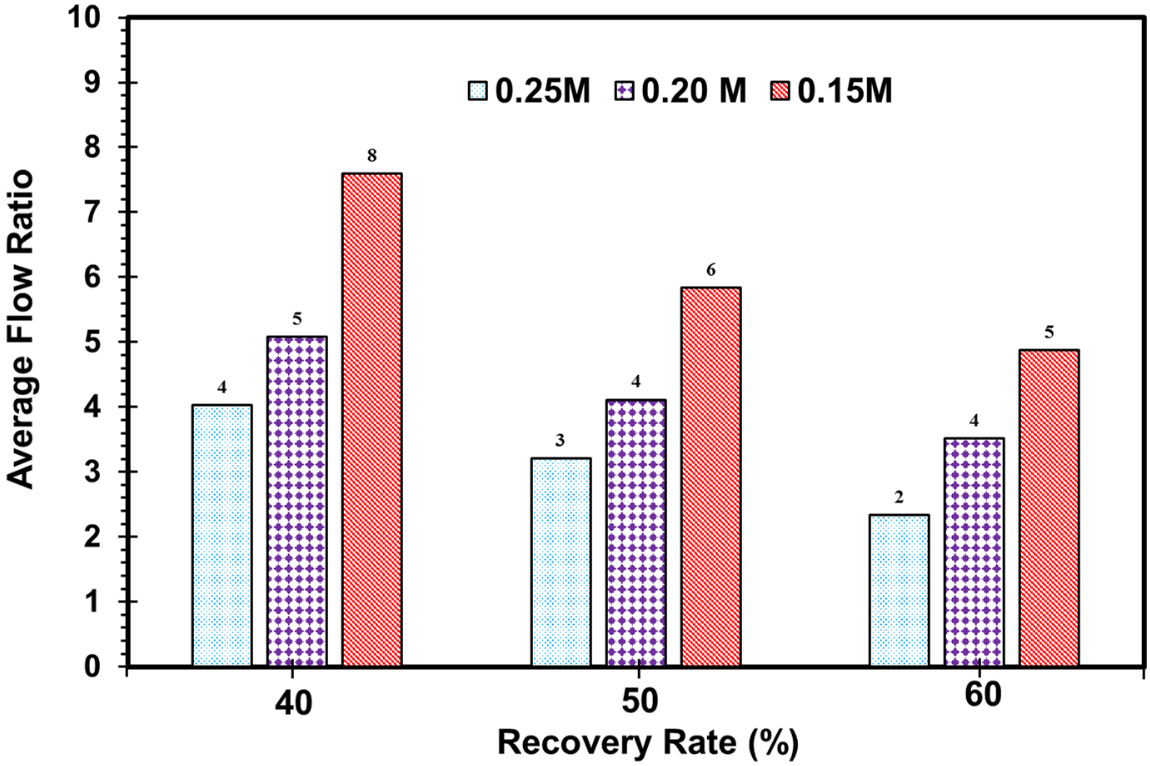


Figure 6.6. Average flow ratios for different design objectives using the plate and frame type membrane element.

Finally from the trends of DS inlet flowrate at different FS inlet flowrate, average flow ratios for different RR and FDC were calculated. Average flow ratio is defined as the ratio of the feed and draw solution inlet flowrate in this study. Figure 6.6 shows the average flow ratios for 40%, 50%, and 60% RR for the FDC of 0.25 M, 0.2 M, and 0.15 M. It can be seen from the figure that the average flow ratio has to be lower for achieving a higher overall RR of the system for any FDC. On the other hand, the average flow ratio has to be increased to achieve a lower FDC. For 0.15 M FDC, the required average flow ratio of the system reduced from 8 to 5, when the recovery rate increased from 40% to 60%. In contrast, for 50% RR the average flow ratio increased from 3 to 6, when the FDC requirement changed from 0.25 M to 0.15 M.

6.5 Conclusions

Based on the theoretical models, performance of a full scale FO system is estimated which was validated and then applied for the optimization of a commercial full-scale plate and frame FO membrane module under various design considerations. An overall performance indicator considering the recovery rate, final DS concentration, and the number of elements was used to determine the optimum DS inlet flowrate and membrane modular configurations. The optimization algorithm was then applied to design and optimize a 1,000 m^3/day FO plant using a commercial PF type membrane element for a desired recovery rate and final DS concentration. Finally, the effects of design objectives and operating conditions on the FO plant design for real scale applications were investigated. Some major findings of this study are listed below:

i. An average $\approx$ 10% difference between the theoretically calculated and experimentally measured performance of the membrane element were observed which shows very good reliability of the developed mathematical models.

ii. Full scale design of a 1,000 m^3/day FO plant to produce 0.2 M final diluted draw solution at 50% recovery rate (20 L/min feed solution inlet flowrate to each module) by using 0.6 M NaCl solution (similar to seawater) as DS and 0.02 M NaCl (similar to secondary effluent) as DS includes:
    a. Total membrane modules: 47.
    b. Number of PFO 100 elements connected serially in a module: 7.
    c. Total membrane elements: 329.
    d. Draw solution inlet flowrate to each module: 5 L/min.

iii. For 50% recovery rate feed solution flowrate was about 3, 4, and 6 times higher than the draw solution inlet flowrate for the plate and frame module when the final draw solution concentrations were 0.25 M, 0.2 M, and 0.15 M.

CHAPTER 7

DEVELOPMENT OF A FORWARD OSMOSIS SYSTEM ANALYSIS (FOSA) SOFTWARE FOR FULL SCALE SYSTEM DESIGN AND OPTIMIZATION

7 Development of a forward osmosis system analysis software for full scale system design and optimization

7.1 Introduction

The global demand for freshwater is increasing at more than twice the rate of the population growth in the last 10 decades (Watkins 2006b). Apart from the population growth, severe climate change, increasing agricultural activities, economic development also contributed to the increase in freshwater demand water demand (Ali et al. 2016; Beddington 2011; Suwaileh et al. 2020). To mitigate this problem freshwater is required to recover from unconventional sources such as seawater, brackish water, wastewater, etc. using an energy efficient and cost effective process. Pressure driven membrane based technologies (RO/NF) are the most commonly used processes for these applications (Ali & Chakraborty 2016; Peñate & García-Rodríguez 2012; Qasim et al. 2019). These processes are thermodynamically non-spontaneous and require a transmembrane pressure difference to drive the mass transport through the membrane exceeding the osmotic pressure of the feed solution. Consequently, the high osmotic pressure of the feed solution results in a limited recovery rate and large hydraulic pressure requirement for these processes (Chong, Wong & Fane 2008; Wang et al. 2018). Therefore, these processes are cost and energy intensive. Addressing these drawbacks of the conventional processes, recently many researches focused on applying FO process for a wide range of potential water filtration applications. It is a thermodynamically spontaneous membrane process driven by the osmotic pressure gradient of two solutions (draw solution (DS) and feed solution (FS)) with dissimilar concentrations (Akther et al. 2015; Ali et al. 2018; Cath, Childress & Elimelech 2006). Although the stand-alone FO process does not produce freshwater directly, it osmotically dilutes the feed solution of the pressure driven processes (which is used as the DS of the FO process) using a less concentrated impaired water as feed solution (Valladares Linares et al. 2014). Subsequently, freshwater is produced by the pressure driven processes applying much lower hydraulic pressure and at a higher recovery rate. Therefore the FO processes significantly improve the overall energy efficiency of the water filtration processes when the regeneration of draw solution is not required (Im et al. 2020; Phuntsho et al. 2016; Seo et al. 2019).

Despite the promising attributes of FO processes, the full scale implementation of the FO system for commercial applications are very limited (Awad et al. 2019; Shaffer et al. 2015). First full scale application of FO process was reported in 2002 by Hydration Technology Innovations (HTI). A sugar based DS was used for this project to recover freshwater from the wastewater produced during the hydraulic fracturing (Hutchings, Appleton & McGinnis 2010). Two 200 m^3/day FO/RO hybrid system has been implemented in the middle east by 2010 (Thompson & Nicoll 2011). Moreover, Toyobo and Aquaporin commercialized hollow fiber module for FO processes, which have higher packing densities compared to the spiral wound FO module (Sengur-Tasdemir et al. 2018). Recently, Porifera Inc. (USA) developed plate and frame membrane module which is performing well with harsh wastewater (Benton & Bakajin 2017; Song et al. 2018). Although numerous commercial membrane modules are available in the market and their applications in some projects have already shown promising outcomes, this technology is still lacking in its full scale implementation. One of the main hurdles for the successful commercialization of this process is the shortage of information about the systematic way of designing and optimizing an FO plant.

Addressing these shortcoming numerical models were developed for the spiral wound

(SW), hollow fiber (HF) and plate and frame (PF) FO module considering their actual dimensions and directions of fluid flow. Simulations were conducted considering the flow directions and dimensions of a commercial TFC 8040 SW module, a CTA HF module, and a PF module [Chapter 4, 5 and 6]. To determine the optimum membrane modular configurations, a novel optimization algorithm was developed considering both recovery rate and final concentration. Number of membrane modules and optimum DS inlet flow rate per pressure vessel was theoretically calculated for various recovery rates and final DS concentrations (this study is discussed in chapter 4 of this dissertation). From these previous studies, it was found that designing and optimizing a full scale FO plant is a long and complicated process. Solution of numerous coupled, complex non-linear equations to design the FO system for various operating conditions requires in-depth fundamental knowledge of the process and often leads to the computational error. On the other hand, a system analysis software could make this design process very simple that may reduce the time and skill required to design the FO system efficiently. Moreover, the previous studies [chapter 4, 5 and 6] although designed the FO system employing a commercial spiral wound, hollow fiber and plate and frame module but did not compare their performance. A system analysis software could be used to compare the design of a large scale FO plant using various types of membrane modules for the same design targets and operating conditions. In addition, for the full RO system design, there are several commercial system analysis software. For example, ROSA, Toray Track, IMSDesign etc. However, for FO system design and optimization there is no software available so far. Therefore, the development of an FO system analysis software would play a vital role in the full scale implementation of the FO system.

The main aim of this study is to develop a system analysis software for the design and optimization of a full scale FO plant. The software is named as FOSA (forward osmosis system analysis) in this study. To develop the frameworks of the software, several comprehensive algorithms were designed. These algorithms were coded using MATLAB programming language to develop the frontend (graphical user interface) and backend (performance simulation and optimization) layers of the FOSA software. It employed the mathematical models of different types of membrane modules and optimization algorithms developed in our previous studies [chapter 4, 5 and 6 of this dissertation] to design and optimize an FO system for large scale applications. Finally, FOSA software was used to compare the design of a 1,000 m^3/day FO plant using a commercial TF8040 spiral wound module, a CTA hollow fiber module, and a plate and frame module to produce 0.2 M diluted DS. FS and DS for the design were considered as 0.6 M NaCl and 0.02 M NaCl solution since they have similar osmotic pressure of seawater and MBR effluent. However, to the best of authors' knowledge, this is the first systematic effort to develop an FO system analysis software to design and optimize a full scale plant.

7.2 Development of FO system analysis software

FO system analysis software (FOSA) aims to design and optimize a full scale FO system. So, it is important to describe a full scale FO system, its performance, and operating parameters before explaining the software development process. A full scale system consists of an array of membrane modules (which is schematically shown in Figure 3.2) according to the design objectives such as diluted DS production capacity, recovery rate, and final DS concentration of the system.

Performance of the FO system is also dependent on the operating conditions of the FO system. The major operating conditions of an FO system are FS inlet flowrate, FS inlet concentration, DS inlet flowrate and DS inlet concentration. Moreover, these operating conditions have both desirable and undesirable effects on system performance. Therefore, to design an FO system it is essential to optimize the operating conditions. So the FO system analysis (FOSA) software developed in this study determines the optimum number of elements and the DS inlet flowrate per pressure vessel to obtain the desired production capacity, final DS concentration, and overall recovery rate of the system at any feed solution inlet flowrate.

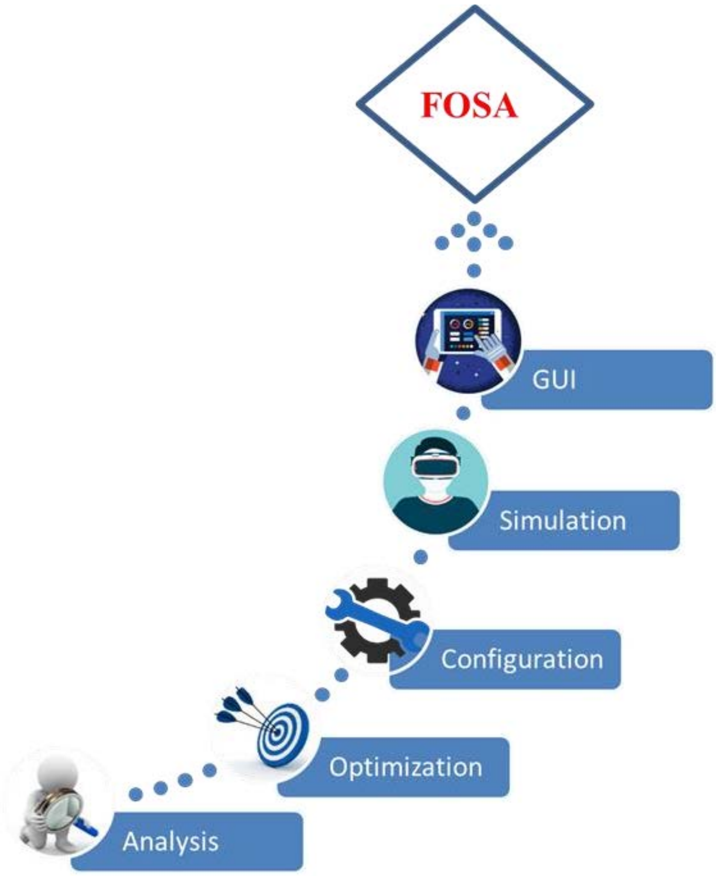


Figure 7.1. FO system analysis software development flow diagram.

Development of the FOSA software involves some sequential operations as shown in Figure 7.1. It begins with the analysis of the FS and DS properties. In order to determine the initial driving force of the process, osmotic pressure of the DS and FS are analyzed for a given FS and DS. After analyzing the solution properties, optimization of the design and operating conditions are carried out to determine the number of elements and DS inlet flowrate per pressure vessel for the desired recovery rate and final DS concentration. Then the total production capacity of the plant is defined in the configuration step. Employing the optimum operating conditions and total desired production capacity, total number of membrane modules, pressure vessels and membrane area are calculated in the simulation step as shown in Figure 7.1. All of these operations (from analysis to simulation) are carried out at the backend of the software which means that the user of the software does not visualize these operations. To run these backend processes, the software requires some input parameters from the user. For example, the compositions of FS and DS are required to analyze the solution. Further, the membrane module type, input operating conditions, and design objectives are required to be defined by the user for the optimization. Therefore, a graphical user interface (GUI) is developed, which acts as the frontend of the software to interact with the user. GUI of the software does not only receive input but also displays the calculated results for further analysis. MATLAB programming was used in this study to develop the user interface of the software and to numerically analyze the performance of the FO system considering its powerful mathematical library. Finally, the combined packages of the backend processes and the graphical user interface results in the development of the proposed forward osmosis system analysis (FOSA) software. For the development of the software, this study used a computer with the following configurations:

Processor: Intel(R) Core (TM) i5 – 6300 U CPU @ 2.4 GHz

RAM: 8 GB

Operating System: Windows 10, 64 bit

7.3 Models and algorithms for software

This study developed a system analysis software for FO system design and optimization. Theoretical models were developed to determine the information required to design a full scale FO system employing the module scale performance of different types of membrane elements. Module scale performances were calculated by employing the mathematical models developed in our previous studies for SW [Chapter 4], HF [Chapter 5] and PF [Chapter 6] type membrane elements. It also used an optimization algorithm developed in our previous study [Chapter 4] to determine optimum membrane modular configuration and operating conditions of the system. Finally, several algorithms were developed to design the general framework of the software. These algorithms were coded by using the MATLAB programming language to develop the software. MATLAB codes for the development of this software are provided in A4 section of the Apendix.

7.3.1 Mathematical models

The first step of the FO system design process is the analysis of the FS and DS properties. In this study, the properties of the solutions were calculated in terms of TDS (total dissolved solid) and osmotic pressure. TDS of the solution is estimated by adding the concentrations of the individual components of the solution as follows,

$$TDS = \sum_{i=1}^{n} C_E\,(i) \tag{7.1}$$

here, C_E is the concentration of any components of the solution, i represents the index of the component when the solution is comprised of total n components. Osmotic pressure of the multi-component feed and draw solution was calculated by an empirical relation provided by Hydraunitcs and Dow chemical, which is given by (DOW 2020),

$$\pi = 1.19(T + 273)\sum_{i=1}^{n} M_E\,(i) \tag{7.2}$$

here, T is the temperature and M_E is the molal concentration of all constituents in a solution. After the solution analysis, module scale performances were evaluated to determine the optimum conditions. For each module, the performance of their smallest unit was first evaluated by using the numerical simulation procedure. Net water permeation, diluted DS production rate, and DS outlet concentration were calculated for a single membrane sheet of a SW membrane module and PF type module. Performance of the total module is then estimated by multiplying the performance parameters by the number of membrane sheets. Similarly, the module scale performance of an HF module was estimated by investigating a single fiber's performance. However, the water and solute permeation through the membrane was calculated by the mass transport models presented in our previous studies [Chapter 4, 5 and 6].

Since the flowrate, concentration, and velocity of both feed and draw solution changes spatially for any membrane module, the mass transport models were applied over numerous small discretized sections of the membrane to assess the local water flux and salt flux. Considering the local flux values, velocity, and concentration of the surrounding locations were determined by coupling the fluid conservation theories with the solution diffusion modules. Then the module scale performances were calculated by combining the local performances. Here, two major module scale performances (such as DS production rate and the final DS concentration) of various membrane modules are given for the system design. Detailed mathematical modeling can be found in chapter 4, 5 and 6. Diluted DS outlet of an SW module is given by,

$$Q_D^{Out} = N_{sheet} \sum_{i=1}^{(m-p)/2} Q_D(i,1) \tag{7.3}$$

here, N_{sheet} is the total number of membrane sheet, Q_D is the DS flowrate, i is the index for the discrete locations as shown in Figure 4.1. Final DS concentration of the SW module is given by,

$$C_D^{Out} = \frac{N_{sheet} \sum_{i=1}^{(m-p)/2} Q_D^{Slt}(i,1)}{Q_D^{Out}} \tag{7.4}$$

here Q_D^{Slt} is the solute flowrate in the draw channel. Similarly, the diluted DS outlet flowrate of the PF type membrane module is expressed by,

$$Q_D^{out} = N_{sheet} \sum_{j=1}^{n} Q_D(m,j) \tag{7.5}$$

Final DS concentration of a PF type membrane module is given below,

$$C_D^{out} = \frac{N_{sheet} \sum_{j=1}^{n} Q_D^{slt}(m,j)}{Q_D^{out}} \tag{7.6}$$

Considering the longitudinal flow direction in a HF module, the DS outlet flowrate is given by,

$$Q_{DS,out} = N_{fibre} \sum_{i=1}^{m} Q_{DS}(i) \tag{7.7}$$

Finally, the DS outlet concentration of the HF module is expressed by,

$$C_{DS,out} = C_{DS}(m) \tag{7.8}$$

Employing the module scale performance, optimum number of membrane modules and DS inlet flowrate per pressure vessel was calculated by using an overall performance index combining the recovery rate and final DS concentration of the system. This performance index is termed as OPR in this study, which is given by,

$$OPR(D,E) = \left[\frac{RF(D,E) \times CF(D,E)}{N_E}\right] \times 100 \tag{7.9}$$

here, $RF(D,E)$ is the recovery fraction at any draw solution inlet flowrate and the number of elements per vessel. Similarly, $CF(D,E)$ is the concentration fraction at any DS inlet flowrate and the number of elements connected serially in a pressure vessel. Recovery fraction is defined as,

$$RF(D,E) = \frac{R(D,E)}{R_D} \tag{7.10}$$

here, $R(D,E)$ is the recovery rate of the system at any draw solution inlet flowrate and after any number of elements. R_D stands here for the desired recovery rate. On the other hand, concentration fraction is given by,

$$CF(D,E) = \frac{C_D^F}{C_D(D,E)} \tag{7.11}$$

here, C_D^F is the desired final DS concentration of the system and $C_D(D,E)$ is the actual DS outlet concentration of the system at any DS inlet flowrate and after any number of elements in a pressure vessel. OPR value was calculated for a range of DS inlet flowrate and the number of elements. The optimum number of elements and DS inlet flowrates are determined based on the maximum value of OPR.

Once the optimum number of elements per pressure vessel was determined, the total diluted DS production rate of each pressure vessel is given by,

$$Q_{PV} = \sum_{s=1}^{E} Q_D^{out}\,(s) \tag{7.12}$$

here, Q_D^{out} is the rate of diluted DS produced from any elements assembled in the pressure vessel. s is an index for the sequence of the serially connected elements and E represents the number of elements connected in a pressure vessel. If the total production capacity of the FO plant is not achieved by a single pressure vessel, several pressure vessels are connected parallel to each other. Thus, the number of pressure vessels required to obtain the production capacity of the plant is expressed by,

$$N_{PV} = \frac{\lambda}{Q_{PV}} \tag{7.13}$$

here, λ is the production capacity of the plant. Total number of elements for the whole system is given by,

$$N_E = N_{PV} \times E \tag{7.14}$$

here, E is the number of elements per pressure vessel and N_{PV} is the total number of pressure vessels. Total membrane area of the system is calculated by using the membrane area of each module.

7.3.2 Algorithm flowchart for the software

One of the most important parts of software development is algorithm design. An algorithm flowchart is a step by step description of operations that should be performed to obtain the outcome of the software. Algorithms are then coded using a programming language to develop the software. Figure 7.2 shows the algorithm of the software developed for the FO system analysis. According to the algorithm, the first operation of the system analysis performed by the FOSA software is receiving the general information about the project to save the information for further documentation and reporting purposes. In the following step, this software requires information about the FS and DS. Properties of the user defined feed and draw solution are analysed in this step. Then the osmotic pressure of the feed and draw solution were compared.

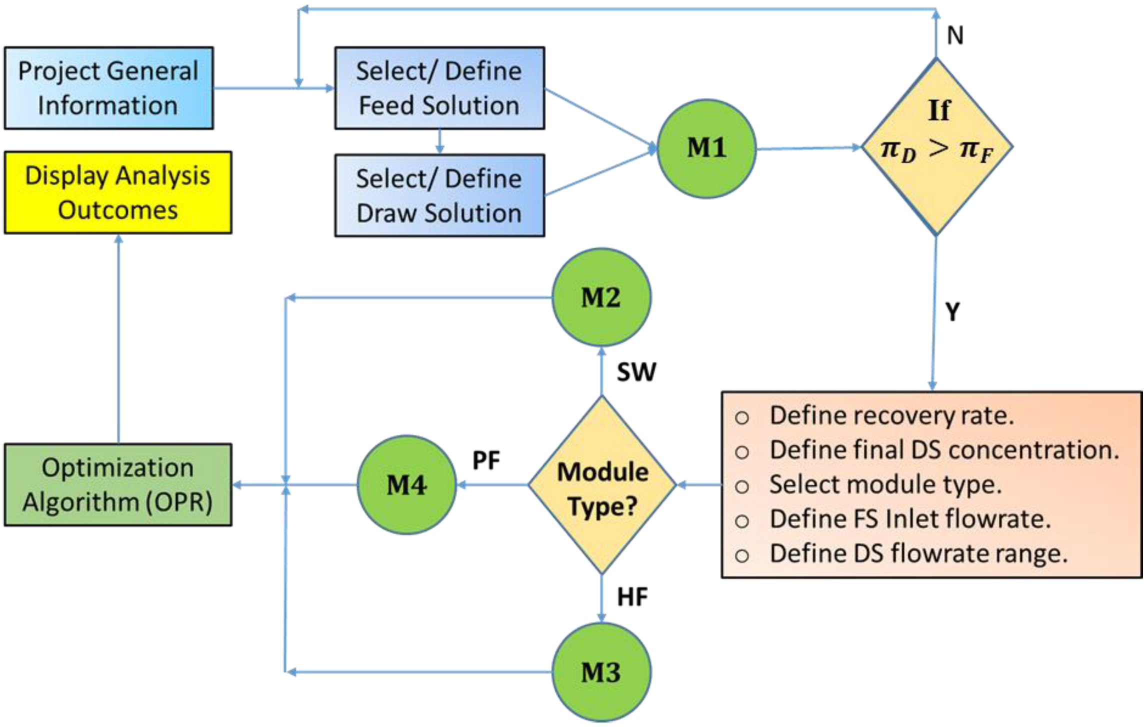
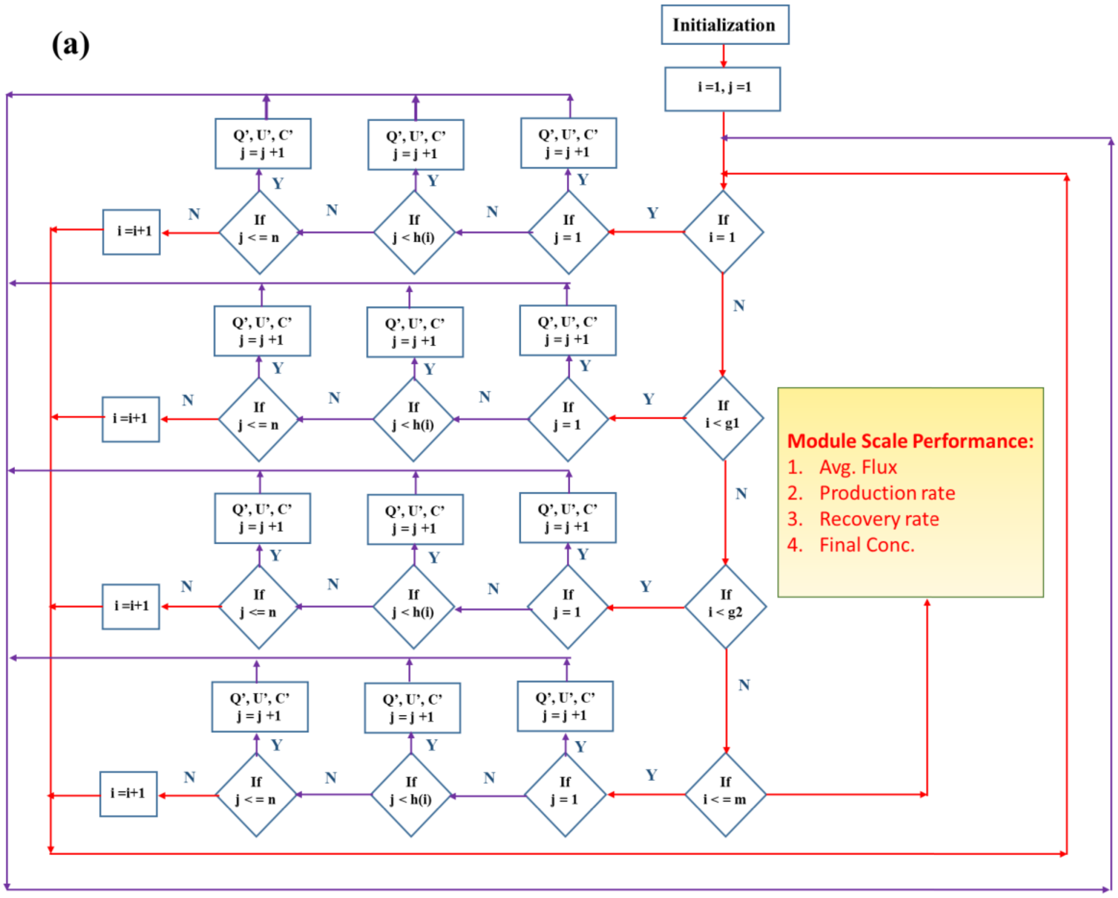


Figure 7.2. The algorithm used for the development of FOSA. M1 is the model used for the solution analysis whereas M2, M3, and M4 are the models for spiral wound, hollow fiber, and plate and frame module.

If the osmotic pressure of the DS is found smaller than the feed solution, user requires to redefine the solutions. Otherwise, the software moves to the next level of system analysis, which is optimization. In this operation, the design target and inlet operating conditions are required to be defined. Desired recovery rate and final draw solution concentration are entered. Membrane module type and FS inlet flowrate are also selected. Then a range of DS inlet flowrate is set to find the optimum conditions. However, after defining the desired design objectives and inlet operating conditions, these parameters are used as the input parameters for the optimization algorithm. The optimization algorithm is discussed in detail in Chapter 4. In this optimization step, the algorithm first finds out the module type (SW, HF or PF) and based on the module type it employs the appropriate mathematical models (M2, M3, and M4) developed in our previous studies (described in chapter 4, 5 and 6). Three more algorithms were also designed to solve these models to investigate the module scale performance analysis of spiral wound, hollow fiber and plate and frame module using the FOSA software, which is shown in Figure 7.3.



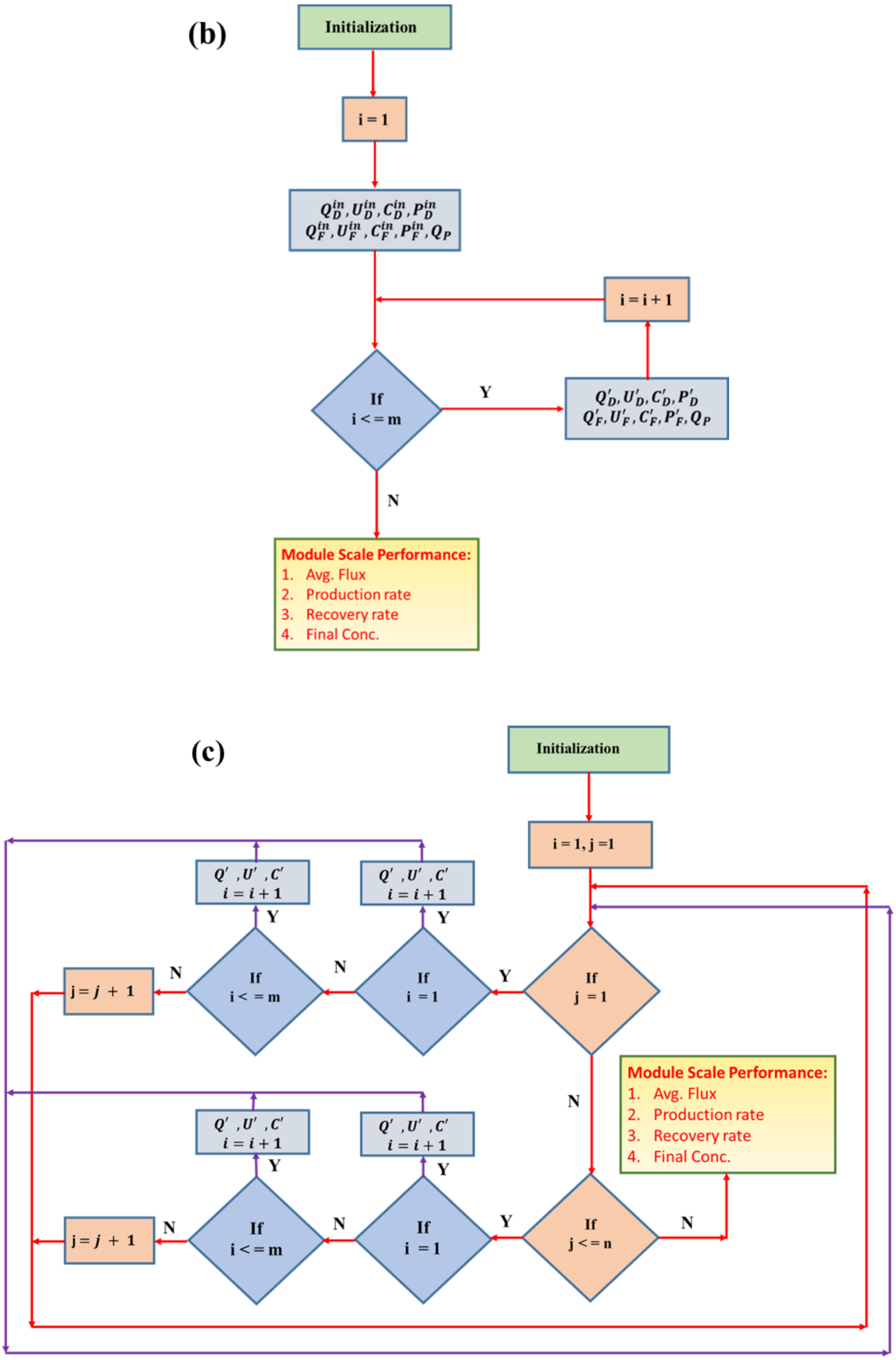


Figure 7.3. Algorithm flowcharts designed to iteratively simulate the (a) spiral wound (b) hollow fiber and (c) plate and frame module performance.

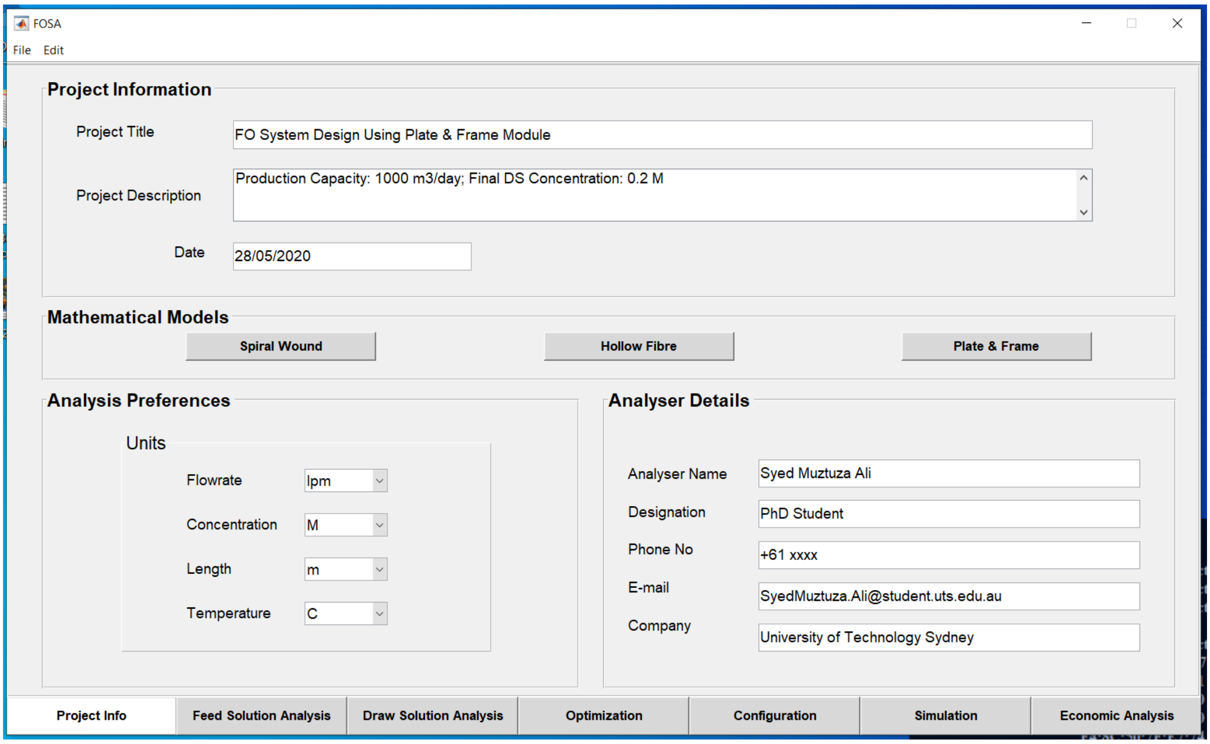
Figure 7.3(a) describes the sequential numerical operations to simulate the performance of a spiral wound module. The simulation begins with the initialization, which receives the inlet operating conditions from the GUI of the FOSA. Then the flowrate, concentration, velocity, water flux, and solute fluxes are calculated at each discrete location on the membrane surface iteratively. The iteration process starts with i = 1 and j = 1 those indicate the first row and columns of the discrete cells as shown in **Error! Reference source not found.**(a). At any value of j, when i = 1 feed solution flowrate, velocity and concentration are the inlet feed solution parameters. Similarly, for 1 =< i <= g1 and j = 1, the DS flowrate, velocity, and concentration are received from the user provided inlet conditions for the draw solution. In addition, DS flowrate and water flux were considered zero on the glue line over the membrane (for g1 =< i <= g2 and j = 1 to h(i)). Here g1 and g2 are the indexes of the first and last discrete cells over glue line in the DS was assumed to change its flow orientation. The iterative simulations first calculate the flow behavior and flux values for the first row (i =1 and any values of j) of the discrete elements, then proceed to the next rows sequentially as shown in Figure 7.3(a). At different sections of the membrane (1 =< i <= g1, g1 =< i <= g2 and g2 =< i <= m) due to the difference in flow direction of the DS the models are different. Therefore appropriate models are used (according to the location) to update the flowrate (Q'), velocity (U') and concentration (C') at each location. Finally, the net water permeation is calculated by adding the water permeation of all discrete locations which further gives the average flux of the module. Diluted DS production rate or DS outlet flowrate is estimated by adding flowrates of all cells at g2 =< i <= m and j=1. The final DS concentration of the module is also calculated considering concentrations of the same cells. Performance of a HF membrane module is simulated considering the algorithm presented in Figure 7.3(b). Since the flow orientation of the module employed in this study was longitudinal, iteration process only varied from i = 1 to m. In contrast, the PF module has an orthogonal flow orientation, therefore the iterative simulation was conducted from i = 1 to m and j = 1 to n as shown in Figure 7.3(c). However, for the optimization process, module scale performances are investigated at different operating conditions, and OPR value is evaluated for all operating conditions. Optimum operating conditions and number of elements per pressure vessel are determined from the maximum OPR value. In the final step, the full scale FO system design and operating parameters are displayed graphically as shown in Figure 7.2.

7.4 Results and discussion

A FO system analysis software was developed in this study to design and optimize a full scale FO system using various types of membrane modules. MATLAB programming language was used to code the frontend (graphical user interface) and backend (solution analysis and optimization) of the software. This software employed the mathematical models and optimization algorithm developed in our previous studies (chapter 4, 5 and 6) to design and optimize an FO system using a commercial TFC 8040 SW, a CTA HF, and a PF module. Finally, it compares the design of the FO system using these commercial membrane modules.

7.4.1 Graphical user interface of the software

A user friendly graphical user interface (GUI) was developed as a visual component of the software (FOSA) to facilitate the interaction between the user and the software. It receives the input for the simulation and optimization (backend operations) and displays the output of the calculations.



Project Information

Project Title: FO System Design Using Plate & Frame Module

Project Description: Production Capacity: 1000 m3/day; Final DS Concentration: 0.2 M

Date: 28/05/2020

Mathematical Models

Analysis Preferences

Units

Flowrate: lpm

Concentration: M

Length: m

Temperature: C

Analyser Details

Analyser Name: Syed Muztuza Ali

Designation: PhD Student

Phone No: +61 xxxx

E-mail: SyedMuztuza.Ali@student.uts.edu.au

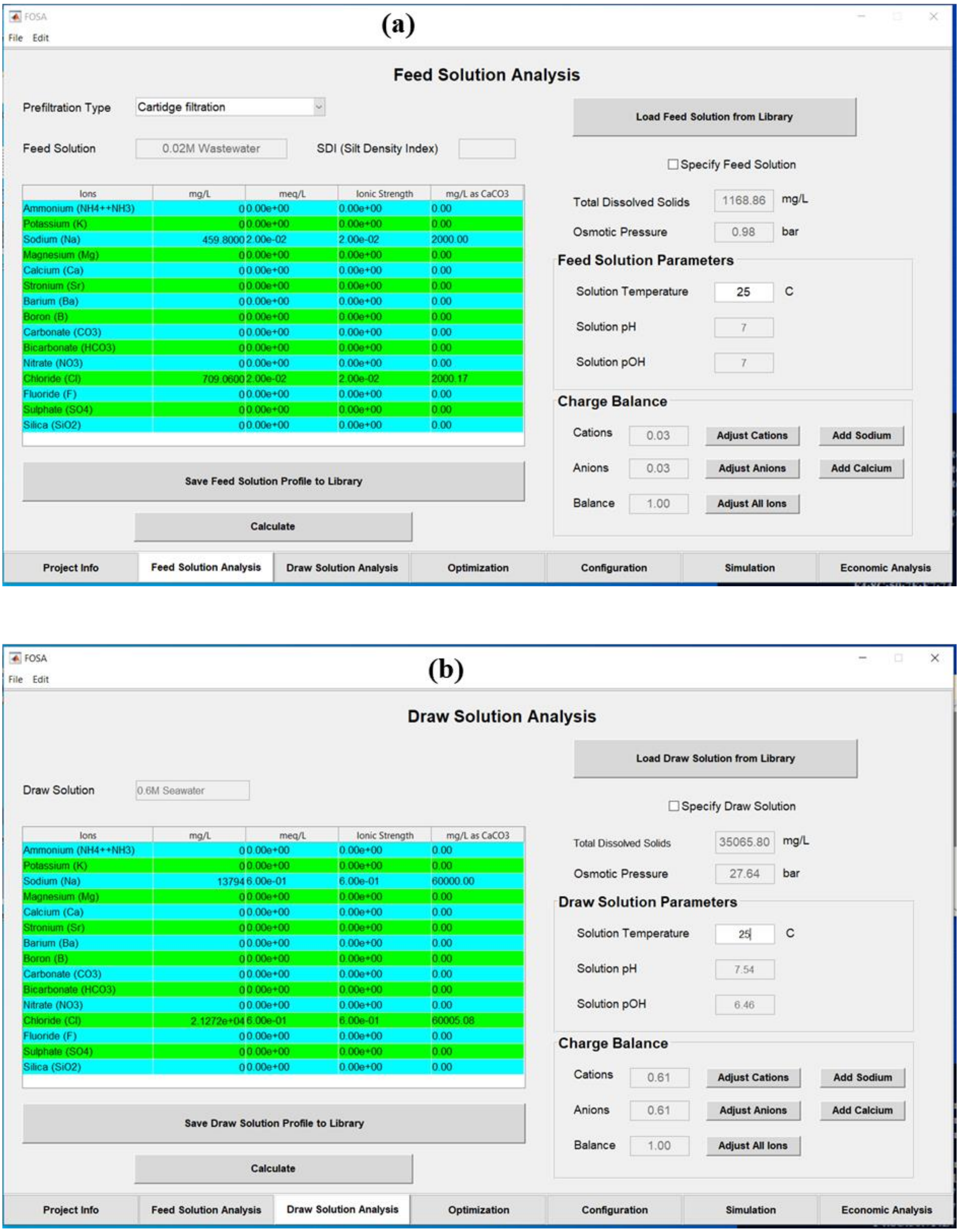
Company: University of Technology Sydney

Figure 7.4. Graphical user interface of the forward osmosis system analysis (FOSA) software

According to the steps of FO system analysis described in section 3 of this study, the analytical operations related to the design and optimization of FO system are dependent on each other and needed to perform in a sequential manner. Therefore, this study designed a GUI of the software that comprised seven easily navigable tabs to carry out these operations consecutively as shown in Figure 7.2. First tab of the GUI only receives information about the project which is shown in Figure 7.4. It keeps the record of project title, description of the project, details of the designer and the units used for the design. This information is required for further documentation. After receiving the information about the FO project, the next operation is the analysis of solution properties. Therefore, this software developed two other tabs for FS and DS analysis. Properties of the feed and draw solution are used for the optimization of operating conditions. So, this software contains another optimization tab. It has two more tabs for the configuration of full scale design objectives and to display the simulated results. Finally, this software has another tab for economic analysis. Since the economic analysis is out of the scope of this thesis. This tab is designed to keep provisions for future studies. The subsequent sections will describe FO system design and optimization using the FOSA software.

7.4.2 Solution properties analysis

The first step of FO system design is the evaluation of feed and draw solution properties. In order to analyze the solution characteristics "Feed Solution Analysis" and "Draw Solution Analysis" tabs were created as shown in Figure 7.5(a) and (b). It can be seen from the figure that the FS analysis tab consists of few fields to receive input from the user, to display the analysis results and few command buttons. In order to receive the composition of the FS, the software provides two options. User can click on the command button "Load Feed Solution from Library", which opens a library of different FS. Selection of any predefined FS loads the composition of the FS for analysis and display the composition as well. In case of a new FS (which is not listed in the library), user can check the button "Specify Feed Solution" which enables the user to enter the concentration of some predefined compounds. The user defined solution can be saved as a new FS in the library, later the solution can be loaded from the library for further analysis. After loading the composition, "Calculate" button on this tab is clicked to evaluate the solution properties in terms of TDS and osmotic pressure using the models presented in section 7.3.1 (equation 7.1 and 7.2). TDS and Osmotic pressure are displayed after the calculation. Similarly, in the "Draw Solution Analysis" tab, DS composition is either loaded from the library or defined by the user. TDS and osmotic pressure of the draw solution are calculated and displayed for further operation.



Feed Solution Analysis

Prefiltration Type: Cartidge filtration

Feed Solution: 0.02M Wastewater

SDI (Silt Density Index):

☐ Specify Feed Solution

Ions	mg/L	meq/L	Ionic Strength	mg/L as CaCO3
Ammonium (NH4++NH3)	0	0.00e+00	0.00e+00	0.00
Potassium (K)	0	0.00e+00	0.00e+00	0.00
Sodium (Na)	459.8000	2.00e-02	2.00e-02	2000.00
Magnesium (Mg)	0	0.00e+00	0.00e+00	0.00
Calcium (Ca)	0	0.00e+00	0.00e+00	0.00
Stronium (Sr)	0	0.00e+00	0.00e+00	0.00
Barium (Ba)	0	0.00e+00	0.00e+00	0.00
Boron (B)	0	0.00e+00	0.00e+00	0.00
Carbonate (CO3)	0	0.00e+00	0.00e+00	0.00
Bicarbonate (HCO3)	0	0.00e+00	0.00e+00	0.00
Nitrate (NO3)	0	0.00e+00	0.00e+00	0.00
Chloride (Cl)	709.0600	2.00e-02	2.00e-02	2000.17
Fluoride (F)	0	0.00e+00	0.00e+00	0.00
Sulphate (SO4)	0	0.00e+00	0.00e+00	0.00
Silica (SiO2)	0	0.00e+00	0.00e+00	0.00

Total Dissolved Solids: 1168.86 mg/L

Osmotic Pressure: 0.98 bar

Feed Solution Parameters

Solution Temperature: 25 C

Solution pH: 7

Solution pOH: 7

Charge Balance

Cations: 0.03

Anions: 0.03

Balance: 1.00

Draw Solution Analysis

Draw Solution: 0.6M Seawater

☐ Specify Draw Solution

Ions	mg/L	meq/L	Ionic Strength	mg/L as CaCO3
Ammonium (NH4++NH3)	0	0.00e+00	0.00e+00	0.00
Potassium (K)	0	0.00e+00	0.00e+00	0.00
Sodium (Na)	13794	6.00e-01	6.00e-01	60000.00
Magnesium (Mg)	0	0.00e+00	0.00e+00	0.00
Calcium (Ca)	0	0.00e+00	0.00e+00	0.00
Stronium (Sr)	0	0.00e+00	0.00e+00	0.00
Barium (Ba)	0	0.00e+00	0.00e+00	0.00
Boron (B)	0	0.00e+00	0.00e+00	0.00
Carbonate (CO3)	0	0.00e+00	0.00e+00	0.00
Bicarbonate (HCO3)	0	0.00e+00	0.00e+00	0.00
Nitrate (NO3)	0	0.00e+00	0.00e+00	0.00
Chloride (Cl)	2.1272e+04	6.00e-01	6.00e-01	60005.08
Fluoride (F)	0	0.00e+00	0.00e+00	0.00
Sulphate (SO4)	0	0.00e+00	0.00e+00	0.00
Silica (SiO2)	0	0.00e+00	0.00e+00	0.00

Total Dissolved Solids: 35065.80 mg/L

Osmotic Pressure: 27.64 bar

Draw Solution Parameters

Solution Temperature: 25 C

Solution pH: 7.54

Solution pOH: 6.46

Charge Balance

Cations: 0.61

Anions: 0.61

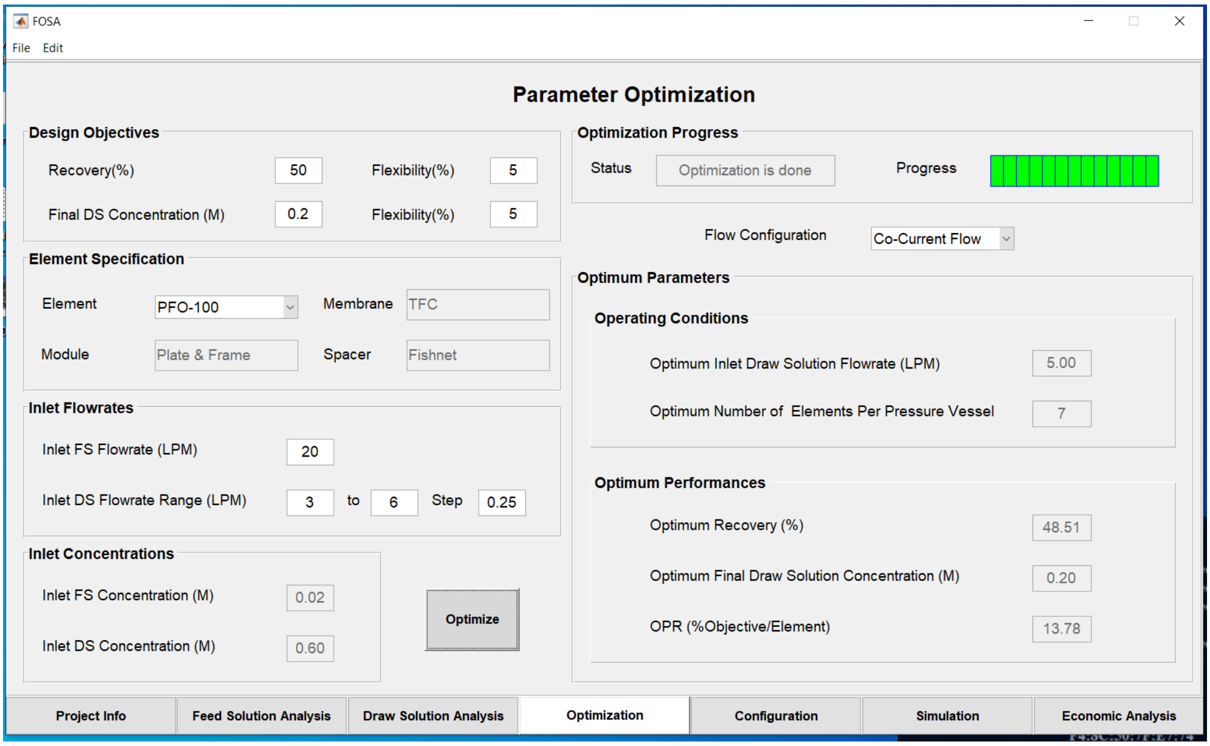
Balance: 1.00

Figure 7.5. Analysis of (a) feed and (b) draw solution properties using FOSA.

Apart from the TDS and osmotic pressure, charge balance of the solution is investigated to validate the solution analysis. This tab also allows the user to select the prefiltration type. Although the effect of prefiltration of FS on system performance is not considered in this study, this option is kept for further improvement of the study. Since this study designed FO system to osmotically dilute seawater, it assumed 0.6 M NaCl Solution as draw solution as its osmotic pressure is almost similar to the seawater. Similarly, 0.02 M NaCl solution was assumed as the feed solution, as this solution's osmotic pressure is almost equal to the same of MBR effluent. It can be seen from Figure 7.5(a) that TDS and osmotic pressure of the FS were 1168 mg/L and 0.98 bar, whereas the Figure 7.5(b) shows that the TDS and osmotic pressure of the DS were 35000 mg/L and 28 bar respectively.

7.4.3 Optimization of operating conditions

This section describes the FO system optimization using the FOSA software developed in this study. For this operation, the software consists of a tab called "Optimization" which is shown in Figure 7.6. Optimum number of elements and DS inlet flowrate per pressure vessel is determined in this process for the desired performance of the system. In this tab, the desired performance parameters are received as an input to the software for the optimization. Desired performance of the FO system is referred to as the design objectives in this study. Recovery rate and final DS concentration are two of the most important performance parameters of the FO system. However, it has been found that searching the optimum condition within a wide range of operating condition leads to the design of a system which may operate at very high recovery rate but at very high final draw solution concentration or it may produce extremely diluted draw solution but at very low recovery rate (Ali et al. 2018; Phuntsho et al. 2017). Therefore, this software requires an additional parameter, flexibility that constrains the search domain within acceptable proximity to the desired recovery rate and final DS concentration. In this study, 5% flexibility for both recovery rate and final DS concentration was used. In addition, the type of membrane module is also selected from a drop-down list for the optimization. This software currently provides the option to select three different types of commercial membrane modules, such as SW 8040 (a commercial spiral wound TFC FO membrane module), HPC3250 (a commercial CTA hollow fiber membrane module) and PF100 (a commercial plate and frame type membrane element). After the selection of membrane modules, the FS and DS inlet flowrates are defined. For the optimization of draw solution inlet flowrate, a range of DS inlet flowrate is defined. Moreover, the inlet concentration/osmotic pressure of the solution is received from the previous feed and draw solution analysis tabs. Finally, the command button “Optimize”, sends these input parameters and design objectives to the optimization algorithm which runs the MATLAB code at the backend of the software to estimate the optimum operating conditions. The optimization algorithm also calls the function coded to determine the performance of a membrane module in terms of diluted DS production rate and final DS concentration at different DS inlet flowrates. Since this optimization process calculates module scale performance at different DS inlet flowrates and the number of elements, it is a computationally expensive process and takes relatively longer time to execute. Therefore, a colour bar is designed which extends its length to indicate the progress of the optimization process. It turns completely green when the optimization is done. Optimum parameters and performances are displayed in this tab as the output of the optimization process. It can be seen from Figure 7.6 that for the commercial plate and frame module used in this study (PFO-100), 7 elements are required to be connected serially in a pressure vessel and 5 L/min DS inlet flowrate per pressure vessel is optimum when the desired recovery rate, FS inlet flowrate and final DS concentration are 50%, 20 L/min and 0.15 M respectively. However, typically the housing of serially connected plate and frame element are called module (discussed in chapter 6). But since the housing for other types of membrane element (SW and HF) are referred to as the pressure vessel, this software used the term pressure vessel for PF type element also. In addition, the optimization algorithm found that operating the system at exactly 50% recovery rate is not optimal as the final DS concentration drops below the desired concentration of 0.2 M. Therefore, considering 5% flexibility, this software determined the recovery rate of the system at optimum operating conditions was 48.5% as shown in Figure 7.6.



Parameter Optimization

Design Objectives
Recovery(%) 50
Flexibility(%) 5
Final DS Concentration (M) 0.2
Flexibility(%) 5

Element Specification
Element PFO-100
Membrane TFC
Module Plate & Frame
Spacer Fishnet

Inlet Flowrates
Inlet FS Flowrate (LPM) 20
Inlet DS Flowrate Range (LPM) 3 to 6 Step 0.25

Inlet Concentrations
Inlet FS Concentration (M) 0.02
Inlet DS Concentration (M) 0.60

Optimization Progress
Status Optimization is done
Progress

Flow Configuration Co-Current Flow

Optimum Parameters

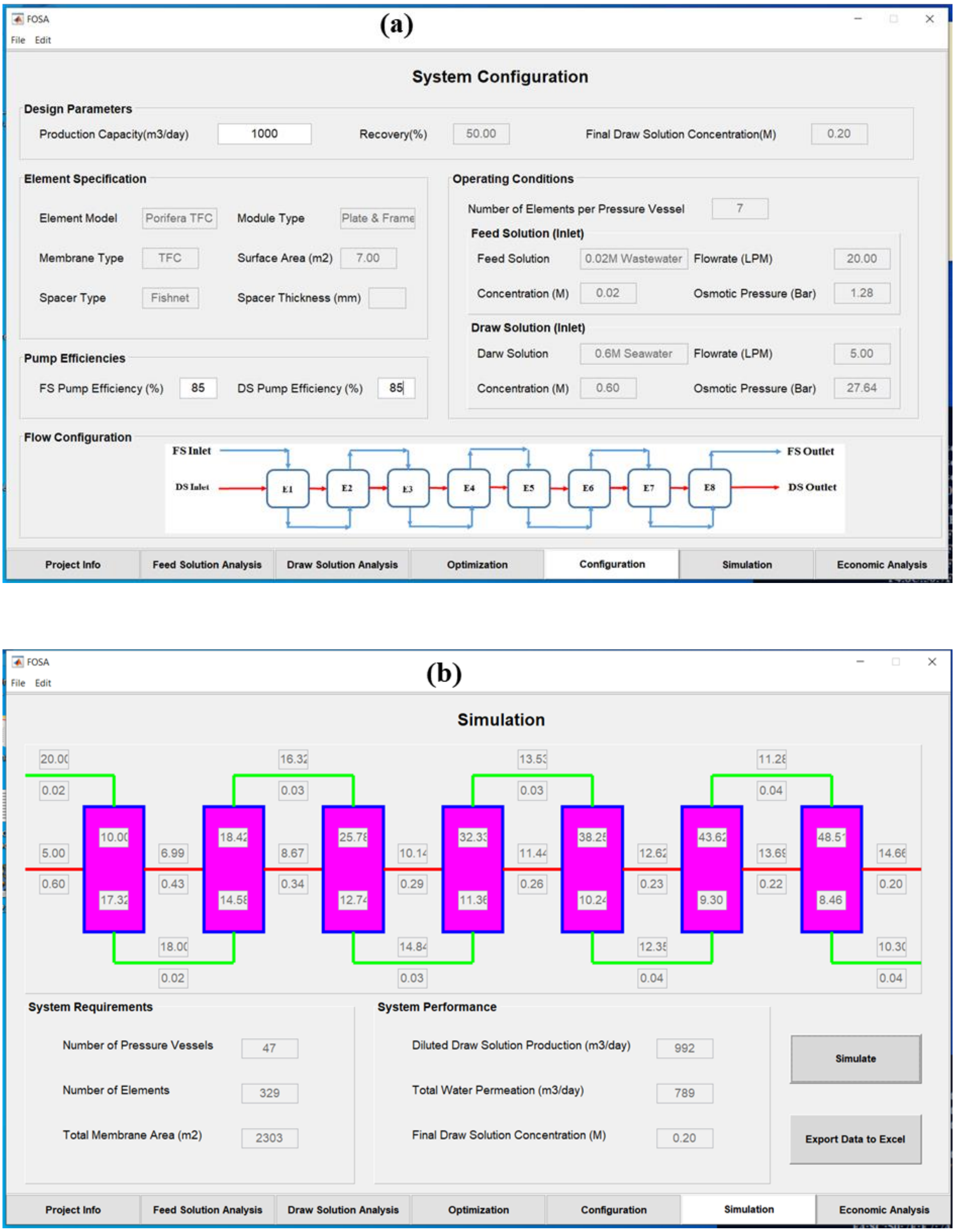
Operating Conditions
Optimum Inlet Draw Solution Flowrate (LPM) 5.00
Optimum Number of Elements Per Pressure Vessel 7

Optimum Performances
Optimum Recovery (%) 48.51
Optimum Final Draw Solution Concentration (M) 0.20
OPR (%Objective/Element) 13.78

Figure 7.6. Optimization of design and operating conditions using FOSA

7.4.4 Full scale system design information

The main objective of the software is to find the information required to design and optimize a full scale FO system. The previous section discussed the optimization process using FOSA software which determined the optimum number of elements and DS inlet flowrate per pressure vessel for the desired recovery rate and final draw solution concentration. This section explains the determination of design information for full scale design using the FOSA software. For this step, FOSA software consists of two more tabs. The first tab is named “Configuration”. In this tab, the daily production capacity of the FO system receives as an input for the system design which is shown in Figure 7.7(a). Moreover, the overall system configuration is displayed here in this tab including the specification of the membrane element used for the design, operating conditions, and flow orientations. Feed and draw solution pump efficiencies are also received as user input. Although this information is not used for the current analysis, these fields are designed to conduct economic analysis using FOSA for our future studies. Figure 7.7(b) shows the simulation tab of the FOSA software which graphically displays all the major simulated results required for the system design. It displays the changes in average flux, recovery rate, FS flowrates and concentrations, DS flowrates, and concentrations of the serially connected elements in a pressure vessel. Average flux reduced from 17.3 L/m^{2}h to 8.5 L/m^{2}h when 7 commercial PFO 100 modules were connected serially. Due to the permeation of water from the feed channel to draw channel, DS flowrate increased from 5 L/min to 14.7 L/min for the same number of elements. In addition, the DS is diluted from 0.6 M to 0.2 M. On the other hand, FS flowrate reduced from 20 L/min to 10.3 L/min and the FS was concentrated from 0.02 M to 0.04 M. Considering the performance of the elements connected in a pressure vessel, this software calculated the total number of pressure vessels, membrane elements, and the total membrane area, those are essential to design a full scale FO system for the desired production capacity, recovery rate, and final DS concentration.



System Configuration

Design Parameters

Production Capacity(m3/day) 1000
Recovery(%) 50.00
Final Draw Solution Concentration(M) 0.20

Element Specification

Element Model Porifera TFC
Module Type Plate & Frame
Membrane Type TFC
Surface Area (m2) 7.00
Spacer Type Fishnet
Spacer Thickness (mm)

Pump Efficiencies

FS Pump Efficiency (%) 85
DS Pump Efficiency (%) 85

Operating Conditions

Number of Elements per Pressure Vessel 7

Feed Solution (Inlet)

Feed Solution 0.02M Wastewater
Flowrate (LPM) 20.00
Concentration (M) 0.02
Osmotic Pressure (Bar) 1.28

Draw Solution (Inlet)

Darw Solution 0.6M Seawater
Flowrate (LPM) 5.00
Concentration (M) 0.60
Osmotic Pressure (Bar) 27.64

Flow Configuration

Simulation

System Requirements

Number of Pressure Vessels 47
Number of Elements 329
Total Membrane Area (m2) 2303

System Performance

Diluted Draw Solution Production (m3/day) 992
Total Water Permeation (m3/day) 789
Final Draw Solution Concentration (M) 0.20

Figure 7.7. (a) Configuration of full scale system and (b) visual display of information required to design a full scale FO system

According to Figure 7.7(b), total 47 pressure vessels are required to be connected parallel to each other to design a 1,000 m^3/day FO system with 50% recovery rate and 0.2 M final DS concentration using the commercial PF membrane element. Considering 7 elements in a pressure vessel, total 329 pressure vessels are required for the FO system. Therefore, total 2,303 m^2 membrane is used for the system. From section 7.5.1 to 7.5.4, application of the developed FOSA software in the design and optimization of a full scale FO system using PFO-100 membrane element was discussed.

7.4.5 Comparison of FO system design using different modules

Previous sections (7.5.1 to 7.5.4) discussed the developed software and its application to design and optimize an FO system for the osmotic dilution of seawater. In this section, the FOSA software compares the design of a full scale system using a commercial HF (HPC3205) and PF membrane module (PF-100) with an SW membrane module (SW 8040) at the same operating conditions and design targets (production capacity 1,000 m^3/day and final DS concentration 0.2 M). It was found that the lowest number of membrane module is required to design the FO system when the spiral wound TFC FO module was used, on the other hand, the highest number of elements are required for the CTA hollow fiber module. Total number of elements required to design the FO system using SW, PF, and HF membrane modules were 78, 329, and 731 respectively as shown in Table 7.1. Considering the membrane area of each module, the total membrane areas required for the FO system were 1,193 m^2, 13,503 m^2 and 2,303 m^2 using the SW, HF and PF module.

Table 7.1 Comparison of full scale FO system design using various types of membrane modules (here the full forms of DS and FS are draw solution and feed solution). Design and operating conditions for 1000 m^3/day FO system to produce 0.2 M diluted seawater. Membrane area of each spiral wound (SW), hollow fiber, and plate and frame (PF) type module is 15.3 m^2, 31.5 m^{2,} and 7 m^2.

Module Type	Total DS Inlet Flowrate (m^3/h)	Total FS Inlet Flowrate (m^3/h)	Number of elements	Total membrane area (m^2)	Volume of module (m^3)
SW 8040	14	55	78	1,193	2.57
HPC 3205	14	64	429	13,503	2.97
PF 100	14	56	329	2,303	8.15

As discussed in chapter 5, the HF module employed in this thesis operates at a very low draw and feed solution flowrate per element due to its very high packing density. Consequently, the average flux of the HF module was lower compared to the other modules. Therefore, the number of elements and membrane area required for the HF module is higher than the others. However, considering the dimension of each module (provided in the Table A1.1, A2.1 and A3.1 of the appendix section), space required for the membrane modules was also estimated which influences the footprint of the system. The space requirements for the SW, HF, and PF type modules were found to be 2.6 m^3, 2.97 $m^{3,}$ and 8.15 m^3.From the above discussion, it can be seen that the SW 8040 spiral wound module showed the best performance compared to the other modules used in this study for the osmotic dilution of seawater when the fouling effect is negligible. In comparison, HPC3205 hollow fiber module although requires the maximum membrane area, space required for the modules is lower compared to the plate and frame module due to its very high packing density. Therefore, hollow fiber module with enhanced water permeability (TFC) and optimized packing density can be very promising for full scale applications. Plate and frame module, however, requires smaller membrane area compared to the hollow fiber module, but due to the lower packing density, space required for the modules is almost double compared to the hollow fiber module. Hence, further improvement in packing density is needed to reduce the system footprint using the plate and frame module.

7.5 Conclusions

This study developed a system analysis software to design and optimize a full scale FO plant using various types of membrane modules. Detailed algorithms were designed to construct the general framework of the software. These algorithms were then coded using MATLAB programming language to design the frontend (graphical user interface) and backend (simulation and optimization of the FO system) layer of the software. Graphical user interface of the software consists of 6 easily navigable tabs to effectively execute the sequential steps of FO system analysis. The software was used to design and optimize a 1,000 m^3/day FO plant to produce 0.2 M diluted draw solution. For the same operating conditions, the designs of FO system using the three different types of commercial modules were compared. TFC spiral wound module showed the best performance among the three modules in terms of membrane area and the spacer requirement. In comparison, the packing density of the plate and frame module should be further optimized for large scale applications. Moreover, the hollow fiber module with enhanced water permeability and optimal packing density can be a very promising option for full scale system design. However, the software is found to be an efficient, accurate, and less time-consuming design tool compared to the manual design process. Moreover, the software can be easily updated by including new draw solutions, feed solutions, and membrane modules according to the user’s requirements. Therefore, the developed software has the potential to provide the researchers or industries with an opportunity to find novel and more efficient FO applications.

CHAPTER 8

CONCLUSIONS AND RECOMMENDATIONS

8 Conclusions and recommendations

8.1 Conclusions

This thesis is aimed at designing and optimizing a FO system for full scale applications. It developed theoretical frameworks to numerically simulate the performance of an FO process using various types of membrane modules (spiral wound, hollow fiber and plate frame) at different operating conditions. Theoretical results were compared with the pilot scale experimental data to validate the models developed in this study. Then, this study also developed a novel optimization algorithm to design and optimize an FO system for any desired performance. Finally to make this design and optimization process more efficient a system analysis software FOSA was developed. It receives design objectives as input and graphically displays the optimum design parameters and operating conditions employing the mathematical models for each type of module.

8.1.1 FO system analysis for optimum design and operating conditions using a spiral wound module

Spiral wound module is so far the most widely used FO membrane module for full scale applications. Therefore before studying the other type of modules, this study focused on investigating the performance of a spiral wound module. It modeled and simulated the performance of the FO system for various operating conditions and design parameters. A novel overall performance parameter was presented to estimate the performance of the system considering both the recovery rate and the final DS concentration. This overall performance parameter was used to compare the performance of the system at various operating conditions. Operating conditions corresponding to the maximum value of this parameter were considered the optimum conditions. It was found from this study that a higher draw solution inlet flowrate was required to obtain the higher recovery rate for the same final draw solution concentration. In contrast, a lower draw solution flowrate was required when the targeted final draw solution concentration was lower for the same recovery rate. In addition, increasing the number of elements in a pressure vessel increased the recovery rate and reduced the final draw solution concentration. But after a certain number, increasing the element was not optimal. Finally, a FO system of 100,000 m^3 diluted draw solution production capacity per day at 0.15 M final concentration was designed by the FOSA software. It was found that for 50% recovery rate, 40 L/min feed solution inlet flowrate, and 4.7 L/min draw solution inlet flowrate, 4 elements are needed to be connected serially in a pressure vessel and total 2879 pressure vessels are required to design the desired 100,000 m^3/day FO system.

8.1.2 FO system design and simulation for real scale applications using a hollow fiber membrane module

Followed by the spiral wound module, this study developed mathematical models to design and optimize a full scale FO system employing a hollow fiber membrane module considering its very high packing density. Accuracy of the developed theoretical models was assessed by comparing it with the pilot scale experimental data using a CTA hollow fiber FO module. Less than 10% difference between the simulation and experimental data supports the accuracy of the developed models. Then the appropriate membrane modular arrangement and the ranges of operating conditions for the module were determined for the optimization. Parallel arrangement of single element module was found more suitable than the serial arrangement for this module. Finally, a 1,000 m^3/day FO plant was designed and optimized by using the hollow fiber module to produce 0.25 M, 0.20 M and 0.15 M diluted seawater. It was found that the optimum feed solution inlet flowrates per module for 0.25 M, 0.2 M, and 0.15 M final draw solution concentration were 2 L/min, 2.5 L/min, and 2.5 L/min respectively. Similarly, 0.8 L/min, 0.55 L/min, and 0.32 L/min draw solution inlet flowrates were the optimum flowrates for the same final draw solution concentration. Moreover, the number of elements to design the FO system for 0.25 M, 0.2 M, and 0.15 M final concentration were 370, 435, and 555 respectively.

8.1.3 Design and optimization of full-scale FO system design using a plate and frame module

With the growing interest in plate and frame module's application in FO processes, this study mathematically modeled a full scale FO system to theoretically estimate its performance using a commercially available plate and frame module under various design considerations. About 10% difference between the theoretically calculated and experimentally measured performance was observed which shows very good reliability of the developed mathematical models. Moreover, an overall performance indicator considering the recovery rate, final draw solution concentration, and the number of elements were used to determine the optimum draw solution inlet flowrate and membrane modular configurations. The optimization algorithm was then applied to design and optimize a 1000 m^3/day FO plant employing a commercial plate and frame type membrane element for a desired recovery rate and final draw solution concentration by using seawater (0.6M NaCl Solution) as draw solution and secondary MBR effluent (0.02 M NaCl Solution) as feed solution. 7 plate and frame elements and 5 L/min draw solution inlet flowrate per module are optimum conditions to design a 1000 m^3/day FO plant to produce 0.2 M diluted draw solution at 50% recovery rate and 20 L/min feed solution inlet flowrate. Finally, the models and the optimization algorithm were employed to investigate the effects of different design objectives and operating conditions on the FO plant design. It was found that for 20 L/min feed solution inlet flowrate and 0.2 M final concentration, to improve the recovery rate of the system from 40% to 60% the draw solution inlet flowrate has to be increased from 3.75 L/min to 6 L/min and the number of elements to be increased from 6 to 9.

8.1.4 Development of a forward osmosis system analysis (FOSA) software for full system design and operation

This study developed a system analysis software to design and optimize full scale FO plant using various types of membrane modules. Detailed algorithms were designed to construct the general framework of the software. These algorithms were then coded using MATLAB programming language to design the frontend (graphical user interface) and backend (simulation and optimization of the FO system) layer of the software. Graphical user interface of the software consists of 6 easily navigable tabs to effectively execute the sequential steps of FO system analysis. The software was used to design and optimize a 1000 m^3/day FO plant to produce 0.2 M diluted draw solution. For the same operating conditions, the designs of FO system using the three different types of commercial modules were compared. TFC spiral wound module showed the best performance among the three modules. In comparison, the water permeability and packing density of the commercial CTA hollow fiber module should be further optimized, whereas the packing density of the plate and frame module is required to be increased further for large scale applications. However, the software is found to be an efficient, accurate, and less time-consuming design tool. Moreover, the software can be easily updated by including new draw solutions, feed solutions, and membrane modules according to the user's requirements. Therefore, the developed software has the potential to provide the researchers or industries with an opportunity to find novel and more efficient FO applications.

The first objective of this thesis was to demonstrate the major components, operating conditions, design and performance parameters of a full scale FO system which is satisfied through review of literature presented in Chapter -2 and the description of the system reported in Chapter -3. Theoretical models presented in Chapter-4, 5 and 6 showed that the second objective of developing a theoretical framework to analytically estimate the performance of the FO system using a spiral wound, hollow fiber, and plate and frame module was fulfilled. The next aim of this research to develop an optimization algorithm was also met by optimization algorithm presented in Chapter 4 and 6. Finally the main aim of this thesis to develop a system analysis software for the design and optimization of a full scale FO plant is attained through the software “FOSA” developed in Chapter – 7. Therefore this thesis successfully satisfied all the objectives presented in the Introduction chapter.

8.2 Recommendations

This study developed a comprehensive theoretical framework to theoretically design and optimize large scale FO plant. All three commercially available FO membrane modules (spiral wound, hollow fiber and plate, and frame module) were mathematically modeled to simulate their performance at various operating conditions. Simulated results were validated by comparing it with pilot-scale experimental data. A novel optimization algorithm was developed to optimize the modular configuration and operating conditions to design the plant for desired production capacity, recovery rate, and final draw solution concentration. Finally, an FO system analysis software was developed to make the design process more efficient.

Nonetheless, there are some shortcomings and scope of improvement of this study. To address the issues of the study, the following recommendations are suggested:

1. In this full scale FO system analysis study, scaling or inorganic fouling potential of the solutions were assumed negligible for simplicity. However, for a more precise estimation of the system performance, the effects of water chemistry should be included in the model.
2. To design FO plants for more realistic applications empirical models should be developed to incorporate the effects of fouling on the system performance considering the quality of feed water and pretreatment type.
3. The FO membranes used in the system do not perform uniformly over a long time. According to the types of application and operating conditions the membrane performance deteriorates at a different rate. Therefore, a membrane performance factor as a function of membrane life and application should be considered for membrane modeling.
4. For any specific commercial membrane modules, the manufacturers recommended some guidelines for safe and efficient operations. For example maximum feed solution or draw solution flowrate, or maximum recovery rate, etc. However, as an initial study, this work did not constrain the limit of the operating conditions. For further improvement, the range of the operating conditions should be defined.
5. Technoeconomic analysis can be further incorporated with the software by calculating the energy consumption of the processes. However, energy consumption is a function of pressure drop across the membrane modules.

Theoretical estimation of the pressure drop across a large scale membrane arrangement is very challenging. Therefore, more empirical relationships are needed to formulate the pressure drop and operating conditions for various membrane modules.

6. It was found that due to the very high packing density of the hollow fibre membrane module, the shell and bore side flow channel is very narrow. As a result the module is operated at a low feed and draw solution flowrate which lead to a lower module average flux. However, reducing the packing density to increase the average water flux will reduce the total membrane surface area that lowers the net production of the module. Therefore an optimization of packing density is essential for the module design.
7. Currently, the FOSA software includes one draw solution (seawater), one feed solution (secondary effluent), and three different commercial membrane modules. In order to widen the applications of this software more draw solutions, feed solutions, and membrane modules are required to be added to the software.
8. Scope of this software is so far limited to the FO processes. It can be further extended to design other osmotically driven membrane processes such as PAO (pressure assisted osmosis), PRO (pressure retarded osmosis) processes, etc.
9. This software can be further converted to a web application to reach out to various industrial and academic users. Feedback from diverse groups of users can be very useful to design the FO system for full scale commercial applications.

9 References

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*, vol. 239, no. 1, pp. 10-21.

Ait-Kadi, M. 2016, 'Water for development and development for water: realizing the sustainable development goals (SDGs) vision', *Aquat. Procedia*, vol. 6, pp. 106-10.

Akther, N., Ali, S.M., Phuntsho, S. & Shon, H. 2020, 'Surface modification of thin-film composite forward osmosis membranes with polyvinyl alcohol–graphene oxide composite hydrogels for antifouling properties', *Desalination*, vol. 491, p. 114591.

Akther, N., Lim, S., Tran, V.H., Phuntsho, S., Yang, Y., Bae, T.-H., Ghaffour, N. & Shon, H.K. 2019, 'The effect of Schiff base network on the separation performance of thin film nanocomposite forward osmosis membranes', *Separation and Purification Technology*, vol. 217, pp. 284-93.

Akther, N., Phuntsho, S., Chen, Y., Ghaffour, N. & Shon, H.K. 2019, 'Recent advances in nanomaterial-modified polyamide thin-film composite membranes for forward osmosis processes', *Journal of Membrane Science*, vol. 584, pp. 20-45.

Akther, N., Sodiq, A., Giwa, A., Daer, S., Arafat, H.A. & Hasan, S.W. 2015, 'Recent advancements in forward osmosis desalination: A review', *Chemical Engineering Journal*, vol. 281, pp. 502-22.

Alcamo, J., DÖLl, P., Henrichs, T., Kaspar, F., Lehner, B., RÖSch, T. & Siebert, S. 2003, 'Development and testing of the WaterGAP 2 global model of water use and availability', *Hydrological Sciences Journal*, vol. 48, no. 3, pp. 317-37.

Ali, S.M. & Chakraborty, A. 2016, 'Adsorption assisted double stage cooling and desalination employing silica gel + water and AQSOA-Z02 + water systems', *Energy Conversion and Management*, vol. 117, pp. 193-205.

Ali, S.M., Haider, P., Sidhu, D.S. & Chakraborty, A. 2016, 'Thermally driven adsorption cooling and desalination employing multi-bed dual-evaporator system', *Applied Thermal Engineering*, vol. 106, pp. 1136-47.

Ali, S.M., Kim, J.E., Phuntsho, S., Jang, A., Choi, J.Y. & Shon, H.K. 2018, 'Forward osmosis system analysis for optimum design and operating conditions', *Water Research*, vol. 145, pp. 429-41.

Altaee, A., Braytee, A., Millar, G.J. & Naji, O. 2019, 'Energy efficiency of hollow fibre membrane module in the forward osmosis seawater desalination process', *Journal of Membrane Science*, vol. 587, p. 117165.

Altaee, A. & Hilal, N. 2015, 'High recovery rate NF–FO–RO hybrid system for inland brackish water treatment', *Desalination*, vol. 363, pp. 19-25.

Anderson, J.L. & Malone, D.M. 1974, 'Mechanism of Osmotic Flow in Porous Membranes', *Biophysical Journal*, vol. 14, no. 12, pp. 957-82.

Anderson, T.L., Scarborough, B. & Watson, L.R. 2013, 'Water Crises, Water Rights, and Water Markets* A2 - Shogren, Jason F', *Encyclopedia of Energy, Natural Resource, and Environmental Economics*, Elsevier, Waltham, pp. 248-54.

Ang, W.L., Wahab Mohammad, A., Johnson, D. & Hilal, N. 2019, 'Forward osmosis research trends in desalination and wastewater treatment: A review of research trends over the past decade', *Journal of Water Process Engineering*, vol. 31, p. 100886.

Attarde, D., Jain, M., Chaudhary, K. & Gupta, S.K. 2015, 'Osmotically driven membrane processes by using a spiral wound module — Modeling, experimentation and numerical parameter estimation', *Desalination*, vol. 361, pp. 81-94.

Attarde, D., Jain, M. & Gupta, S.K. 2016, 'Modeling of a forward osmosis and a pressure-retarded osmosis spiral wound module using the Spiegler-Kedem model and experimental validation', *Separation and Purification Technology*, vol. 164, pp. 182-97.

Attarde, D., Jain, M., Singh, P.K. & Gupta, S.K. 2017, 'Energy-efficient seawater desalination and wastewater treatment using osmotically driven membrane processes', *Desalination*, vol. 413, pp. 86-100.

Awad, A.M., Jalab, R., Minier-Matar, J., Adham, S., Nasser, M.S. & Judd, S.J. 2019, 'The status of forward osmosis technology implementation', *Desalination*, vol. 461, pp. 10-21.

Bamaga, O.A., Yokochi, A., Zabara, B. & Babaqi, A.S. 2011, 'Hybrid FO/RO desalination system: Preliminary assessment of osmotic energy recovery and designs of new FO membrane module configurations', *Desalination*, vol. 268, no. 1, pp. 163-9.

Banchik, L.D., Weiner, A.M., Al-Anzi, B. & Lienhard V, J.H. 2016, 'System scale analytical modeling of forward and assisted forward osmosis mass exchangers with a case study on fertigation', *Journal of Membrane Science*, vol. 510, pp. 533-45.

Beddington, J. 2011, 'Foresight: The future of food and farming', *London: The Government Office for Science*.

Belfort, G. 1988, 'Membrane modules: comparison of different configurations using fluid mechanics', *Journal of Membrane Science*, vol. 35, no. 3, pp. 245-70.

Benton, C. & Bakajin, O. 2017, *Separation systems, elemets, and methods for separation utilizing stacked membrane and spacers*, United States US 9,636,635 B2.

Cath, T.Y., Childress, A.E. & Elimelech, M. 2006, 'Forward osmosis: Principles, applications, and recent developments', *Journal of Membrane Science*, vol. 281, no. 1, pp. 70-87.

Chekli, L., Phuntsho, S., Kim, J.E., Kim, J., Choi, J.Y., Choi, J.-S., Kim, S., Kim, J.H., Hong, S., Sohn, J. & Shon, H.K. 2016, 'A comprehensive review of hybrid forward osmosis systems: Performance, applications and future prospects', *Journal of Membrane Science*, vol. 497, pp. 430-49.

Chen, L., Gu, Y., Cao, C., Zhang, J., Ng, J.-W. & Tang, C. 2014, 'Performance of a submerged anaerobic membrane bioreactor with forward osmosis membrane for low-strength wastewater treatment', *Water Research*, vol. 50, pp. 114-23.

Choi, B.G., Zhan, M., Shin, K., Lee, S. & Hong, S. 2017, 'Pilot-scale evaluation of FO-RO osmotic dilution process for treating wastewater from coal-fired power plant integrated with seawater desalination', *Journal of Membrane Science*, vol. 540, pp. 78-87.

Chong, T.H., Wong, F.S. & Fane, A.G. 2008, 'Implications of critical flux and cake enhanced osmotic pressure (CEOP) on colloidal fouling in reverse osmosis: Experimental observations', *Journal of Membrane Science*, vol. 314, no. 1, pp. 101-11.

Chou, S., Shi, L., Wang, R., Tang, C.Y., Qiu, C. & Fane, A.G. 2010, 'Characteristics and potential applications of a novel forward osmosis hollow fiber membrane', *Desalination*, vol. 261, no. 3, pp. 365-72.

Corzo, B., de la Torre, T., Sans, C., Ferrero, E. & Malfeito, J.J. 2017, 'Evaluation of draw solutions and commercially available forward osmosis membrane modules for wastewater reclamation at pilot scale', *Chemical Engineering Journal*, vol. 326, pp. 1-8.

Da Costa, A.R., Fane, A.G. & Wiley, D.E. 1994, 'Spacer characterization and pressure drop modelling in spacer-filled channels for ultrafiltration', *Journal of Membrane Science*, vol. 87, no. 1, pp. 79-98.

Deshmukh, A., Yip, N.Y., Lin, S. & Elimelech, M. 2015, 'Desalination by forward osmosis: Identifying performance limiting parameters through module-scale modeling', *Journal of Membrane Science*, vol. 491, pp. 159-67.

DOW 2020, *FilmTec™ Reverse Osmosis Membranes Technical Manual*.

Field, R.W., Siddiqui, F.A., Ang, P. & Wu, J.J. 2017, 'Analysis of the influence of module construction upon forward osmosis performance', *Desalination*.

Field, R.W. & Wu, J.J. 2018, 'On boundary layers and the attenuation of driving forces in forward osmosis and other membrane processes', *Desalination*, vol. 429, pp. 167-74.

Gu, B., Kim, D.Y., Kim, J.H. & Yang, D.R. 2011a, 'Mathematical model of flat sheet membrane modules for FO process: Plate-and-frame module and spiral-wound module', *Journal of Membrane Science*, vol. 379, no. 1, pp. 403-15.

Gu, B., Kim, D.Y., Kim, J.H. & Yang, D.R. 2011b, 'Mathematical model of flat sheet membrane modules for FO process: Plate-and-frame module and spiral-wound module', *Journal of Membrane Science*, vol. 379, no. 1–2, pp. 403-15.

Hou, D., Lu, L. & Ren, Z.J. 2016, 'Microbial fuel cells and osmotic membrane bioreactors have mutual benefits for wastewater treatment and energy production', *Water research*, vol. 98, pp. 183-9.

Hunt, C.E. 2007, *Thirsty planet: Strategies for sustainable water management*, Academic Foundation.

Hutchings, N.R., Appleton, E.W. & McGinnis, R.A. 2010, 'Making High Quality Frac Water out of Oilfield Waste', paper presented to the SPE Annual Technical Conference and Exhibition, Florence, Italy, 2010/1/1/, <https://doi.org/10.2118/135469-MS>.

Ibrar, I., Altaee, A., Zhou, J.L., Naji, O. & Khanafer, D. 2020, 'Challenges and potentials of forward osmosis process in the treatment of wastewater', *Critical Reviews in Environmental Science and Technology*, vol. 50, no. 13, pp. 1339-83.

Im, S.-J., Go, G.-W., Lee, S.-H., Park, G.-H. & Jang, A. 2016, 'Performance evaluation of two-stage spiral wound forward osmosis elements at various operation conditions', *Desalination and Water Treatment*, vol. 57, no. 51, pp. 24583-94.

Im, S.-J., Jeong, S. & Jang, A. 2018, 'Feasibility evaluation of element scale forward osmosis for direct connection with reverse osmosis', *Journal of Membrane Science*, vol. 549, pp. 366-76.

Im, S.J., Jeong, S., Jeong, S. & Jang, A. 2020, 'Techno-economic evaluation of an element-scale forward osmosis-reverse osmosis hybrid process for seawater desalination', *Desalination*, vol. 476, p. 114240.

Jeon, J., Jung, J., Lee, S., Choi, J.Y. & Kim, S. 2018, 'A simple modeling approach for a forward osmosis system with a spiral wound module', *Desalination*, vol. 433, pp. 120-31.

Johnson, D.J., Suwaileh, W.A., Mohammed, A.W. & Hilal, N. 2018, 'Osmotic's potential: An overview of draw solutes for forward osmosis', *Desalination*, vol. 434, pp. 100-20.

Jung, D.H., Lee, J., Kim, D.Y., Lee, Y.G., Park, M., Lee, S., Yang, D.R. & Kim, J.H. 2011, 'Simulation of forward osmosis membrane process: Effect of membrane orientation and flow direction of feed and draw solutions', *Desalination*, vol. 277, no. 1, pp. 83-91.

Khayet, M., Sanmartino, J.A., Essalhi, M., García-Payo, M.C. & Hilal, N. 2016, 'Modeling and optimization of a solar forward osmosis pilot plant by response surface methodology', *Solar Energy*, vol. 137, pp. 290-302.

Kim, J., Blandin, G., Phuntsho, S., Verliefde, A., Le-Clech, P. & Shon, H. 2017, 'Practical considerations for operability of an 8″ spiral wound forward osmosis module: Hydrodynamics, fouling behaviour and cleaning strategy', *Desalination*, vol. 404, pp. 249-58.

Kim, J.E., Phuntsho, S., Ali, S.M., Choi, J.Y. & Shon, H.K. 2018, 'Forward osmosis membrane modular configurations for osmotic dilution of seawater by forward osmosis and reverse osmosis hybrid system', *Water Research*, vol. 128, pp. 183-92.

Kim, J.E., Phuntsho, S., Chekli, L., Hong, S., Ghaffour, N., Leiknes, T., Choi, J.Y. & Shon, H.K. 2017, 'Environmental and economic impacts of fertilizer drawn forward osmosis and nanofiltration hybrid system', *Desalination*, vol. 416, pp. 76-85.

Kim, J.E., Phuntsho, S., Lotfi, F. & Shon, H.K. 2015, 'Investigation of pilot-scale 8040 FO membrane module under different operating conditions for brackish water desalination', *Desalination and Water Treatment*, vol. 53, no. 10, pp. 2782-91.

Kim, Y.C. & Park, S.-J. 2011, 'Experimental Study of a 4040 Spiral-Wound Forward-Osmosis Membrane Module', *Environmental Science & Technology*, vol. 45, no. 18, pp. 7737-45.

Kook, S., Lee, C., Nguyen, T.T., Lee, J., Shon, H.K. & Kim, I.S. 2018, 'Serially connected forward osmosis membrane elements of pressure-assisted forward osmosis-reverse osmosis hybrid system: Process performance and economic analysis', *Desalination*, vol. 448, pp. 1-12.

Kumar, M.D., Bassi, N. & Singh, O.P. 2020, 'Rethinking on the methodology for assessing global water and food challenges', *International Journal of Water Resources Development*, vol. 36, no. 2-3, pp. 547-64.

Lambrechts, R. & Sheldon, M.S. 2019, 'Performance and energy consumption evaluation of a fertiliser drawn forward osmosis (FDFO) system for water recovery from brackish water', *Desalination*, vol. 456, pp. 64-73.

Lee, J. & Ghaffour, N. 2019, 'Predicting the performance of large-scale forward osmosis module using spatial variation model: Effect of operating parameters including temperature', *Desalination*, vol. 469, p. 114095.

Lee, S., Choi, J., Park, Y.-G., Shon, H., Ahn, C.H. & Kim, S.-H. 2019, 'Hybrid desalination processes for beneficial use of reverse osmosis brine: Current status and future prospects', *Desalination*, vol. 454, pp. 104-11.

Lee, S. & Kim, Y.C. 2018, 'Performance analysis of plate-and-frame forward osmosis membrane elements and implications for scale-up design', *Journal of Membrane Science*, vol. 550, pp. 219-29.

Leflaive, X., Witmer, M., Martin-Hurtado, R., Bakker, M., Kram, T., Bouwman, L., Visser, H., Bouwman, A., Hilderink, H. & Kim, K. 2012, *OECD Environmental Outlook to 2050*.

Li, D., Wang, R. & Chung, T.-S. 2004, 'Fabrication of lab-scale hollow fiber membrane modules with high packing density', *Separation and Purification Technology*, vol. 40, no. 1, pp. 15-30.

Li, F., Cheng, Q., Tian, Q., Yang, B. & Chen, Q. 2016, 'Biofouling behavior and performance of forward osmosis membranes with bioinspired surface modification in osmotic membrane bioreactor', *Bioresour Technol*, vol. 211, pp. 751-8.

Li, H. 2017, *Global trends & challenges in water science, research and management*, IWA Publishing.

Li, Z., Siddiqi, A., Anadon, L.D. & Narayanamurti, V. 2018, 'Towards sustainability in water-energy nexus: Ocean energy for seawater desalination', *Renewable and Sustainable Energy Reviews*, vol. 82, pp. 3833-47.

Lim, S., Tran, V.H., Akther, N., Phuntsho, S. & Shon, H.K. 2019, 'Defect-free outer-selective hollow fiber thin-film composite membranes for forward osmosis applications', *Journal of Membrane Science*, vol. 586, pp. 281-91.

Lin, S. 2016, 'Mass transfer in forward osmosis with hollow fiber membranes', *Journal of Membrane Science*, vol. 514, pp. 176-85.

Liyanaarachchi, S., Jegatheesan, V., Muthukumaran, S., Gray, S. & Shu, L. 2016, 'Mass balance for a novel RO/FO hybrid system in seawater desalination', *Journal of Membrane Science*, vol. 501, pp. 199-208.

Lu, Y. & He, Z. 2015, 'Mitigation of salinity buildup and recovery of wasted salts in a hybrid osmotic membrane bioreactor–electrodialysis system', *Environmental science & technology*, vol. 49, no. 17, pp. 10529-35.

Luo, W., Phan, H.V., Li, G., Hai, F.I., Price, W.E., Elimelech, M. & Nghiem, L.D. 2017, 'An Osmotic Membrane Bioreactor–Membrane Distillation System for Simultaneous Wastewater Reuse and Seawater Desalination: Performance and Implications', *Environmental science & technology*, vol. 51, no. 24, pp. 14311-20.

Luo, W., Xie, M., Hai, F.I., Price, W.E. & Nghiem, L.D. 2016, 'Biodegradation of cellulose triacetate and polyamide forward osmosis membranes in an activated sludge bioreactor: Observations and implications', *Journal of Membrane Science*, vol. 510, pp. 284-92.

Mazlan, N.M., Peshev, D. & Livingston, A.G. 2016a, 'Energy consumption for desalination - A comparison of forward osmosis with reverse osmosis, and the potential for perfect membranes', *Desalination*, vol. 377, pp. 138-51.

Mazlan, N.M., Peshev, D. & Livingston, A.G. 2016b, 'Energy consumption for desalination — A comparison of forward osmosis with reverse osmosis, and the potential for perfect membranes', *Desalination*, vol. 377, pp. 138-51.

Molden, D. 2013, *Water for food water for life: a comprehensive assessment of water management in agriculture*, Routledge.

Mondal, S., Field, R.W. & Wu, J.J. 2017, 'Novel approach for sizing forward osmosis membrane systems', *Journal of Membrane Science*, vol. 541, pp. 321-8.

Nikbakht Fini, M., Madsen, H.T., Sørensen, J.L. & Muff, J. 2020, 'Moving from lab to pilot scale in forward osmosis for pesticides rejection using aquaporin membranes', *Separation and Purification Technology*, vol. 240, p. 116616.

Park, K., Kim, J., Yang, D.R. & Hong, S. 2020, 'Towards a low-energy seawater reverse osmosis desalination plant: A review and theoretical analysis for future directions', *Journal of Membrane Science*, vol. 595, p. 117607.

Peñate, B. & García-Rodríguez, L. 2012, 'Current trends and future prospects in the design of seawater reverse osmosis desalination technology', *Desalination*, vol. 284, pp. 1-8.

Phuntsho, S., Hong, S., Elimelech, M. & Shon, H.K. 2014, 'Osmotic equilibrium in the forward osmosis process: Modelling, experiments and implications for process performance', *Journal of Membrane Science*, vol. 453, pp. 240-52.

Phuntsho, S., Kim, J.E., Hong, S., Ghaffour, N., Leiknes, T., Choi, J.Y. & Shon, H.K. 2017, 'A closed-loop forward osmosis-nanofiltration hybrid system: Understanding process implications through full-scale simulation', *Desalination*, vol. 421, pp. 169-78.

Phuntsho, S., Kim, J.E., Johir, M.A.H., Hong, S., Li, Z., Ghaffour, N., Leiknes, T. & Shon, H.K. 2016, 'Fertiliser drawn forward osmosis process: Pilot-scale desalination of mine impaired water for fertigation', *Journal of Membrane Science*, vol. 508, pp. 22-31.

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*, vol. 375, no. 1, pp. 172-81.

Programme, U.N.H.S. 2003, *Water and sanitation in the world's cities: Local action for global goals*, Earthscan.

Qasim, M., Badrelzaman, M., Darwish, N.N., Darwish, N.A. & Hilal, N. 2019, 'Reverse osmosis desalination: A state-of-the-art review', *Desalination*, vol. 459, pp. 59-104.

Ren, J., Chowdhury, M.R., Xia, L., Ma, C., Bollas, G.M. & McCutcheon, J.R. 2020, 'A computational fluid dynamics model to predict performance of hollow fiber membrane modules in forward osmosis', *Journal of Membrane Science*, vol. 603, p. 117973.

Ren, J. & McCutcheon, J.R. 2018, 'A new commercial biomimetic hollow fiber membrane for forward osmosis', *Desalination*, vol. 442, pp. 44-50.

Sengur-Tasdemir, R., Sayinli, B., Urper, G.M., Tutuncu, H.E., Gul-Karaguler, N., Ates-Genceli, E., Tarabara, V.V. & Koyuncu, I. 2018, 'Hollow fiber nanofiltration membranes with integrated aquaporin Z', *New Journal of Chemistry*, vol. 42, no. 21, pp. 17769-78.

Seo, J., Kim, Y.M., Chae, S.H., Lim, S.J., Park, H. & Kim, J.H. 2019, 'An optimization strategy for a forward osmosis-reverse osmosis hybrid process for wastewater reuse and seawater desalination: A modeling study', *Desalination*, vol. 463, pp. 40-9.

Shaffer, D.L., Werber, J.R., Jaramillo, H., Lin, S. & Elimelech, M. 2015, 'Forward osmosis: Where are we now?', *Desalination*, vol. 356, pp. 271-84.

Shibuya, M., Yasukawa, M., Goda, S., Sakurai, H., Takahashi, T., Higa, M. & Matsuyama, H. 2016, 'Experimental and theoretical study of a forward osmosis hollow fiber membrane module with a cross-wound configuration', *Journal of Membrane Science*, vol. 504, pp. 10-9.

Shiklomanov, I.A. 1993, 'World fresh water resources. Water in Crisis: A Guide to the World's Fresh Water Resources', Oxford University Press, New York.

Shon, H.K., Phuntsho, S., Zhang, T.C. & Surampalli, R.Y. 2015, *Forward Osmosis: Fundamentals and Applications*, American Society of Civil Engineers.

Siddiqui, F.A., She, Q., Fane, A.G. & Field, R.W. 2018, 'Exploring the differences between forward osmosis and reverse osmosis fouling', *Journal of Membrane Science*, vol. 565, pp. 241-53.

Sim, S.T.V., Krantz, W.B., Chong, T.H. & Fane, A.G. 2015, 'Online monitor for the reverse osmosis spiral wound module — Development of the canary cell', *Desalination*, vol. 368, pp. 48-59.

Sivertsen, E., Holt, T., Thelin, W. & Brekke, G. 2013, 'Pressure retarded osmosis efficiency for different hollow fibre membrane module flow configurations', *Desalination*, vol. 312, pp. 107-23.

Song, M., Im, S.-J., Jeong, S. & Jang, A. 2018, 'Evaluation of an element-scale plate-type forward osmosis: Effect of structural parameters and operational conditions', *Desalination*, vol. 430, pp. 15-23.

Stefan, M.I. 2017, *Advanced oxidation processes for water treatment: fundamentals and applications*, IWA Publishing.

Suwaileh, W., Pathak, N., Shon, H. & Hilal, N. 2020, 'Forward osmosis membranes and processes: A comprehensive review of research trends and future outlook', *Desalination*, vol. 485, p. 114455.

Teklu, H., Gautam, D.K. & Subbiah, S. 2020, 'Axial flow hollow fiber forward osmosis module analysis for optimum design and operating conditions in desalination applications', *Chemical Engineering Science*, vol. 216, p. 115494.

Thompson, N.A. & Nicoll, P.G. 2011, 'Forward Osmosis Desalination: A Commercial Reality,' paper presented to the IDA World Congress, Perth Convention and Exhibition Centre (PCEC), Perth, Western Australia, September 4-9, 2011

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*, vol. 444, pp. 523-38.

TOYOBO 2016, 'Toyobo membrane module for forward osmosis'.

Tran, V.H., Lim, S., Han, D.S., Pathak, N., Akther, N., Phuntsho, S., Park, H. & Shon, H.K. 2019, 'Efficient fouling control using outer-selective hollow fiber thin-film composite membranes for osmotic membrane bioreactor applications', *Bioresource Technology*, vol. 282, pp. 9-17.

Trenberth, K.E., Smith, L., Qian, T., Dai, A. & Fasullo, J. 2007, 'Estimates of the Global Water Budget and Its Annual Cycle Using Observational and Model Data', *Journal of Hydrometeorology*, vol. 8, no. 4, pp. 758-69.

Valladares Linares, R., Li, Z., Sarp, S., Bucs, S.S., Amy, G. & Vrouwenvelder, J.S. 2014, 'Forward osmosis niches in seawater desalination and wastewater reuse', *Water Research*, vol. 66, pp. 122-39.

Valladares Linares, R., Li, Z., Yangali-Quintanilla, V., Ghaffour, N., Amy, G., Leiknes, T. & Vrouwenvelder, J.S. 2016, 'Life cycle cost of a hybrid forward osmosis – low pressure reverse osmosis system for seawater desalination and wastewater recovery', *Water Research*, vol. 88, pp. 225-34.

Volpin, F., Fons, E., Chekli, L., Kim, J.E., Jang, A. & Shon, H.K. 2018, 'Hybrid forward osmosis-reverse osmosis for wastewater reuse and seawater desalination: Understanding the optimal feed solution to minimise fouling', *Process Safety and Environmental Protection*, vol. 117, pp. 523-32.

Wan, C.F., Yang, T., Lipscomb, G.G., Stookey, D.J. & Chung, T.-S. 2017, 'Design and fabrication of hollow fiber membrane modules', *Journal of Membrane Science*, vol. 538, pp. 96-107.

Wang, Y.-N., Goh, K., Li, X., Setiawan, L. & Wang, R. 2018, 'Membranes and processes for forward osmosis-based desalination: Recent advances and future prospects', *Desalination*, vol. 434, pp. 81-99.

Wang, Y.-N., Järvelä, E., Wei, J., Zhang, M., Kyllönen, H., Wang, R. & Tang, C.Y. 2016, 'Gypsum scaling and membrane integrity of osmotically driven membranes: The effect of membrane materials and operating conditions', *Desalination*, vol. 377, pp. 1-10.

Wang, Y., Zhang, M., Liu, Y., Xiao, Q. & Xu, S. 2016, 'Quantitative evaluation of concentration polarization under different operating conditions for forward osmosis process', *Desalination*, vol. 398, pp. 106-13.

Watkins, K. 2006a, *Beyond Scarcity: Power, Poverty and the Global Water Crisis* UNDP

Watkins, K. 2006b, *Human Development Report 2006 - Beyond Scarcity: Power, Poverty and the Global Water Crisis*

Wu, T.Y., Mohammad, A.W., Lim, S.L., Lim, P.N. & Hay, J.X.W. 2013, 'Recent Advances in the Reuse of Wastewaters for Promoting Sustainable Development', in S.K. Sharma & R. Sanghi (eds), *Wastewater Reuse and Management*, Springer Netherlands, Dordrecht, pp. 47-103.

WWAP 2012, *The United Nations World Water Development Report 4: Managing Water under Uncertainty and Risk*, Paris.

Xiao, D., Li, W., Chou, S., Wang, R. & Tang, C.Y. 2012, 'A modeling investigation on optimizing the design of forward osmosis hollow fiber modules', *Journal of Membrane Science*, vol. 392-393, pp. 76-87.

Xie, M., Nghiem, L.D., Price, W.E. & Elimelech, M. 2014, 'Toward Resource Recovery from Wastewater: Extraction of Phosphorus from Digested Sludge Using a Hybrid Forward Osmosis–Membrane Distillation Process', *Environmental Science & Technology Letters*, vol. 1, no. 2, pp. 191-5.

Xu, Y., Peng, X., Tang, C.Y., Fu, Q.S. & Nie, S. 2010, 'Effect of draw solution concentration and operating conditions on forward osmosis and pressure retarded osmosis performance in a spiral wound module', *Journal of Membrane Science*, vol. 348, no. 1, pp. 298-309.

Xue, W., Yamamoto, K., Tobino, T. & Ratanatamskul, C. 2016, 'Modeling prediction of the process performance of seawater-driven forward osmosis for nutrients enrichment: Implication for membrane module design and system operation', *Journal of Membrane Science*, vol. 515, pp. 7-21.

Yadav, S., Saleem, H., Ibrar, I., Naji, O., Hawari, A.A., Alanezi, A.A., Zaidi, S.J., Altaee, A. & Zhou, J. 2020, 'Recent developments in forward osmosis membranes using carbon-based nanomaterials', *Desalination*, vol. 482, p. 114375.

Yap, W.J., Zhang, J., Lay, W.C., Cao, B., Fane, A.G. & Liu, Y. 2012, 'State of the art of osmotic membrane bioreactors for water reclamation', *Bioresour Technol*, vol. 122, pp. 217-22.

Yip, N.Y., Tiraferri, A., Phillip, W.A., Schiffman, J.D. & Elimelech, M. 2010, 'High performance thin-film composite forward osmosis membrane', *Environmental science & technology*, vol. 44, no. 10, pp. 3812-8.

Zhang, Y., Pinoy, L., Meesschaert, B. & Van der Bruggen, B. 2013, 'A Natural Driven Membrane Process for Brackish and Wastewater Treatment: Photovoltaic Powered ED and FO Hybrid System', *Environmental Science & Technology*, vol. 47, no. 18, pp. 10548-55.

Zhao, P., Gao, B., Yue, Q., Liu, S. & Shon, H.K. 2016, 'Effect of high salinity on the performance of forward osmosis: Water flux, membrane scaling and removal efficiency', *Desalination*, vol. 378, pp. 67-73.

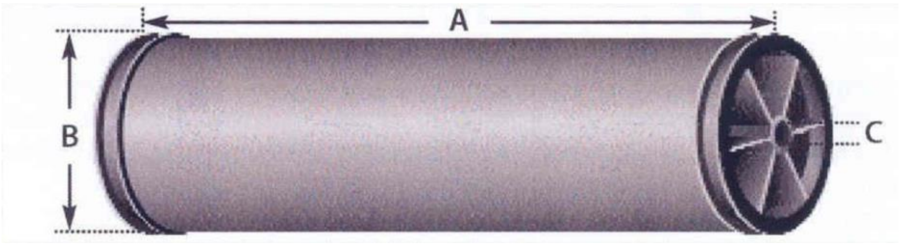
Zhao, P., Yue, Q., Gao, B., Kong, J., Rong, H., Liu, P., Shon, H.K. & Li, Q. 2014, 'Influence of different ion types and membrane orientations on the forward osmosis performance', *Desalination*, vol. 344, pp. 123-8.

Ziolkowska, J.R. 2015, 'Is Desalination Affordable?—Regional Cost and Price Analysis', *Water Resources Management*, vol. 29, no. 5, pp. 1385-97.

Appendices

A1 (Supplementary information for spiral wound module)

Table A1.1 Parameters used for simulating an FO system using spiral wound module.

Operating Conditions		
Parameters	Unit	Value
Draw solution flowrate	L/min	0.25 - 20
Feed solution flowrate	L/min	35 - 65
Draw solution (NaCl Solution) concentration	M	0.6
Feed Solution (NaCl Solution) concentration	M	0.02
Operating temperature	℃	25
Design Parameters		
Module Type		8040 Spiral wound
Nominal dimensions		
		
A : 40 inches;	B = 8 inches;	C = 1.125 inches
Membrane type		TFC
Water permeability coefficient, A	$Lm^{-2}h^{-1}bar^{-1}$	5.54
Solute permeability coefficient, B	$Lm^{-2}h^{-1}bar^{-1}$	2.26
Structural parameter, S	µm	300
Number of membrane sheets		24
Length of membrane sheet	mm	935
Width of membrane sheet	mm	965
Glue line width	mm	55
Central glue line length	mm	450
Effective membrane area	m^2	15
Spacer type		Diamond
Spacer thickness	mm	1.19
Feed spacer filament length	mm	5
Permeate career thickness	mm	0.2
Design Constrains		
Flexibility (Fl)		0.05 (5%)
Minimum dilution for adding element (ΔC_E)	M	0.001

A2 (Suplimenary information for hollow fiber module)

Table A2.1 Parameters used for designing an FO system using hollow fibre module.

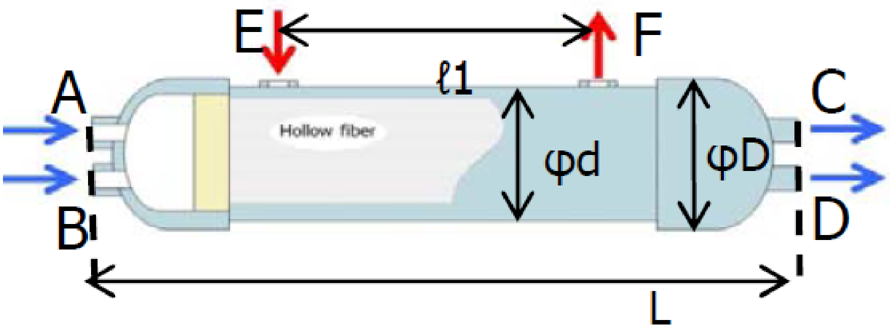
Operating Conditions		
Parameters	Unit	Value
Draw solution flowrate	L/min	0.25 - 1
Feed solution flowrate	L/min	2 -3
Draw solution (NaCl Solution) concentration	M	0.6
Feed Solution (NaCl Solution) concentration	M	0.02
Operating temperature	℃	25
Membrane Element Dimensions		
Module Type		HPC3205 Hollow Fibre
Nominal dimensions		
		
Water permeability coefficient, A	$Lm^{-2}h^{-1}bar^{-1}$	0.27
Solute permeability coefficient, B	$Lm^{-2}h^{-1}$	0.035
Structural parameter, S	µm	1024
Total Membrane Surface Area	m^2	31.5
Module diameter	m	7.57E-2
Module active length	m	0.56
Fiber outer diameter	*µm*	175
Fiber inner diameter	*µm*	85
Porosity		0.45
Membrane material		CTA
Design Objectives		
Production capacity	m^3/day	1,000
Final DS Conc. (FDC)	(M)	0.25 – 0.15

Table A2.2 Parameters used for designing an FO system using hollow fibre module.

FS inlet flowrate	**Pressure drop (Model)**	**Pressure drop (Expt)**	**Error**	**RMS Error Average**
L/min	(Bar)			
2.59	3.08	3.16	-0.078	0.071
2.44	2.78	2.84	-0.054	
2.29	2.50	2.46	0.044	
1.99	1.98	1.96	0.025	
1.68	1.50	1.42	0.089	
1.38	1.10	0.97	0.137	

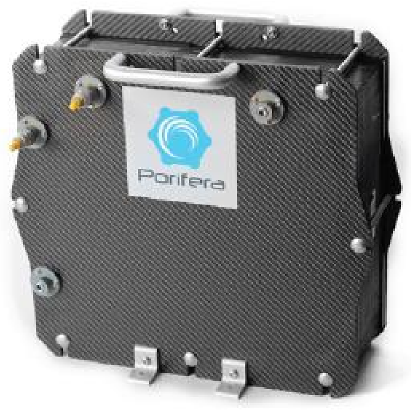
Pressure drop value was calculated using the Ergun model,

$$\Delta P = \left[\frac{K_1(1-\varepsilon)^2\mu U_{FS}}{\varepsilon^3(1.5d_0)^2} + \frac{K_2(1-\varepsilon)\rho {U_{FS}}^2}{\varepsilon^3(1.5d_0)^2}\right] L$$

Minimum RMS error (0.071) for the Ergun Equation was found when K_1 = 150 and K_2 = 45.

A3 (Suplimenary information for plate and frame type membrane element)

Table A3.1 Parameters used for designing and optimizing an FO plant using a commercial plate and frame type membrane element.

Operating Conditions		
Parameters	Unit	Value
Draw solution flowrate	L/min	0.5 - 20
Feed solution flowrate	L/min	10 - 40
Draw solution (NaCl Solution) concentration	M	0.6
Feed Solution (NaCl Solution) concentration	M	0.02
Operating temperature	℃	25
Membrane Element Dimensions		
Module Type		PFO 100 Plate and Frame
Nominal dimensions		
		
Water permeability coefficient, A	$Lm^{-2}h^{-1}bar^{-1}$	2.22
Solute permeability coefficient, B	$Lm^{-2}h^{-1}$	0.49
Structural parameter, S	μm	269
Number of membrane plate cell		33
Number of membrane sheets		66
Length of membrane sheet	mm	380
Width of membrane sheet	mm	280
Effective membrane area	m^2	7
Spacer type		Fishnet
Spacer thickness	mm	0.76
Design Objectives		
Production capacity	m^3/day	1,000
Recovery rate (RR)	%	40 – 60
Final DS Conc. (FDC)	(M)	0.25 – 0.15

A4 (MATLAB Simulation Code for the Software Development)

```
function main = FOSA_Gui_PF_V1()

clc; clear all;

global Flag Prog_Location Prog_Name Prog_ext
Bin_Location Lib_Location Font_tab_title Font_title
Font_large Font_medium Font_small
global Fig_H1 Col_H1 Col_H2 Col_H3 Col_H4 Col_H5
Col_H6 Col_H7 Col_H8 m_PanelWidth m_PanelHeight Tbl_FS
Tbl_DS
global Tab1_H1 Tab1_H2 Tab2_H1 Tab2_H2 Tab3_H1 Tab3_H2
Tab4_H1 Tab4_H2
global Tab5_H1 Tab5_H2 Tab6_H1 Tab6_H2 Tab7_H1 Tab7_H2
global Tab2_edit_temp Tab2_edit_pH Tab2_edit_pOH
Tab2_TDS_text Tab2_OP_FS Tab2_FS_disp
global Tab2_Balance_text Tab2_Cations_text
Tab2_Anions_text
global Tab3_DS_disp Tab3_edit_temp Tab3_edit_pH
Tab3_edit_pOH Tab3_TDS_text Tab3_OP_DS
global Tab3_Balance_text Tab3_Cations_text
Tab3_Anions_text
global Tab4_edit_MemType Tab4_edit_ModType
Tab4_edit_SpacerType Tab4_Q_D_in_i Tab4_Q_D_in_f
Tab4_Q_F_in_i
global Tab4_del_Q_D_in Tab4_C_F_in Tab4_C_D_in
Tab4_edit_Rec Tab4_edit_Rec_Flex Tab4_edit_Conc
Tab4_edit_Conc_Flex
global Tab4_status_disp Tab_4_Panel_Status
Tab4_ax_status Tab4_opt_Q_DS_in Tab4_opt_NE
Tab4_opt_Rec Tab4_opt_C_DS Tab4_opt_OPR
global Tab5_Prod_Cap Tab5_Rec Tab5_C_D_f Tab5_El_Mod
Tab5_Mod_Type Tab5_Mem_Type Tab5_Area Tab5_Sp_Type
Tab5_Sp_thickness Tab5_pan_Sys
global Tab5_NE Tab5_FS_Type Tab5_FS_Flr Tab5_FS_C
Tab5_FS_OP Tab5_DS_Type Tab5_DS_Flr Tab5_DS_C
Tab5_DS_OP hImageAxes
global Tab_6_Panel_Sim_Out Tab6_NPV Tab6_TNE
Tab6_TArea Tab6_Q_DS Tab6_Q_P Tab6_C_D_f
global Tab7_Unit_Prod_Cost Tab7_Unit_Opex
Tab7_Unit_Capex Tab7_Sp_En_Cons Tab7_Loan_ROI
Tab7_Loan_Capex Tab7_Loan_Opex Tab7_Net_Opex
global Tab7_Net_Capex Tab7_Net_En_Cons
Tab7_Op_Pwr_Cost Tab7_Op_FS_Pump_Rep
```

```
Tab7_Op_DS_Pump_Rep Tab7_Op_El_Rep Tab7_Op_PV_Rep
Tab7_Op_Emp_Exp
global Tab7_Cap_El_Price Tab7_Cap_PV_Price
Tab7_Cap_DS_Pmp_Cap Tab7_Cap_DS_Pmp_Price
Tab7_Cap_Land_Cost Tab7_Cap_Inf_Cost Tab7_FS_Pmp_Cap
Tab7_Cap_FS_Pmp_Price
global Tab7_Plant_Life Tab7_El_Life Tab7_PV_Life
Tab7_Pmp_Life Tab7_Op_Time

global MCL_Ammonium MCL_Potassium MCL_Sodium
MCL_Magnesium MCL_Calcium MCL_Strontium
global MCL_Barium MCL_Boron MCL_Carbonate
MCL_Bicarbonate MCL_Nitrate MCL_Chloride
global MCL_Fluoride MCL_Sulfate MCL_Silica MCL_TDS

global VC_Ammonium VC_Potassium VC_Sodium VC_Magnesium
VC_Calcium VC_Strontium VC_Barium
global VC_Boron VC_Carbonate VC_Bicarbonate VC_Nitrate
VC_Chloride VC_Fluoride VC_Sulfate VC_Silica

global MW_Ammonium MW_Potassium MW_Sodium MW_Magnesium
MW_Calcium MW_Strontium MW_Barium  MW_Boron
global MW_Carbonate MW_Bicarbonate MW_Nitrate
MW_Chloride MW_Fluoride MW_Sulfate MW_Silica

global C_ativity

Flag = 0;

MCL_Ammonium = 0; MCL_Potassium = 0; MCL_Sodium = 0;
MCL_Magnesium = 0.05; MCL_Calcium = 0;
MCL_Strontium = 0; MCL_Barium = 2; MCL_Boron = 0;
MCL_Carbonate = 0; MCL_Bicarbonate = 0;
MCL_Nitrate = 10; MCL_Chloride = 250; MCL_Fluoride =
0; MCL_Sulfate = 250; MCL_Silica = 0;
MCL_TDS = 500;

VC_Ammonium = 0; VC_Potassium = 1; VC_Sodium = 1;
VC_Magnesium = 2; VC_Calcium = 2; VC_Strontium = 2;
VC_Barium = 2; VC_Boron = 0; VC_Carbonate = 2;
VC_Bicarbonate = 1; VC_Nitrate = 1; VC_Chloride = 1;
VC_Fluoride = 1; VC_Sulfate = 2; VC_Silica = 0;

MW_Ammonium = 0; MW_Potassium = 39.1; MW_Sodium =
22.99; MW_Magnesium = 24.31; MW_Calcium = 40.08;
MW_Strontium = 87.62; MW_Barium = 137.34; MW_Boron =
0; MW_Carbonate = 60; MW_Bicarbonate = 61;
```

```
MW_Nitrate = 14; MW_Chloride = 35.45; MW_Fluoride =
19; MW_Sulfate = 96; MW_Silica = 0;

C_ativity = 0.161121287007952;

%Locacation for current matlab file
[folder, name, ext] =
fileparts(which('FOSA_Gui_PF_V1'));
Prog_Location = folder;
Prog_Name = name;
Prog_ext = ext;

%location for bin flder
Bin_Location = strcat(Prog_Location, '\bin');

%location for Lib flder
Lib_Location = strcat(Prog_Location, '\Lib');

Tab1_Lab = 'Project Info';
Tab2_Lab = 'Feed Solution Analysis';
Tab3_Lab = 'Draw Solution Analysis';
Tab4_Lab = 'Optimization';
Tab5_Lab = 'Configuration';
Tab6_Lab = 'Simulation';
Tab7_Lab = 'Economic Analysis';

% Set the color varables.
Col_H1 = [0.94  0.94  0.94];            % TMC (Typical
Matlab Color) - Selected tab color
Col_H2 = 0.9*Col_H1;                    % Light Grey -
Background color
Col_H3 = [1  1  1];                     % White Color
Col_H4 = [0  1  1];                     %Cyan
Col_H5 = [1 0 0];                       %Red
Col_H6 = [1 1 0];                       %Yellow
Col_H7 = [0 1 0];                       %Green
Col_H8 = [1 0 1];                       %magenta

% Get user screen size
 SC = get(0, 'ScreenSize');
 MaxMonitorX = SC(3);
 MaxMonitorY = SC(4);

% Set the figure window size values
MainFigScale = .8;          % Change this value to
adjust the figure size
MaxWindowX = round(MaxMonitorX*MainFigScale);
```

```
MaxWindowY = round(MaxMonitorY*MainFigScale);
XBorder = (MaxMonitorX-MaxWindowX)/2;
YBorder = (MaxMonitorY-MaxWindowY)/2;
TabOffset = 0;                  % This value offsets the
tabs inside the figure.
m_ButtonHeight = 40;

m_PanelWidth = MaxWindowX-2*TabOffset;

m_PanelHeight = MaxWindowY-1*m_ButtonHeight-
2*TabOffset;
m_ButtonWidth = round(m_PanelWidth/7);

Font_tab_title_ratio = 76.81;
Font_title_ratio = 87.79;
Font_large_ratio = 106.87;
Font_medium_ratio = 122.9;
Font_small_ratio = 153.63;

Font_tab_title = m_PanelWidth/Font_tab_title_ratio;
Font_title = m_PanelWidth/Font_title_ratio;
Font_large = m_PanelWidth/Font_large_ratio;
Font_medium = m_PanelWidth/Font_medium_ratio;
Font_small = m_PanelWidth/Font_small_ratio;

 %%   Create a main figure for the tabs
 Fig_H1 = figure('Units', 'pixels','Toolbar',
'none','Position',[ XBorder, YBorder, MaxWindowX,
MaxWindowY ],...
          'NumberTitle', 'off','Name',
'FOSA','MenuBar', 'none','Resize',
'off','DockControls', 'off','Color', Col_H2);

hold on;

%   Create menue bar for the main figure

file_H1 = uimenu(Fig_H1,'Label','File');
file_H2 = uimenu(file_H1,'Label','New Project');

set(file_H2, 'callback',{@new_menu_Callback, file_H2})

file_H3 = uimenu(file_H1,'Label','Save Project');
file_H4 = uimenu(file_H1,'Label','Export Project');
file_H5 = uimenu(file_H1,'Label','Close Project');
```

```
set(file_H5, 'callback',{@exit_menu_Callback,
file_H5})

Edit_H1 = uimenu(Fig_H1,'Label','Edit');
Edit_H2 = uimenu(Edit_H1,'Label','Cut');
Edit_H3 = uimenu(Edit_H1,'Label','Copy');
Edit_H4 = uimenu(Edit_H1,'Label','paste');

%%   Build the Tab_1 (Project Info)

% create a UIPanel
Tab1_H1 = uipanel('Units', 'pixels','Visible',
'off','Backgroundcolor', Col_H1,'BorderWidth',1, ...
          'Position', [TabOffset m_ButtonHeight
m_PanelWidth m_PanelHeight]);

% create a selection pushbutton
Tab1_H2 = uicontrol('Style', 'pushbutton','Units',
'pixels','BackgroundColor', Col_H2,'Position',...
          [TabOffset TabOffset m_ButtonWidth
m_ButtonHeight],'String',
Tab1_Lab,'HorizontalAlignment', 'center',...
          'FontName', 'arial','FontWeight',
'bold','FontSize', Font_medium);

 %%     Build the Tab_2 (Feed Solution Analysis)

 % create a UIPanel
 Tab2_H1 = uipanel('Units', 'pixels','Visible',
'off','Backgroundcolor',
Col_H1,'BorderWidth',1,'Position',...
           [TabOffset m_ButtonHeight m_PanelWidth
m_PanelHeight]);

% create a selection pushbutton
Tab2_H2 = uicontrol('Style', 'pushbutton','Units',
'pixels','BackgroundColor',
Col_H2,'Position',[TabOffset+m_ButtonWidth
TabOffset...
          m_ButtonWidth m_ButtonHeight],'String',
Tab2_Lab,'HorizontalAlignment', 'center','FontName',
'arial','FontWeight', 'bold',...
          'FontSize', Font_medium);
 %%     Build the Tab_3 (Draw Solution Analysis)

% create a UIPanel
```

```
 Tab3_H1 = uipanel('Units', 'pixels','Visible',
'off','Backgroundcolor',
Col_H1,'BorderWidth',1,'Position', [TabOffset
m_ButtonHeight ...
           m_PanelWidth m_PanelHeight]);

% create a selection pushbutton
 Tab3_H2 = uicontrol('Style', 'pushbutton','Units',
'pixels','BackgroundColor', Col_H2,'Position',
[TabOffset+2*m_ButtonWidth TabOffset...
           m_ButtonWidth m_ButtonHeight],'String',
Tab3_Lab,'HorizontalAlignment', 'center','FontName',
'arial','FontWeight', 'bold',...
           'FontSize', Font_medium);
 %%     Build the Tab_4 (Optimization)

 % create a UIPanel
 Tab4_H1 = uipanel('Units', 'pixels','Visible',
'off','Backgroundcolor',
Col_H1,'BorderWidth',1,'Position', [TabOffset
m_ButtonHeight ...
           m_PanelWidth m_PanelHeight]);

% create a selection pushbutton
Tab4_H2 = uicontrol('Style', 'pushbutton','Units',
'pixels','BackgroundColor', Col_H2,'Position',
[TabOffset+3*m_ButtonWidth TabOffset...
          m_ButtonWidth m_ButtonHeight],'String',
Tab4_Lab,'HorizontalAlignment', 'center','FontName',
'arial','FontWeight', 'bold',...
          'FontSize', Font_medium);
 %%     Build the Tab_5 (Configuration)

% create a UIPanel
Tab5_H1 = uipanel('Units', 'pixels','Visible',
'off','Backgroundcolor',
Col_H1,'BorderWidth',1,'Position', [TabOffset
m_ButtonHeight ...
          m_PanelWidth m_PanelHeight]);

% create a selection pushbutton
Tab5_H2 = uicontrol('Style', 'pushbutton','Units',
'pixels','BackgroundColor', Col_H2,'Position',
[TabOffset+4*m_ButtonWidth TabOffset...
```

```
          m_ButtonWidth m_ButtonHeight],'String',
Tab5_Lab,'HorizontalAlignment', 'center','FontName',
'arial','FontWeight', 'bold',...
          'FontSize', Font_medium);

 %%      Build the Tab_6 (Simulation)

% create a UIPanel
Tab6_H1 = uipanel('Units', 'pixels','Visible',
'off','Backgroundcolor',
Col_H1,'BorderWidth',1,'Position', [TabOffset
m_ButtonHeight ...
          m_PanelWidth m_PanelHeight]);

% create a selection pushbutton
Tab6_H2 = uicontrol('Style', 'pushbutton','Units',
'pixels','BackgroundColor', Col_H2,'Position',
[TabOffset+5*m_ButtonWidth TabOffset...
          m_ButtonWidth m_ButtonHeight],'String',
Tab6_Lab,'HorizontalAlignment', 'center','FontName',
'arial','FontWeight', 'bold',...
          'FontSize', Font_medium);

 %%      Build the Tab_7 (Report)

% create a UIPanel
Tab7_H1 = uipanel('Units', 'pixels','Visible',
'off','Backgroundcolor',
Col_H1,'BorderWidth',1,'Position', [TabOffset
m_ButtonHeight ...
          m_PanelWidth m_PanelHeight]);

% create a selection pushbutton
Tab7_H2 = uicontrol('Style', 'pushbutton','Units',
'pixels','BackgroundColor', Col_H2,'Position',
[TabOffset+6*m_ButtonWidth TabOffset...
          m_ButtonWidth m_ButtonHeight],'String',
Tab7_Lab,'HorizontalAlignment', 'center','FontName',
'arial','FontWeight', 'bold',...
          'FontSize', Font_medium);
 %%    Define the callbacks for the Tab Buttons

        set(Tab1_H1, 'Visible', 'on');
        set(Tab1_H2, 'Backgroundcolor', Col_H3);

        set (Tab1_H2, 'callback',{@TabSellectCallback,
Tab1_H2})
```

```
        set (Tab2_H2, 'callback',{@TabSellectCallback,
Tab2_H2})
        set (Tab3_H2, 'callback',{@TabSellectCallback,
Tab3_H2})
        set (Tab4_H2, 'callback',{@TabSellectCallback,
Tab4_H2})
        set (Tab5_H2, 'callback',{@TabSellectCallback,
Tab5_H2})
        set (Tab6_H2, 'callback',{@TabSellectCallback,
Tab6_H2})
        set (Tab7_H2, 'callback',{@TabSellectCallback,
Tab7_H2})

%%   Define content for the Project Information Tab
(Tab_1)

% create a UIPanel "Project Information"
uipanel('Parent', Tab1_H1, 'Units',
'pixels','Visible', 'on','Backgroundcolor',
Col_H1,'BorderWidth',1.5, 'Title', 'Project
Information',...
'FontSize', Font_title,'FontWeight', 'bold',
'Position', [0.03*m_PanelWidth 0.63*m_PanelHeight
0.95*m_PanelWidth 0.35*m_PanelHeight]);

 % create a Static Text "Project Title"

uicontrol('Parent', Tab1_H1,'Style',
'text','Position', [0.058*m_PanelWidth
0.84*m_PanelHeight 0.2*m_PanelWidth
0.065*m_PanelHeight],'string', 'Project Title',...
'HorizontalAlignment', 'left','FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create an edit Text box for "Project Title"
uicontrol('Parent', Tab1_H1,'Style',
'edit','Position', [0.19*m_PanelWidth
0.865*m_PanelHeight 0.72*m_PanelWidth
0.0456*m_PanelHeight],...
'HorizontalAlignment', 'left','Backgroundcolor',
Col_H3,'FontName', 'arial','FontWeight',
'normal','FontSize', Font_large);

% create a Static Text "Project Description"
```

```
uicontrol('Parent', Tab1_H1,'Style',
'text','Position', [0.058*m_PanelWidth
0.74*m_PanelHeight 0.2*m_PanelWidth
0.065*m_PanelHeight],'string',...
'Project Description','HorizontalAlignment',
'left','FontName', 'arial','FontWeight',
'normal','FontSize', Font_large);

 % create an edit Text box for "Project Description"
 Tab1_edit1_H = uicontrol('Parent', Tab1_H1,'Style',
'edit', 'Position', [0.19*m_PanelWidth
0.75*m_PanelHeight 0.72*m_PanelWidth
0.08463*m_PanelHeight],...
               'HorizontalAlignment',
'left','Backgroundcolor', Col_H3,'FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

set(Tab1_edit1_H, 'Min', 0.19*m_PanelWidth, 'Max',
0.72*m_PanelWidth);

% create a Static Text "Date"
uicontrol('Parent', Tab1_H1,'Style',
'text','Position', [0.14*m_PanelWidth
0.65*m_PanelHeight 0.14*m_PanelWidth
0.065*m_PanelHeight],'string', 'Date',...
'HorizontalAlignment', 'left','FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create an edit Text box for "Date"
 uicontrol('Parent', Tab1_H1,'Style',
'edit','Position', [0.19*m_PanelWidth
0.672*m_PanelHeight 0.2*m_PanelWidth
0.0453*m_PanelHeight],'HorizontalAlignment',...
 'left','Backgroundcolor', Col_H3,'FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create a UIPanel "Mathematical Models"
uipanel('Parent', Tab1_H1, 'Units',
'pixels','Visible', 'on','Backgroundcolor',
Col_H1,'BorderWidth',1.5, 'Title', 'Mathematical
Models','FontSize', Font_title,...
```

```
'FontWeight', 'bold', 'Position', [0.03*m_PanelWidth
0.50*m_PanelHeight 0.95*m_PanelWidth
0.12*m_PanelHeight]);

% create a pushbutton for Hollow Fibre Module
Tab1_SW = uicontrol('Parent', Tab1_H1,'Style',
'pushbutton','Units', 'pixels','BackgroundColor',
Col_H2,'Position', [0.15*m_PanelWidth
0.53*m_PanelHeight ...
0.16*m_PanelWidth 0.0455*m_PanelHeight],'String',
'Spiral Wound','HorizontalAlignment',
'center','FontName', 'arial','FontWeight',
'bold','FontSize', Font_medium);

% create a pushbutton for Hollow Fibre Module
Tab1_co_flow = uicontrol('Parent', Tab1_H1,'Style',
'pushbutton','Units', 'pixels','BackgroundColor',
Col_H2,'Position', [0.45*m_PanelWidth
0.53*m_PanelHeight ...
0.16*m_PanelWidth 0.0455*m_PanelHeight],'String',
'Hollow Fibre','HorizontalAlignment',
'center','FontName', 'arial','FontWeight',
'bold','FontSize', Font_medium);

set(Tab1_co_flow, 'callback',{@Co_Flow_Call1_Callback,
Tab1_co_flow})

% create a pushbutton for Counter-Current Flow
Tab1_counter_flow = uicontrol('Parent',
Tab1_H1,'Style', 'pushbutton','Units',
'pixels','BackgroundColor', Col_H2,'Position',
[0.75*m_PanelWidth 0.53*m_PanelHeight ...
0.16*m_PanelWidth 0.0455*m_PanelHeight],'String',
'Plate & Frame','HorizontalAlignment',
'center','FontName', 'arial','FontWeight',
'bold','FontSize', Font_medium);

set(Tab1_counter_flow,
'callback',{@Counter_Flow_Call1_Callback,
Tab1_counter_flow})

% create a UIPanel "Analysis Preferences"
 uipanel('Parent', Tab1_H1, 'Units',
'pixels','Visible', 'on', 'Backgroundcolor',
Col_H1,'BorderWidth',1.5, 'Title', 'Analysis
Preferences','FontSize', Font_title,...
```

```
'FontWeight', 'bold', 'Position', [0.03*m_PanelWidth
0.013*m_PanelHeight 0.45*m_PanelWidth
0.475*m_PanelHeight]);

% create a UIPanel "Units"
uipanel('Parent', Tab1_H1, 'Units',
'pixels','Visible', 'on', 'Backgroundcolor',
Col_H1,'BorderWidth',1.5, 'BorderType',
'beveledout','Title', 'Units','FontSize',
Font_title,...
'FontWeight', 'normal', 'Position',
[0.096*m_PanelWidth 0.0715*m_PanelHeight
0.31*m_PanelWidth 0.35*m_PanelHeight]);

% create a Static Text "Flowrate"
uicontrol('Parent', Tab1_H1,'Style', 'text',
'Position', [0.15*m_PanelWidth 0.29*m_PanelHeight
0.2*m_PanelWidth 0.065*m_PanelHeight],'string',
'Flowrate',...
'HorizontalAlignment', 'left','FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create a popup menue for "Flowrate"
uicontrol('Parent', Tab1_H1,'Style',
'popupmenu','Position', [0.25*m_PanelWidth
0.295*m_PanelHeight 0.07*m_PanelWidth
0.065*m_PanelHeight],'string',...
{'','lpm','L/Hr'},'HorizontalAlignment',
'left','Backgroundcolor', Col_H3,'FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create a Static Text "Concentration"
 uicontrol('Parent', Tab1_H1,'Style',
'text','Position', [0.15*m_PanelWidth
0.22*m_PanelHeight 0.2*m_PanelWidth
0.065*m_PanelHeight],'string',...
 'Concentration','HorizontalAlignment',
'left','FontName', 'arial','FontWeight',
'normal','FontSize', Font_large);

% create a popup menue for "Concentration"
uicontrol('Parent', Tab1_H1,'Style',
'popupmenu','Position', [0.25*m_PanelWidth
```

```
0.225*m_PanelHeight 0.07*m_PanelWidth
0.065*m_PanelHeight],'string',...
{'','mg/L','M'},'HorizontalAlignment',
'left','Backgroundcolor', Col_H3,'FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create a Static Text "Length"
uicontrol('Parent', Tab1_H1,'Style',
'text','Position', [0.15*m_PanelWidth
0.15*m_PanelHeight 0.2*m_PanelWidth
0.065*m_PanelHeight],'string', 'Length',...
'HorizontalAlignment', 'left','FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create a popup menue for "Length"
uicontrol('Parent', Tab1_H1,'Style',
'popupmenu','Position', [0.25*m_PanelWidth
0.155*m_PanelHeight 0.07*m_PanelWidth
0.065*m_PanelHeight],'string',...
{'','m','ft'},'HorizontalAlignment',
'left','Backgroundcolor', Col_H3,'FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create a Static Text "Temperature"
 uicontrol('Parent', Tab1_H1,'Style',
'text','Position', [0.15*m_PanelWidth
0.08*m_PanelHeight 0.2*m_PanelWidth
0.065*m_PanelHeight],'string',...
 'Temperature','HorizontalAlignment',
'left','FontName', 'arial','FontWeight',
'normal','FontSize', Font_large);

% create a popup menue for "Temperature"
 uicontrol('Parent', Tab1_H1,'Style',
'popupmenu','Position', [0.25*m_PanelWidth
0.085*m_PanelHeight 0.07*m_PanelWidth
0.065*m_PanelHeight],...
 'string', {'','C','F', 'K'},'HorizontalAlignment',
'left','Backgroundcolor', Col_H3,'FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create a UIPanel "Analyser Details"
```

```
 uipanel('Parent', Tab1_H1, 'Units',
'pixels','Visible', 'on', 'Backgroundcolor',
Col_H1,'BorderWidth',1.5, 'Title', 'Analyser
Details',...
 'FontSize', Font_title,'FontWeight', 'bold',
'Position', [0.5*m_PanelWidth 0.013*m_PanelHeight
0.48*m_PanelWidth 0.475*m_PanelHeight]);

 % create a Static Text "Analyser Name"
 uicontrol('Parent', Tab1_H1,'Style',
'text','Position', [0.52*m_PanelWidth
0.30*m_PanelHeight 0.2*m_PanelWidth
0.065*m_PanelHeight],'string', 'Analyser Name',...
 'HorizontalAlignment', 'left','FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create an edit Text box for "Analyser Name"
uicontrol('Parent', Tab1_H1,'Style',
'edit','Position', [0.63*m_PanelWidth
0.33*m_PanelHeight 0.32*m_PanelWidth
0.045*m_PanelHeight],'HorizontalAlignment',...
'left','Backgroundcolor', Col_H3,'FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create a Static Text "Designation"
uicontrol('Parent', Tab1_H1,'Style',
'text','Position', [0.52*m_PanelWidth
0.24*m_PanelHeight 0.2*m_PanelWidth
0.065*m_PanelHeight],'string', 'Designation',...
'HorizontalAlignment', 'left','FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create an edit Text box for "Designation"
uicontrol('Parent', Tab1_H1,'Style',
'edit','Position', [0.63*m_PanelWidth
0.265*m_PanelHeight 0.32*m_PanelWidth
0.045*m_PanelHeight],'HorizontalAlignment',...
'left','Backgroundcolor', Col_H3,'FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create a Static Text "Phone No"
```

```
uicontrol('Parent', Tab1_H1,'Style',
'text','Position', [0.52*m_PanelWidth
0.18*m_PanelHeight 0.2*m_PanelWidth
0.065*m_PanelHeight],'string', 'Phone No',...
'HorizontalAlignment', 'left','FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create an edit Text box for "Phone No"
uicontrol('Parent', Tab1_H1,'Style',
'edit','Position', [0.63*m_PanelWidth
0.2*m_PanelHeight 0.32*m_PanelWidth
0.045*m_PanelHeight],'HorizontalAlignment',...
'left','Backgroundcolor', Col_H3,'FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create a Static Text "E-mail"
 uicontrol('Parent', Tab1_H1,'Style',
'text','Position', [0.52*m_PanelWidth
0.12*m_PanelHeight 0.2*m_PanelWidth
0.065*m_PanelHeight],'string', 'E-mail',...
 'HorizontalAlignment', 'left','FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

 % create an edit Text box for "E-mail"
 uicontrol('Parent', Tab1_H1,'Style',
'edit','Position', [0.63*m_PanelWidth
0.135*m_PanelHeight 0.32*m_PanelWidth
0.045*m_PanelHeight],'HorizontalAlignment',...
 'left', 'Backgroundcolor', Col_H3,'FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create a Static Text "Company"
uicontrol('Parent', Tab1_H1,'Style',
'text','Position', [0.52*m_PanelWidth
0.06*m_PanelHeight 0.2*m_PanelWidth
0.065*m_PanelHeight],'string', 'Company',...
'HorizontalAlignment', 'left','FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create an edit Text box for "Company"
```

```
uicontrol('Parent', Tab1_H1,'Style',
'edit','Position', [0.63*m_PanelWidth
0.07*m_PanelHeight 0.32*m_PanelWidth
0.045*m_PanelHeight],...
'HorizontalAlignment', 'left', 'Backgroundcolor',
Col_H3,'FontName', 'arial','FontWeight',
'normal','FontSize', Font_large);

%%   Define content for the Feed Solution Analysis Tab
(Tab_2)

% create a Static Text for the title of the table
"Feed Solution Analysis"
uicontrol('Style', 'text','Position',[0.4*m_PanelWidth
0.9*m_PanelHeight 0.22*m_PanelWidth
0.065*m_PanelHeight ],'Parent', Tab2_H1,'string',...
'Feed Solution Analysis','BackgroundColor',
Col_H1,'HorizontalAlignment', 'center','FontName',
'arial','FontWeight', 'bold','FontSize',
Font_tab_title);

%   Build the data cell array to display
Tbl_FS_DisplayData = cell(14,5);
Tbl_FS_DisplayData{1,1} = 'Ammonium (NH4++NH3)';
Tbl_FS_DisplayData{2,1} = 'Potassium (K)';
Tbl_FS_DisplayData{3,1} = 'Sodium (Na)';
Tbl_FS_DisplayData{4,1} = 'Magnesium (Mg)';
Tbl_FS_DisplayData{5,1} = 'Calcium (Ca)';
Tbl_FS_DisplayData{6,1} = 'Stronium (Sr)';
Tbl_FS_DisplayData{7,1} = 'Barium (Ba)';
Tbl_FS_DisplayData{8,1} = 'Boron (B)';
Tbl_FS_DisplayData{9,1} = 'Carbonate (CO3)';
Tbl_FS_DisplayData{10,1} = 'Bicarbonate (HCO3)';
Tbl_FS_DisplayData{11,1} = 'Nitrate (NO3)';
Tbl_FS_DisplayData{12,1} = 'Chloride (Cl)';
Tbl_FS_DisplayData{13,1} = 'Fluoride (F)';
Tbl_FS_DisplayData{14,1} = 'Sulphate (SO4)';
Tbl_FS_DisplayData{15,1} = 'Silica (SiO2)';

Tbl_FS_ColumnNames = {' Ions ' ' mg/L ' ' meq/L ' '
Ionic Strength ' ' mg/L as CaCO3 '};
Tbl_FS_Width =
((m_PanelWidth/2)/5)+0.006*m_PanelWidth;
Tbl_FS_ColumnWidths = {1.3*Tbl_FS_Width
0.925*Tbl_FS_Width 0.925*Tbl_FS_Width
0.925*Tbl_FS_Width 0.925*Tbl_FS_Width};
```

```
Tbl_FS_foregroundColor = [0 0 0];
Tbl_FS_backgroundColor = [0 1 1; 0 1 0];

%   Create the table
Tbl_FS = uitable('Parent', Tab2_H1, 'ColumnName',
Tbl_FS_ColumnNames,'FontSize',Font_medium,'Position',[
0.02*m_PanelWidth 0.21*m_PanelHeight Tbl_FS_Width*5
0.52*m_PanelHeight],...
'ColumnWidth', Tbl_FS_ColumnWidths,'RowName',
[],'Data', Tbl_FS_DisplayData);

set(Tbl_FS, 'ForegroundColor',
Tbl_FS_foregroundColor);
set(Tbl_FS, 'BackgroundColor',
Tbl_FS_backgroundColor);

set(Tbl_FS,'ColumnEditable',logical([0 0 0 0 0]));

% create a Static Text "Prefiltration Type"
uicontrol('Parent', Tab2_H1,'Style',
'text','Position', [0.02*m_PanelWidth
0.83*m_PanelHeight 0.12*m_PanelWidth
0.065*m_PanelHeight],...
'string', 'Prefiltration Type','HorizontalAlignment',
'left','FontName', 'arial','FontWeight',
'normal','FontSize', Font_large);

% create a popup menue for "Prefiltration Type"
uicontrol('Parent', Tab2_H1,'Style',
'popupmenu','Position', [0.14*m_PanelWidth
0.825*m_PanelHeight 0.2*m_PanelWidth
0.078*m_PanelHeight],...
'string', {'','Cartidge filtration','Media
filtration', 'Microfiltration',
'Ultrafiltration'},'HorizontalAlignment',
'left','Backgroundcolor', Col_H3,'FontName',
'arial','FontWeight', 'normal',...
'FontSize', Font_large);

% create a Static Text "Feed Solution"
uicontrol('Parent', Tab2_H1,'Style',
'text','Position', [0.02*m_PanelWidth
0.75*m_PanelHeight 0.12*m_PanelWidth
0.065*m_PanelHeight],'string', 'Feed Solution',...
```

```
'HorizontalAlignment', 'left','FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create a static text to display the "Feed Solution"
Tab2_FS_disp = uicontrol('Parent', Tab2_H1,'Style',
'edit', 'Enable', 'off', 'Position',
[0.14*m_PanelWidth 0.78*m_PanelHeight
0.16*m_PanelWidth 0.04*m_PanelHeight],...
'HorizontalAlignment', 'center', 'Backgroundcolor',
Col_H3,'FontName', 'arial', 'FontWeight',
'normal','FontSize', Font_large);

% create a Static Text "SDI (Silt Density Index)"
uicontrol('Parent', Tab2_H1,'Style',
'text','Position', [0.33*m_PanelWidth
0.75*m_PanelHeight 0.18*m_PanelWidth
0.065*m_PanelHeight],...
'string', 'SDI (Silt Density
Index)','HorizontalAlignment', 'left','FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create a Static Text to display "SDI (Silt Density
Index)"
uicontrol('Parent', Tab2_H1,'Style', 'edit', 'Enable',
'off','Position', [0.48*m_PanelWidth
0.78*m_PanelHeight 0.06*m_PanelWidth
0.04*m_PanelHeight],...
'HorizontalAlignment', 'center', 'Backgroundcolor',
Col_H3,'FontName', 'arial','FontWeight',
'normal','FontSize', Font_medium);

% create a pushbutton for "Load Feed Solution from
Library"
Tab2_FS_Load = uicontrol('Parent', Tab2_H1,'Style',
'pushbutton','Units', 'pixels','BackgroundColor',
Col_H2,'Position', [0.6*m_PanelWidth
0.825*m_PanelHeight ...
0.3*m_PanelWidth 0.078*m_PanelHeight],'String', 'Load
Feed Solution from Library','HorizontalAlignment',
'center','FontName', 'arial','FontWeight', 'bold',...
'FontSize', Font_medium);

set(Tab2_FS_Load, 'callback',{@FS_Load_Callback,
Tab2_FS_Load});
```

```
% create a Check box for "Specify Feed Solution"
Tab2_chk_box_1 = uicontrol('Parent',
Tab2_H1,'Style','checkbox','String','Specify Feed
Solution', 'FontName', 'arial','FontWeight',
'normal','FontSize', Font_large, 'Value',0,...
'Position',[0.7*m_PanelWidth 0.75*m_PanelHeight
0.16*m_PanelWidth 0.04*m_PanelHeight]);

set (Tab2_chk_box_1, 'callback',{@SP_FS_Callback,
Tab2_chk_box_1})

% create a Static Text "Total Dissolved Solids"
uicontrol('Parent', Tab2_H1,'Style',
'text','Position', [0.6*m_PanelWidth
0.63*m_PanelHeight 0.3*m_PanelWidth
0.078*m_PanelHeight],'string', 'Total Dissolved
Solids',...
'HorizontalAlignment', 'left','FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create a Static Text to display the calculated
"Total Dissolved Solids"
Tab2_TDS_text = uicontrol('Parent', Tab2_H1,'Style',
'edit', 'Enable', 'off','Position', [0.75*m_PanelWidth
0.675*m_PanelHeight 0.0625*m_PanelWidth
0.045*m_PanelHeight],'string', '',...
'HorizontalAlignment', 'center','Backgroundcolor',
Col_H3,'FontName', 'arial','FontWeight',
'normal','FontSize', Font_large);

% create a Static Text  "mg/L"
uicontrol('Parent', Tab2_H1,'Style',
'text','Position', [0.82*m_PanelWidth
0.67*m_PanelHeight 0.05*m_PanelWidth
0.045*m_PanelHeight],'string', 'mg/L',...
'HorizontalAlignment', 'left','FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create a Static Text "Osmotic Pressure"
uicontrol('Parent', Tab2_H1,'Style',
'text','Position', [0.6*m_PanelWidth
0.57*m_PanelHeight 0.3*m_PanelWidth
0.078*m_PanelHeight],'string', 'Osmotic Pressure',...
```

```
'HorizontalAlignment', 'left','FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create a Static Text to display the calculated
"Osmotic Pressure"
Tab2_OP_FS = uicontrol('Parent', Tab2_H1,'Style',
'edit', 'Enable', 'off','Position', [0.75*m_PanelWidth
0.615*m_PanelHeight 0.0625*m_PanelWidth
0.04*m_PanelHeight],'string', '',...
'HorizontalAlignment', 'center','Backgroundcolor',
Col_H3,'FontName', 'arial','FontWeight',
'normal','FontSize', Font_large);

% create a Static Text  "bar"
uicontrol('Parent', Tab2_H1,'Style',
'text','Position', [0.82*m_PanelWidth
0.61*m_PanelHeight 0.06*m_PanelWidth
0.04*m_PanelHeight],'string', 'bar',...
'HorizontalAlignment', 'left','FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create a UIPanel "Feed Solution Parameters"
uipanel('Parent', Tab2_H1, 'Units',
'pixels','Visible', 'on', 'Backgroundcolor',
Col_H1,'BorderWidth',1.5, 'Title', 'Feed Solution
Parameters','FontSize', Font_title,...
'FontWeight', 'bold', 'Position', [0.58*m_PanelWidth
0.32*m_PanelHeight 0.38*m_PanelWidth
0.28*m_PanelHeight]);

% create a Static Text  "Temperature"
uicontrol('Parent', Tab2_H1,'Style',
'text','Position', [0.6*m_PanelWidth
0.49*m_PanelHeight 0.16*m_PanelWidth
0.04*m_PanelHeight],'string', ' Solution
Temperature',...
'HorizontalAlignment', 'left','FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create an edit Text box for "Temperature"
Tab2_edit_temp = uicontrol('Parent', Tab2_H1,'Style',
'edit','Position', [0.75*m_PanelWidth
```

```
0.495*m_PanelHeight 0.0625*m_PanelWidth
0.04*m_PanelHeight],...
'HorizontalAlignment', 'center', 'Backgroundcolor',
Col_H3,'FontName', 'arial','FontWeight',
'normal','FontSize', Font_large);

% create a Static Text for "C"
uicontrol('Parent', Tab2_H1,'Style',
'text','Position', [0.82*m_PanelWidth
0.49*m_PanelHeight 0.03*m_PanelWidth
0.04*m_PanelHeight],'string', ' C',...
'HorizontalAlignment', 'left','FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create a Static Text  "Solution pH"
uicontrol('Parent', Tab2_H1,'Style',
'text','Position', [0.6*m_PanelWidth
0.42*m_PanelHeight 0.16*m_PanelWidth
0.04*m_PanelHeight],'string', ' Solution pH',...
'HorizontalAlignment', 'left','FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create an edit Text box for "pH"
Tab2_edit_pH = uicontrol('Parent', Tab2_H1,'Style',
'edit', 'Enable', 'off','Position', [0.75*m_PanelWidth
0.425*m_PanelHeight 0.0625*m_PanelWidth
0.04*m_PanelHeight],...
'HorizontalAlignment', 'center', 'Backgroundcolor',
Col_H3,'FontName', 'arial','FontWeight',
'normal','FontSize', Font_medium);

% create a Static Text  "Solution pOH"
uicontrol('Parent', Tab2_H1,'Style',
'text','Position', [0.6*m_PanelWidth
0.35*m_PanelHeight 0.16*m_PanelWidth
0.04*m_PanelHeight],'string', ' Solution pOH',...
'HorizontalAlignment', 'left','FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create an edit Text box for "pOH"
Tab2_edit_pOH = uicontrol('Parent', Tab2_H1,'Style',
'edit', 'Enable', 'off','Position', [0.75*m_PanelWidth
```

```
0.355*m_PanelHeight 0.0625*m_PanelWidth
0.04*m_PanelHeight],...
'HorizontalAlignment', 'center', 'Backgroundcolor',
Col_H3,'FontName', 'arial','FontWeight',
'normal','FontSize', Font_medium);

% create a UIPanel "Charge Balance"
uipanel('Parent', Tab2_H1, 'Units',
'pixels','Visible', 'on', 'Backgroundcolor',
Col_H1,'BorderWidth',1.5, 'Title', 'Charge
Balance','FontSize', Font_title,...
'FontWeight', 'bold', 'Position', [0.58*m_PanelWidth
0.03*m_PanelHeight 0.38*m_PanelWidth
0.29*m_PanelHeight]);

% create a Static Text "Cations"
uicontrol('Parent', Tab2_H1,'Style',
'text','Position', [0.6*m_PanelWidth
0.21*m_PanelHeight 0.16*m_PanelWidth
0.04*m_PanelHeight],'string', 'Cations',...
'HorizontalAlignment', 'left','FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create a Static Text to display the calculated
"Cations"
Tab2_Cations_text = uicontrol('Parent',
Tab2_H1,'Style', 'edit', 'Enable', 'off','Position',
[0.66*m_PanelWidth 0.21*m_PanelHeight
0.0625*m_PanelWidth 0.04*m_PanelHeight],'string',
'',...
'HorizontalAlignment', 'center','Backgroundcolor',
Col_H3,'FontName', 'arial','FontWeight',
'normal','FontSize', Font_large);

% create a Static Text "Anions"
uicontrol('Parent', Tab2_H1,'Style',
'text','Position', [0.6*m_PanelWidth
0.14*m_PanelHeight 0.16*m_PanelWidth
0.04*m_PanelHeight],'string', 'Anions',...
'HorizontalAlignment', 'left','FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create a Static Text to display the calculated
"Anions"
```

```
Tab2_Anions_text = uicontrol('Parent',
Tab2_H1,'Style', 'edit', 'Enable', 'off','Position',
[0.66*m_PanelWidth 0.145*m_PanelHeight
0.0625*m_PanelWidth 0.04*m_PanelHeight],'string',
'',...
'HorizontalAlignment', 'center','Backgroundcolor',
Col_H3,'FontName', 'arial','FontWeight',
'normal','FontSize', Font_large);

% create a Static Text "Balance"
uicontrol('Parent', Tab2_H1,'Style',
'text','Position', [0.6*m_PanelWidth
0.07*m_PanelHeight 0.16*m_PanelWidth
0.04*m_PanelHeight],'string', 'Balance',...
'HorizontalAlignment', 'left','FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create a Static Text to display the calculated
"Balance"
Tab2_Balance_text = uicontrol('Parent',
Tab2_H1,'Style', 'edit', 'Enable', 'off','Position',
[0.66*m_PanelWidth 0.075*m_PanelHeight
0.0625*m_PanelWidth 0.04*m_PanelHeight],'string',
'',...
'HorizontalAlignment', 'center','Backgroundcolor',
Col_H3,'FontName', 'arial','FontWeight',
'normal','FontSize', Font_large);

% create a pushbutton for "Adjust Cations"
uicontrol('Parent', Tab2_H1,'Style',
'pushbutton','Units', 'pixels','BackgroundColor',
Col_H2,'Position', [0.74*m_PanelWidth
0.21*m_PanelHeight...
0.1*m_PanelWidth 0.04*m_PanelHeight],'String', 'Adjust
Cations','HorizontalAlignment', 'center','FontName',
'arial','FontWeight', 'bold','FontSize', Font_medium);

% create a pushbutton for "Adjust Anions"
uicontrol('Parent', Tab2_H1,'Style',
'pushbutton','Units', 'pixels','BackgroundColor',
Col_H2,'Position', [0.74*m_PanelWidth
0.145*m_PanelHeight...
0.1*m_PanelWidth 0.04*m_PanelHeight],'String', 'Adjust
Anions','HorizontalAlignment', 'center','FontName',
'arial','FontWeight', 'bold','FontSize', Font_medium);
```

```
% create a pushbutton for "Adjust All Ions"
uicontrol('Parent', Tab2_H1,'Style',
'pushbutton','Units', 'pixels','BackgroundColor',
Col_H2,'Position', [0.74*m_PanelWidth
0.075*m_PanelHeight...
0.1*m_PanelWidth 0.04*m_PanelHeight],'String', 'Adjust
All Ions','HorizontalAlignment', 'center','FontName',
'arial','FontWeight', 'bold','FontSize', Font_medium);

% create a pushbutton for "Add Sodium"
uicontrol('Parent', Tab2_H1,'Style',
'pushbutton','Units', 'pixels','BackgroundColor',
Col_H2,'Position', [0.86*m_PanelWidth
0.21*m_PanelHeight...
0.09*m_PanelWidth 0.04*m_PanelHeight],'String', 'Add
Sodium','HorizontalAlignment', 'center','FontName',
'arial','FontWeight', 'bold','FontSize', Font_medium);

% create a pushbutton for "Add Calcium"
uicontrol('Parent', Tab2_H1,'Style',
'pushbutton','Units', 'pixels','BackgroundColor',
Col_H2,'Position', [0.86*m_PanelWidth
0.145*m_PanelHeight...
0.09*m_PanelWidth 0.04*m_PanelHeight],'String', 'Add
Calcium','HorizontalAlignment', 'center','FontName',
'arial','FontWeight', 'bold','FontSize', Font_medium);

% create a pushbutton for "Save Feed Solution Profile
to Library"
Tab2_FS_Save = uicontrol('Parent', Tab2_H1,'Style',
'pushbutton','Units', 'pixels','BackgroundColor',
Col_H2,'Position', [0.02*m_PanelWidth
0.1*m_PanelHeight Tbl_FS_Width*5
0.08*m_PanelHeight],...
'String', 'Save Feed Solution Profile to
Library','HorizontalAlignment', 'center','FontName',
'arial','FontWeight', 'bold','FontSize', Font_medium);

set(Tab2_FS_Save, 'callback',{@FS_Save_Callback,
Tab2_FS_Save});

% create a pushbutton for "Calculate"
Tab2_cal1 = uicontrol('Parent', Tab2_H1,'Style',
'pushbutton','Units', 'pixels','BackgroundColor',
Col_H2,'Position', [1.3*Tbl_FS_Width
```

```
0.02*m_PanelHeight 3*0.925*Tbl_FS_Width
0.06*m_PanelHeight],...
'String', 'Calculate','HorizontalAlignment',
'center','FontName', 'arial','FontWeight',
'bold','FontSize', Font_medium);

set(Tab2_cal1, 'callback',{@FS_Call1_Callback,
Tab2_cal1})

%%   Define content for the Draw Solution Analysis Tab
(Tab_3)

% create a Static Text for the title of the table
"Draw Solution Analysis"
uicontrol('Style', 'text','Position',[0.4*m_PanelWidth
0.9*m_PanelHeight 0.25*m_PanelWidth
0.065*m_PanelHeight ],'Parent', Tab3_H1,'string',...
'Draw Solution Analysis','BackgroundColor',
Col_H1,'HorizontalAlignment', 'center','FontName',
'arial','FontWeight', 'bold','FontSize',
Font_tab_title);

%   Build the data cell array to display
Tbl_DS_DisplayData = cell(14,5);

Tbl_DS_DisplayData{1,1} = 'Ammonium (NH4++NH3)';
Tbl_DS_DisplayData{2,1} = 'Potassium (K)';
Tbl_DS_DisplayData{3,1} = 'Sodium (Na)';
Tbl_DS_DisplayData{4,1} = 'Magnesium (Mg)';
Tbl_DS_DisplayData{5,1} = 'Calcium (Ca)';
Tbl_DS_DisplayData{6,1} = 'Stronium (Sr)';
Tbl_DS_DisplayData{7,1} = 'Barium (Ba)';
Tbl_DS_DisplayData{8,1} = 'Boron (B)';
Tbl_DS_DisplayData{9,1} = 'Carbonate (CO3)';
Tbl_DS_DisplayData{10,1} = 'Bicarbonate (HCO3)';
Tbl_DS_DisplayData{11,1} = 'Nitrate (NO3)';
Tbl_DS_DisplayData{12,1} = 'Chloride (Cl)';
Tbl_DS_DisplayData{13,1} = 'Fluoride (F)';
Tbl_DS_DisplayData{14,1} = 'Sulphate (SO4)';
Tbl_DS_DisplayData{15,1} = 'Silica (SiO2)';

Tbl_DS_ColumnNames = {' Ions ' ' mg/L ' ' meq/L ' '
Ionic Strength ' ' mg/L as CaCO3 '};
Tbl_DS_Width =
((m_PanelWidth/2)/5)+0.006*m_PanelWidth;
```

```
Tbl_DS_ColumnWidths = {1.3*Tbl_DS_Width
0.925*Tbl_DS_Width 0.925*Tbl_DS_Width
0.925*Tbl_DS_Width 0.925*Tbl_DS_Width};

%   Create the table
Tbl_DS = uitable('Parent', Tab3_H1, 'ColumnName',
Tbl_DS_ColumnNames,'FontSize',Font_medium,'Position',[
0.02*m_PanelWidth 0.21*m_PanelHeight Tbl_DS_Width*5
0.52*m_PanelHeight],...
'ColumnWidth', Tbl_DS_ColumnWidths,'RowName',
[],'Data', Tbl_DS_DisplayData);

Tbl_DS_foregroundColor = [0 0 0];
Tbl_DS_backgroundColor = [0 1 1; 0 1 0];

set(Tbl_DS, 'ForegroundColor',
Tbl_DS_foregroundColor);
set(Tbl_DS, 'BackgroundColor',
Tbl_DS_backgroundColor);
set(Tbl_DS,'ColumnEditable',logical([0 0 0 0 0]));

% create a Static Text "Draw Solution"
uicontrol('Parent', Tab3_H1,'Style',
'text','Position', [0.02*m_PanelWidth
0.75*m_PanelHeight 0.12*m_PanelWidth
0.065*m_PanelHeight],'string', 'Draw Solution',...
'HorizontalAlignment', 'left','FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create an edit text to display the "Draw Solution"
Tab3_DS_disp = uicontrol('Parent', Tab3_H1,'Style',
'edit','Enable', 'off','Position', [0.14*m_PanelWidth
0.78*m_PanelHeight 0.12*m_PanelWidth
0.04*m_PanelHeight],...
'HorizontalAlignment', 'left', 'Backgroundcolor',
Col_H3,'FontName', 'arial', 'FontWeight',
'normal','FontSize', Font_medium);

% create a pushbutton for "Load Draw Solution from
Library"
Tab3_DS_Load = uicontrol('Parent', Tab3_H1,'Style',
'pushbutton','Units', 'pixels','BackgroundColor',
Col_H2,'Position', [0.6*m_PanelWidth
0.825*m_PanelHeight ...
```

```
0.3*m_PanelWidth 0.078*m_PanelHeight],'String', 'Load
Draw Solution from Library','HorizontalAlignment',
'center','FontName', 'arial','FontWeight', 'bold',...
'FontSize', Font_medium);

set(Tab3_DS_Load, 'callback',{@DS_Load_Callback,
Tab3_DS_Load});

% create a Check box for "Specify Draw Solution"
Tab3_chk_box_1 = uicontrol('Parent',
Tab3_H1,'Style','checkbox','String','Specify Draw
Solution', 'FontName', 'arial','FontWeight',
'normal','FontSize', Font_large, 'Value',0,...
'Position',[0.7*m_PanelWidth 0.75*m_PanelHeight
0.16*m_PanelWidth 0.04*m_PanelHeight]);

set (Tab3_chk_box_1, 'callback',{@SP_DS_Callback,
Tab3_chk_box_1});

% create a Static Text "Total Dissolved Solids"
uicontrol('Parent', Tab3_H1,'Style',
'text','Position', [0.6*m_PanelWidth
0.63*m_PanelHeight 0.3*m_PanelWidth
0.078*m_PanelHeight],'string', 'Total Dissolved
Solids',...
'HorizontalAlignment', 'left','FontName',
'arial','FontWeight', 'normal','FontSize',
Font_medium);

% create an edit Text to display the calculated "Total
Dissolved Solids"
Tab3_TDS_text = uicontrol('Parent', Tab3_H1,'Style',
'edit','Enable', 'off','Position', [0.75*m_PanelWidth
0.675*m_PanelHeight 0.0625*m_PanelWidth
0.045*m_PanelHeight],'string', '',...
'HorizontalAlignment', 'center','Backgroundcolor',
Col_H3,'FontName', 'arial','FontWeight',
'normal','FontSize', Font_large);

% create a Static Text  "mg/L"
uicontrol('Parent', Tab3_H1,'Style',
'text','Position', [0.82*m_PanelWidth
0.67*m_PanelHeight 0.05*m_PanelWidth
0.045*m_PanelHeight],'string', 'mg/L',...
```

```
'HorizontalAlignment', 'left','FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create a Static Text "Osmotic Pressure"
uicontrol('Parent', Tab3_H1,'Style',
'text','Position', [0.6*m_PanelWidth
0.57*m_PanelHeight 0.3*m_PanelWidth
0.078*m_PanelHeight],'string', 'Osmotic Pressure',...
'HorizontalAlignment', 'left','FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create an edit Text to display the calculated
"Osmotic Pressure"
Tab3_OP_DS = uicontrol('Parent', Tab3_H1,'Style',
'edit','Enable', 'off','Position', [0.75*m_PanelWidth
0.615*m_PanelHeight 0.0625*m_PanelWidth
0.04*m_PanelHeight],'string', '',...
'HorizontalAlignment', 'center','Backgroundcolor',
Col_H3,'FontName', 'arial','FontWeight',
'normal','FontSize', Font_large);

% create a Static Text  "bar"
uicontrol('Parent', Tab3_H1,'Style',
'text','Position', [0.82*m_PanelWidth
0.61*m_PanelHeight 0.06*m_PanelWidth
0.04*m_PanelHeight],'string', 'bar',...
'HorizontalAlignment', 'left','FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create a UIPanel "Draw Solution Parameters"
uipanel('Parent', Tab3_H1, 'Units',
'pixels','Visible', 'on', 'Backgroundcolor',
Col_H1,'BorderWidth',1.5, 'Title', 'Draw Solution
Parameters','FontSize', Font_title,...
'FontWeight', 'bold', 'Position', [0.58*m_PanelWidth
0.32*m_PanelHeight 0.38*m_PanelWidth
0.28*m_PanelHeight]);

% create a Static Text  "Temperature"
uicontrol('Parent', Tab3_H1,'Style',
'text','Position', [0.6*m_PanelWidth
0.49*m_PanelHeight 0.16*m_PanelWidth
```

```
0.04*m_PanelHeight],'string', ' Solution
Temperature',...
'HorizontalAlignment', 'left','FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create an edit Text box for "Temperature"
Tab3_edit_temp = uicontrol('Parent', Tab3_H1,'Style',
'edit','Position', [0.75*m_PanelWidth
0.495*m_PanelHeight 0.0625*m_PanelWidth
0.04*m_PanelHeight],...
'HorizontalAlignment', 'center', 'Backgroundcolor',
Col_H3,'FontName', 'arial','FontWeight',
'normal','FontSize', Font_medium);

% create a Static Text for "C"
uicontrol('Parent', Tab3_H1,'Style',
'text','Position', [0.82*m_PanelWidth
0.49*m_PanelHeight 0.03*m_PanelWidth
0.04*m_PanelHeight],'string', ' C',...
'HorizontalAlignment', 'left','FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create a Static Text  "Solution pH"
uicontrol('Parent', Tab3_H1,'Style',
'text','Position', [0.6*m_PanelWidth
0.42*m_PanelHeight 0.16*m_PanelWidth
0.04*m_PanelHeight],'string', ' Solution pH',...
'HorizontalAlignment', 'left','FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create an edit Text box for "pH"
Tab3_edit_pH = uicontrol('Parent', Tab3_H1,'Style',
'edit','Enable', 'off','Position', [0.75*m_PanelWidth
0.425*m_PanelHeight 0.0625*m_PanelWidth
0.04*m_PanelHeight],...
'HorizontalAlignment', 'center', 'Backgroundcolor',
Col_H3,'FontName', 'arial','FontWeight',
'normal','FontSize', Font_medium);

% create a Static Text  "Solution pOH"
uicontrol('Parent', Tab3_H1,'Style',
'text','Position', [0.6*m_PanelWidth
```

```
0.35*m_PanelHeight 0.16*m_PanelWidth
0.04*m_PanelHeight],'string', ' Solution pOH',...
'HorizontalAlignment', 'left','FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create an edit Text box for "pOH"
Tab3_edit_pOH = uicontrol('Parent', Tab3_H1,'Style',
'edit','Enable', 'off','Position', [0.75*m_PanelWidth
0.355*m_PanelHeight 0.0625*m_PanelWidth
0.04*m_PanelHeight],...
'HorizontalAlignment', 'center', 'Backgroundcolor',
Col_H3,'FontName', 'arial','FontWeight',
'normal','FontSize', Font_medium);

% create a UIPanel "Charge Balance"
uipanel('Parent', Tab3_H1, 'Units',
'pixels','Visible', 'on', 'Backgroundcolor',
Col_H1,'BorderWidth',1.5, 'Title', 'Charge
Balance','FontSize', Font_title,...
'FontWeight', 'bold', 'Position', [0.58*m_PanelWidth
0.03*m_PanelHeight 0.38*m_PanelWidth
0.29*m_PanelHeight]);

% create a Static Text "Cations"
uicontrol('Parent', Tab3_H1,'Style',
'text','Position', [0.6*m_PanelWidth
0.21*m_PanelHeight 0.16*m_PanelWidth
0.04*m_PanelHeight],'string', 'Cations',...
'HorizontalAlignment', 'left','FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create a Static Text to display the calculated
"Cations"
Tab3_Cations_text = uicontrol('Parent',
Tab3_H1,'Style', 'edit','Enable', 'off','Position',
[0.66*m_PanelWidth 0.21*m_PanelHeight
0.0625*m_PanelWidth 0.04*m_PanelHeight],'string',
'',...
'HorizontalAlignment', 'center','Backgroundcolor',
Col_H3,'FontName', 'arial','FontWeight',
'normal','FontSize', Font_large);

% create a Static Text "Anions"
```

```
uicontrol('Parent', Tab3_H1,'Style',
'text','Position', [0.6*m_PanelWidth
0.14*m_PanelHeight 0.16*m_PanelWidth
0.04*m_PanelHeight],'string', 'Anions',...
'HorizontalAlignment', 'left','FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create a Static Text to display the calculated
"Anions"
Tab3_Anions_text = uicontrol('Parent',
Tab3_H1,'Style', 'edit','Enable', 'off','Position',
[0.66*m_PanelWidth 0.145*m_PanelHeight
0.0625*m_PanelWidth 0.04*m_PanelHeight],'string',
'',...
'HorizontalAlignment', 'center','Backgroundcolor',
Col_H3,'FontName', 'arial','FontWeight',
'normal','FontSize', Font_large);

% create a Static Text "Balance"
uicontrol('Parent', Tab3_H1,'Style',
'text','Position', [0.6*m_PanelWidth
0.07*m_PanelHeight 0.16*m_PanelWidth
0.04*m_PanelHeight],'string', 'Balance',...
'HorizontalAlignment', 'left','FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create a Static Text to display the calculated
"Balance"
Tab3_Balance_text = uicontrol('Parent',
Tab3_H1,'Style', 'edit','Enable', 'off','Position',
[0.66*m_PanelWidth 0.075*m_PanelHeight
0.0625*m_PanelWidth 0.04*m_PanelHeight],'string',
'',...
'HorizontalAlignment', 'center','Backgroundcolor',
Col_H3,'FontName', 'arial','FontWeight',
'normal','FontSize', Font_large);

% create a pushbutton for "Adjust Cations"
uicontrol('Parent', Tab3_H1,'Style',
'pushbutton','Units', 'pixels','BackgroundColor',
Col_H2,'Position', [0.74*m_PanelWidth
0.21*m_PanelHeight...
```

```
0.1*m_PanelWidth 0.04*m_PanelHeight],'String', 'Adjust
Cations','HorizontalAlignment', 'center','FontName',
'arial','FontWeight', 'bold','FontSize', Font_medium);

% create a pushbutton for "Adjust Anions"
uicontrol('Parent', Tab3_H1,'Style',
'pushbutton','Units', 'pixels','BackgroundColor',
Col_H2,'Position', [0.74*m_PanelWidth
0.145*m_PanelHeight...
0.1*m_PanelWidth 0.04*m_PanelHeight],'String', 'Adjust
Anions','HorizontalAlignment', 'center','FontName',
'arial','FontWeight', 'bold','FontSize', Font_medium);

% create a pushbutton for "Adjust All Ions"
uicontrol('Parent', Tab3_H1,'Style',
'pushbutton','Units', 'pixels','BackgroundColor',
Col_H2,'Position', [0.74*m_PanelWidth
0.075*m_PanelHeight...
0.1*m_PanelWidth 0.04*m_PanelHeight],'String', 'Adjust
All Ions','HorizontalAlignment', 'center','FontName',
'arial','FontWeight', 'bold','FontSize', Font_medium);

% create a pushbutton for "Add Sodium"
uicontrol('Parent', Tab3_H1,'Style',
'pushbutton','Units', 'pixels','BackgroundColor',
Col_H2,'Position', [0.86*m_PanelWidth
0.21*m_PanelHeight...
0.09*m_PanelWidth 0.04*m_PanelHeight],'String', 'Add
Sodium','HorizontalAlignment', 'center','FontName',
'arial','FontWeight', 'bold','FontSize', Font_medium);

% create a pushbutton for "Add Calcium"
uicontrol('Parent', Tab3_H1,'Style',
'pushbutton','Units', 'pixels','BackgroundColor',
Col_H2,'Position', [0.86*m_PanelWidth
0.145*m_PanelHeight...
0.09*m_PanelWidth 0.04*m_PanelHeight],'String', 'Add
Calcium','HorizontalAlignment', 'center','FontName',
'arial','FontWeight', 'bold','FontSize', Font_medium);

% create a pushbutton for "Save Draw Solution Profile
to Library"
Tab3_DS_Save = uicontrol('Parent', Tab3_H1,'Style',
'pushbutton','Units', 'pixels','BackgroundColor',
Col_H2,'Position', [0.02*m_PanelWidth
```

```
0.1*m_PanelHeight Tbl_FS_Width*5
0.08*m_PanelHeight],...
'String', 'Save Draw Solution Profile to
Library','HorizontalAlignment', 'center','FontName',
'arial','FontWeight', 'bold','FontSize', Font_medium);

set(Tab3_DS_Save, 'callback',{@DS_Save_Callback,
Tab3_DS_Save});

% create a pushbutton for "Calculate"
Tab3_cal1 = uicontrol('Parent', Tab3_H1,'Style',
'pushbutton','Units', 'pixels','BackgroundColor',
Col_H2,'Position', [1.3*Tbl_FS_Width
0.02*m_PanelHeight 3*0.925*Tbl_FS_Width
0.06*m_PanelHeight],...
'String', 'Calculate','HorizontalAlignment',
'center','FontName', 'arial','FontWeight',
'bold','FontSize', Font_medium);

set(Tab3_cal1, 'callback',{@DS_Cal1_Callback,
Tab3_cal1})

%% Define Content for Tab4 (FO Optimization)

% create a Static Text for the title of the tab
"Parameter Optimization"
uicontrol('Style', 'text','Position',[0.4*m_PanelWidth
0.9*m_PanelHeight 0.25*m_PanelWidth
0.065*m_PanelHeight ],'Parent', Tab4_H1,'string',...
'Parameter Optimization','BackgroundColor',
Col_H1,'HorizontalAlignment', 'center','FontName',
'arial','FontWeight', 'bold','FontSize',
Font_tab_title);

% create a UIPanel "Design Objective"
Tab4_Pan_Des_Obj = uipanel('Parent', Tab4_H1, 'Units',
'pixels','Visible', 'on','Backgroundcolor',
Col_H1,'BorderWidth',1.5, 'Title', 'Design
Objectives',...
'FontSize', Font_large,'FontWeight', 'bold',
'Position', [0.015*m_PanelWidth 0.715*m_PanelHeight
0.45*m_PanelWidth 0.19*m_PanelHeight]);

% create a Static Text "Recovery(%)"
uicontrol('Parent', Tab4_Pan_Des_Obj,'Style',
'text','Position', [0.02*m_PanelWidth
```

```
0.085*m_PanelHeight 0.08*m_PanelWidth
0.04*m_PanelHeight],...
'string', 'Recovery(%)','HorizontalAlignment',
'left','FontName', 'arial','FontWeight',
'normal','FontSize', Font_large);

% create an edit Text box for "Recovery(%)"
Tab4_edit_Rec = uicontrol('Parent',
Tab4_Pan_Des_Obj,'Style', 'edit','Position',
[0.21*m_PanelWidth 0.09*m_PanelHeight
0.04*m_PanelWidth 0.043*m_PanelHeight],...
'HorizontalAlignment', 'center', 'Backgroundcolor',
Col_H3,'FontName', 'arial', 'FontWeight',
'normal','FontSize', Font_large);

% create a Static Text "Flexibility(%)"
uicontrol('Parent', Tab4_Pan_Des_Obj,'Style',
'text','Position', [0.29*m_PanelWidth
0.085*m_PanelHeight 0.08*m_PanelWidth
0.04*m_PanelHeight],...
'string', 'Flexibility(%)','HorizontalAlignment',
'left','FontName', 'arial','FontWeight',
'normal','FontSize', Font_large);

% create an edit Text box for "Flexibility(%)"
Tab4_edit_Rec_Flex = uicontrol('Parent',
Tab4_Pan_Des_Obj,'Style', 'edit','Position',
[0.39*m_PanelWidth 0.09*m_PanelHeight
0.04*m_PanelWidth 0.043*m_PanelHeight],...
'HorizontalAlignment', 'center', 'Backgroundcolor',
Col_H3,'FontName', 'arial', 'FontWeight',
'normal','FontSize', Font_large);

% create a Static Text "Final DS Concentration (M)"
uicontrol('Parent', Tab4_Pan_Des_Obj,'Style',
'text','Position', [0.02*m_PanelWidth
0.015*m_PanelHeight 0.17*m_PanelWidth
0.04*m_PanelHeight],...
'string', 'Final DS Concentration
(M)','HorizontalAlignment', 'left','FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create an edit Text box for "Final DS Concentration
(M)"
```

```
Tab4_edit_Conc = uicontrol('Parent',
Tab4_Pan_Des_Obj,'Style', 'edit','Position',
[0.21*m_PanelWidth 0.02*m_PanelHeight
0.04*m_PanelWidth 0.043*m_PanelHeight],...
'HorizontalAlignment', 'center', 'Backgroundcolor',
Col_H3,'FontName', 'arial', 'FontWeight',
'normal','FontSize', Font_large);

% create a Static Text "Flexibility(%)"
uicontrol('Parent', Tab4_Pan_Des_Obj,'Style',
'text','Position', [0.29*m_PanelWidth
0.015*m_PanelHeight 0.08*m_PanelWidth
0.04*m_PanelHeight],...
'string', 'Flexibility(%)','HorizontalAlignment',
'left','FontName', 'arial','FontWeight',
'normal','FontSize', Font_large);

% create an edit Text box for "Flexibility(%)"
Tab4_edit_Conc_Flex = uicontrol('Parent',
Tab4_Pan_Des_Obj,'Style', 'edit','Position',
[0.39*m_PanelWidth 0.02*m_PanelHeight
0.04*m_PanelWidth 0.043*m_PanelHeight],...
'HorizontalAlignment', 'center', 'Backgroundcolor',
Col_H3,'FontName', 'arial', 'FontWeight',
'normal','FontSize', Font_large);

% create a UIPanel for "Optimization Progress"
Tab_4_Panel_Status = uipanel('Parent', Tab4_H1,
'Units', 'pixels','Visible', 'on','Backgroundcolor',
Col_H1,'BorderWidth',1.5, 'Title', 'Optimization
Progress',...
'FontSize', Font_large,'FontWeight', 'bold',
'Position', [0.473*m_PanelWidth 0.775*m_PanelHeight
0.52*m_PanelWidth 0.13*m_PanelHeight]);

% create a Static Text "Status"
uicontrol('Parent', Tab_4_Panel_Status,'Style',
'text','Position', [0.015*m_PanelWidth
0.03*m_PanelHeight 0.08*m_PanelWidth
0.04*m_PanelHeight],'string', 'Status',...
'HorizontalAlignment', 'left','FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create a Static Text to display the Satus
Information
```

```
Tab4_status_disp = uicontrol('Parent',
Tab_4_Panel_Status,'Style', 'edi', 'Enable',
'off','Position', [0.07*m_PanelWidth
0.027*m_PanelHeight 0.15*m_PanelWidth
0.05*m_PanelHeight],'string', '',...
'HorizontalAlignment', 'center','Backgroundcolor',
Col_H3,'FontName', 'arial','FontWeight',
'normal','FontSize', Font_large);

% create a Static Text "Progress"
uicontrol('Parent', Tab_4_Panel_Status,'Style',
'text','Position', [0.27*m_PanelWidth
0.03*m_PanelHeight 0.08*m_PanelWidth
0.04*m_PanelHeight],'string', 'Progress',...
'HorizontalAlignment', 'left','FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

Tab4_ax_status = axes('Parent',
Tab_4_Panel_Status,'Units', 'pixels','Position',
[0.35*m_PanelWidth 0.027*m_PanelHeight
0.14*m_PanelWidth 0.05*m_PanelHeight],'XLim',[0
1],'YLim',[0 1],'Box', 'on', 'visible', 'off');
rectangle('Position', [0 0 1 1],
'FaceColor',Col_H3,'EdgeColor','k','LineWidth',0.25);
drawnow

% create a Static Text "Flow Configuration"
uicontrol('Parent', Tab4_H1,'Style',
'text','Position', [0.583*m_PanelWidth
0.7*m_PanelHeight 0.12*m_PanelWidth
0.04*m_PanelHeight],'string', 'Flow Configuration',...
'HorizontalAlignment', 'left','FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create a popup menue for "Flow Configuration"
Tab4_popup_FlCon = uicontrol('Parent',
Tab4_H1,'Style', 'popupmenu','Position',
[0.723*m_PanelWidth 0.7*m_PanelHeight
0.12*m_PanelWidth 0.04*m_PanelHeight],'string',...
{'','Co-Current Flow','Counter Current
Flow'},'HorizontalAlignment',
'left','Backgroundcolor', Col_H3,'FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);
```

```
set(Tab4_popup_FlCon,
'callback',{@Flow_Config_Callback, Tab4_popup_FlCon})

% create a UIPanel "Element Specification"
Tab4_Pan_El_spec = uipanel('Parent', Tab4_H1, 'Units',
'pixels','Visible', 'on','Backgroundcolor',
Col_H1,'BorderWidth',1.5, 'Title', 'Element
Specification',...
'FontSize', Font_large,'FontWeight', 'bold',
'Position', [0.015*m_PanelWidth 0.48*m_PanelHeight
0.45*m_PanelWidth 0.225*m_PanelHeight]);

% create a Static Text "Element"
uicontrol('Parent', Tab4_Pan_El_spec,'Style',
'text','Position', [0.015*m_PanelWidth
0.11*m_PanelHeight 0.1*m_PanelWidth
0.04*m_PanelHeight],'string', 'Element',...
'HorizontalAlignment', 'left','FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create a popup menue for "Element Model"
Tab4_popup_ElModel = uicontrol('Parent',
Tab4_Pan_El_spec,'Style', 'popupmenu','Position',
[0.11*m_PanelWidth 0.11*m_PanelHeight
0.12*m_PanelWidth 0.04*m_PanelHeight],'string',...
{'','TFC 8040','TFC 4040', 'CTA 8040', 'HPC3205',
'PFO-100'},'HorizontalAlignment',
'left','Backgroundcolor', Col_H3,'FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

set(Tab4_popup_ElModel,
'callback',{@El_Model_Callback, Tab4_popup_ElModel})

% create a Static Text "Membrane Type"
 uicontrol('Parent', Tab4_Pan_El_spec,'Style',
'text','Position', [0.25*m_PanelWidth
0.11*m_PanelHeight 0.08*m_PanelWidth
0.04*m_PanelHeight],'string', 'Membrane',...
'HorizontalAlignment', 'left','FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create a Static Text to display the "Membrane Type"
```

```
Tab4_edit_MemType = uicontrol('Parent',
Tab4_Pan_El_spec,'Style', 'edit', 'Enable',
'off','Position', [0.32*m_PanelWidth
0.11*m_PanelHeight 0.12*m_PanelWidth
0.05*m_PanelHeight],'string', '',...
'Backgroundcolor', Col_H3,'HorizontalAlignment',
'left','FontName', 'arial','FontWeight',
'normal','FontSize', Font_large);

% create a Static Text "Module Type"
uicontrol('Parent', Tab4_Pan_El_spec,'Style',
'text','Position', [0.014*m_PanelWidth
0.03*m_PanelHeight 0.1*m_PanelWidth
0.04*m_PanelHeight],'string', 'Module',...
'HorizontalAlignment', 'left','FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create an edit Text to display the "Module Type"
Tab4_edit_ModType = uicontrol('Parent',
Tab4_Pan_El_spec,'Style', 'edit','Enable',
'off','Position', [0.11*m_PanelWidth
0.03*m_PanelHeight 0.12*m_PanelWidth
0.05*m_PanelHeight],'string', '',...
'Backgroundcolor', Col_H3,'HorizontalAlignment',
'left','FontName', 'arial','FontWeight',
'normal','FontSize', Font_large);

% create a Static Text "Spacer Type"
uicontrol('Parent', Tab4_Pan_El_spec,'Style',
'text','Position', [0.25*m_PanelWidth
0.03*m_PanelHeight 0.06*m_PanelWidth
0.04*m_PanelHeight],'string', 'Spacer',...
'HorizontalAlignment', 'left','FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create an edit Text to display the "Spacer Type"
Tab4_edit_SpacerType = uicontrol('Parent',
Tab4_Pan_El_spec,'Style', 'edit', 'Enable',
'off','Position', [0.32*m_PanelWidth
0.03*m_PanelHeight 0.12*m_PanelWidth
0.05*m_PanelHeight],'string', '',...
'Backgroundcolor', Col_H3,'HorizontalAlignment',
'left','FontName', 'arial','FontWeight',
'normal','FontSize', Font_large);
```

```
% create a UIPanel "Inlet Flowrates"
Tab_4_Panel_FO = uipanel('Parent', Tab4_H1, 'Units',
'pixels','Visible', 'on','Backgroundcolor',
Col_H1,'BorderWidth',1.5, 'Title', 'Inlet
Flowrates',...
'FontSize', Font_large,'FontWeight', 'bold',
'Position',[0.015*m_PanelWidth 0.25*m_PanelHeight
0.45*m_PanelWidth 0.22*m_PanelHeight]);

% create a Static Text "Inlet Feed Solution Flowrate
(LPM)"
uicontrol('Parent', Tab_4_Panel_FO,'Style',
'text','Position', [0.015*m_PanelWidth
0.11*m_PanelHeight 0.2*m_PanelWidth
0.04*m_PanelHeight],...
'string', 'Inlet FS Flowrate
(LPM)','HorizontalAlignment', 'left','FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create an edit Text box for "Inlet Feed Solution
Flowrate Range (LPM)"
Tab4_Q_F_in_i = uicontrol('Parent',
Tab_4_Panel_FO,'Style', 'edit','Position',
[0.22*m_PanelWidth 0.11*m_PanelHeight
0.04*m_PanelWidth 0.043*m_PanelHeight],...
'HorizontalAlignment', 'center', 'Backgroundcolor',
Col_H3,'FontName', 'arial', 'FontWeight',
'normal','FontSize', Font_large);

% create a Static Text "Inlet Draw Solution Flowrate
Range (LPM)"
uicontrol('Parent', Tab_4_Panel_FO,'Style',
'text','Position', [0.015*m_PanelWidth
0.03*m_PanelHeight 0.2*m_PanelWidth
0.04*m_PanelHeight],...
'string', 'Inlet DS Flowrate Range
(LPM)','HorizontalAlignment', 'left','FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create an edit Text box for "Draw Solution Flowrate
Range"
Tab4_Q_D_in_i = uicontrol('Parent',
Tab_4_Panel_FO,'Style', 'edit','Position',
```

```
[0.22*m_PanelWidth 0.03*m_PanelHeight
0.04*m_PanelWidth 0.043*m_PanelHeight],...
'HorizontalAlignment', 'center', 'Backgroundcolor',
Col_H3,'FontName', 'arial', 'FontWeight',
'normal','FontSize', Font_large);

% create a Static Text "to"
uicontrol('Parent', Tab_4_Panel_FO,'Style',
'text','Position', [0.27*m_PanelWidth
0.03*m_PanelHeight 0.03*m_PanelWidth
0.04*m_PanelHeight],...
'string', 'to','HorizontalAlignment',
'left','FontName', 'arial','FontWeight',
'normal','FontSize', Font_large);

% create an edit Text box for "Draw Solution Flowrate
Range"
Tab4_Q_D_in_f = uicontrol('Parent',
Tab_4_Panel_FO,'Style', 'edit','Position',
[0.29*m_PanelWidth 0.03*m_PanelHeight
0.04*m_PanelWidth 0.043*m_PanelHeight],...
'HorizontalAlignment', 'center', 'Backgroundcolor',
Col_H3,'FontName', 'arial', 'FontWeight',
'normal','FontSize', Font_large);

% create a Static Text "Step"
uicontrol('Parent', Tab_4_Panel_FO,'Style',
'text','Position', [0.34*m_PanelWidth
0.03*m_PanelHeight 0.03*m_PanelWidth
0.04*m_PanelHeight],...
'string', 'Step','HorizontalAlignment',
'left','FontName', 'arial','FontWeight',
'normal','FontSize', Font_large);

% create an edit Text box for "Step"
Tab4_del_Q_D_in = uicontrol('Parent',
Tab_4_Panel_FO,'Style', 'edit','Position',
[0.38*m_PanelWidth 0.03*m_PanelHeight
0.04*m_PanelWidth 0.043*m_PanelHeight],...
'HorizontalAlignment', 'center', 'Backgroundcolor',
Col_H3,'FontName', 'arial', 'FontWeight',
'normal','FontSize', Font_large);

% create a UIPanel "Inlet Concentrations"
Tab_4_Panel_DO = uipanel('Parent', Tab4_H1, 'Units',
'pixels','Visible', 'on','Backgroundcolor',
```

```
Col_H1,'BorderWidth',1.5, 'Title', 'Inlet
Concentrations',...
'FontSize', Font_large,'FontWeight', 'bold',
'Position',[0.015*m_PanelWidth 0.02*m_PanelHeight
0.3*m_PanelWidth 0.22*m_PanelHeight]);

% create a Static Text "Inlet Feed Solution
Concentration (M)"
uicontrol('Parent', Tab_4_Panel_DO,'Style',
'text','Position', [0.015*m_PanelWidth
0.11*m_PanelHeight 0.2*m_PanelWidth
0.04*m_PanelHeight],...
'string', 'Inlet FS Concentration
(M)','HorizontalAlignment', 'left','FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create an edit Text box for "Inlet Feed Solution
Concentration (M)"
Tab4_C_F_in = uicontrol('Parent',
Tab_4_Panel_DO,'Style', 'edit','Enable', 'off',
'Position', [0.22*m_PanelWidth 0.11*m_PanelHeight
0.04*m_PanelWidth 0.043*m_PanelHeight],...
'HorizontalAlignment', 'center', 'Backgroundcolor',
Col_H3,'FontName', 'arial', 'FontWeight',
'normal','FontSize', Font_large);

% create a Static Text "Inlet Draw Solution
oncentration(M)"
uicontrol('Parent', Tab_4_Panel_DO,'Style',
'text','Position', [0.015*m_PanelWidth
0.03*m_PanelHeight 0.19*m_PanelWidth
0.04*m_PanelHeight],...
'string', 'Inlet DS Concentration
(M)','HorizontalAlignment', 'left','FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create an edit Text box for "Inlet Draw Solution
oncentration(M)"
Tab4_C_D_in = uicontrol('Parent',
Tab_4_Panel_DO,'Style', 'edit','Enable',
'off','Position', [0.22*m_PanelWidth
0.03*m_PanelHeight 0.04*m_PanelWidth
0.043*m_PanelHeight],...
```

```
'HorizontalAlignment', 'center', 'Backgroundcolor',
Col_H3,'FontName', 'arial', 'FontWeight',
'normal','FontSize', Font_large);

% create a pushbutton for "Optimization"
Tab4_button_Opt = uicontrol('Parent', Tab4_H1,'Style',
'pushbutton','Units', 'pixels','BackgroundColor',
Col_H2,'Position', [0.35*m_PanelWidth
0.07*m_PanelHeight ...
0.08*m_PanelWidth 0.1*m_PanelHeight],'String',
'Optimize','HorizontalAlignment', 'center','FontName',
'arial','FontWeight', 'bold','FontSize', Font_medium);

set(Tab4_button_Opt, 'callback',{@Opt_Callback,
Tab4_button_Opt});

% create a UIPanel "Optimum Parameters"
Tab_4_Panel_Opt = uipanel('Parent', Tab4_H1, 'Units',
'pixels','Visible', 'on','Backgroundcolor',
Col_H1,'BorderWidth',1.5, 'Title', 'Optimum
Parameters',...
'FontSize', Font_large,'FontWeight', 'bold',
'Position',[0.473*m_PanelWidth 0.02*m_PanelHeight
0.52*m_PanelWidth 0.655*m_PanelHeight]);

% create a UIPanel "Operating Conditions"
Tab_4_Panel_Opt_Cond = uipanel('Parent',
Tab_4_Panel_Opt, 'Units', 'pixels','Visible', 'on',
'Backgroundcolor', Col_H1,'BorderWidth',1.5,
'BorderType', 'beveledout','Title', 'Operating
Conditions','FontSize', Font_large,...
'FontWeight', 'bold', 'Position', [0.015*m_PanelWidth
0.37*m_PanelHeight 0.49*m_PanelWidth
0.22*m_PanelHeight]);

% create a Static Text "Optimum Inlet Draw Solution
Flowrate (LPM)"
uicontrol('Parent', Tab_4_Panel_Opt_Cond,'Style',
'text','Position', [0.05*m_PanelWidth
0.105*m_PanelHeight 0.28*m_PanelWidth
0.04*m_PanelHeight],...
'string', 'Optimum Inlet Draw Solution Flowrate
(LPM)','HorizontalAlignment', 'left','FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);
```

```
% create an edit text to display the "Optimum Inlet
Draw Solution Flowrate (LPM)"
Tab4_opt_Q_DS_in = uicontrol('Parent',
Tab_4_Panel_Opt_Cond,'Style', 'edit', 'Enable', 'off',
'Position', [0.37*m_PanelWidth 0.11*m_PanelHeight
0.05*m_PanelWidth 0.043*m_PanelHeight],...
'HorizontalAlignment', 'center', 'Backgroundcolor',
Col_H3,'FontName', 'arial', 'FontWeight',
'normal','FontSize', Font_large);

% create a Static Text "Optimum Number of  Elements
Per Pressure Vessel"
uicontrol('Parent', Tab_4_Panel_Opt_Cond,'Style',
'text','Position', [0.05*m_PanelWidth
0.03*m_PanelHeight 0.3*m_PanelWidth
0.04*m_PanelHeight],...
'string', 'Optimum Number of  Elements Per Pressure
Vessel','HorizontalAlignment', 'left','FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create an edit Text to display the "Optimum Number
of  Elements Per Pressure Vessel"
Tab4_opt_NE = uicontrol('Parent',
Tab_4_Panel_Opt_Cond,'Style', 'edit', 'Enable', 'off',
'Position', [0.37*m_PanelWidth 0.03*m_PanelHeight
0.05*m_PanelWidth 0.043*m_PanelHeight],...
'HorizontalAlignment', 'center', 'Backgroundcolor',
Col_H3,'FontName', 'arial', 'FontWeight',
'normal','FontSize', Font_large);

% create a UIPanel "Optimum Performances"
Tab_4_Panel_Opt_Perf = uipanel('Parent',
Tab_4_Panel_Opt, 'Units', 'pixels','Visible', 'on',
'Backgroundcolor', Col_H1,'BorderWidth',1.5,
'BorderType', 'beveledout',...
'Title', 'Optimum Performances','FontSize',
Font_large,'FontWeight', 'bold', 'Position',
[0.015*m_PanelWidth 0.03*m_PanelHeight
0.49*m_PanelWidth 0.3*m_PanelHeight]);

% create a Static Text "Optimum Recovery (%)"
uicontrol('Parent', Tab_4_Panel_Opt_Perf,'Style',
'text','Position', [0.05*m_PanelWidth
0.19*m_PanelHeight 0.28*m_PanelWidth
0.04*m_PanelHeight],...
```

```
'string', 'Optimum Recovery
(%)','HorizontalAlignment', 'left','FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create an edit text to display the "Optimum Recovery
(%)"
 Tab4_opt_Rec =  uicontrol('Parent',
Tab_4_Panel_Opt_Perf,'Style', 'edit', 'Enable', 'off',
'Position', [0.37*m_PanelWidth 0.19*m_PanelHeight
0.05*m_PanelWidth 0.043*m_PanelHeight],...
'HorizontalAlignment', 'center', 'Backgroundcolor',
Col_H3,'FontName', 'arial', 'FontWeight',
'normal','FontSize', Font_large);

% create a Static Text "Optimum Final Draw Solution
Concentration (M)"
uicontrol('Parent', Tab_4_Panel_Opt_Perf,'Style',
'text','Position', [0.05*m_PanelWidth
0.11*m_PanelHeight 0.28*m_PanelWidth
0.04*m_PanelHeight],...
'string', 'Optimum Final Draw Solution Concentration
(M)','HorizontalAlignment', 'left','FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create an edit Text to display the "Optimum Final
Draw Solution Concentration (M)"
Tab4_opt_C_DS = uicontrol('Parent',
Tab_4_Panel_Opt_Perf,'Style', 'edit', 'Enable', 'off',
'Position', [0.37*m_PanelWidth 0.11*m_PanelHeight
0.05*m_PanelWidth 0.043*m_PanelHeight],...
'HorizontalAlignment', 'center', 'Backgroundcolor',
Col_H3,'FontName', 'arial', 'FontWeight',
'normal','FontSize', Font_large);

% create a Static Text "OPR (%Objective/Element)"
uicontrol('Parent', Tab_4_Panel_Opt_Perf,'Style',
'text','Position', [0.05*m_PanelWidth
0.03*m_PanelHeight 0.3*m_PanelWidth
0.04*m_PanelHeight],...
'string', 'OPR
(%Objective/Element)','HorizontalAlignment',
'left','FontName', 'arial','FontWeight',
'normal','FontSize', Font_large);
```

```
% create an edit Text to display the "OPR
(%Objective/Element)"
 Tab4_opt_OPR = uicontrol('Parent',
Tab_4_Panel_Opt_Perf,'Style', 'edit', 'Enable', 'off',
'Position', [0.37*m_PanelWidth 0.03*m_PanelHeight
0.05*m_PanelWidth 0.043*m_PanelHeight],...
'HorizontalAlignment', 'center', 'Backgroundcolor',
Col_H3,'FontName', 'arial', 'FontWeight',
'normal','FontSize', Font_large);


%% Define Content for Tab5 (FO Configuration)

% create a Static Text for the title of the table
"System Configuration"
uicontrol('Style', 'text','Position',[0.4*m_PanelWidth
0.9*m_PanelHeight 0.25*m_PanelWidth
0.065*m_PanelHeight ],'Parent', Tab5_H1,'string',...
'System Configuration','BackgroundColor',
Col_H1,'HorizontalAlignment', 'center','FontName',
'arial','FontWeight', 'bold','FontSize',
Font_tab_title);

% create a UIPanel "Design Parameters"
Tab5_pan_Design = uipanel('Parent', Tab5_H1, 'Units',
'pixels','Visible', 'on','Backgroundcolor',
Col_H1,'BorderWidth',1.5, 'Title', 'Design
Parameters',...
'FontSize', Font_large,'FontWeight', 'bold',
'Position', [0.015*m_PanelWidth 0.78*m_PanelHeight
0.95*m_PanelWidth 0.12*m_PanelHeight]);

% create a Static Text "Production Capacity(m3/day)"
uicontrol('Parent', Tab5_pan_Design,'Style',
'text','Position', [0.02*m_PanelWidth
0.025*m_PanelHeight 0.17*m_PanelWidth
0.04*m_PanelHeight],...
'string', 'Production
Capacity(m3/day)','HorizontalAlignment',
'left','FontName', 'arial','FontWeight',
'normal','FontSize', Font_large);

% create an eidt Text to display the "Production
Capacity(%)"
Tab5_Prod_Cap = uicontrol('Parent',
Tab5_pan_Design,'Style', 'edit','Position',
```

```
[0.21*m_PanelWidth 0.03*m_PanelHeight 0.1*m_PanelWidth
0.043*m_PanelHeight],...
'HorizontalAlignment', 'center', 'Backgroundcolor',
Col_H3,'FontName', 'arial', 'FontWeight',
'normal','FontSize', Font_large);

% create a Static Text "Recovery(%)"
uicontrol('Parent', Tab5_pan_Design,'Style',
'text','Position', [0.36*m_PanelWidth
0.025*m_PanelHeight 0.08*m_PanelWidth
0.04*m_PanelHeight],...
'string', 'Recovery(%)','HorizontalAlignment',
'left','FontName', 'arial','FontWeight',
'normal','FontSize', Font_large);

% create an edit Text to display the "Recovery(%)"
Tab5_Rec = uicontrol('Parent',
Tab5_pan_Design,'Style', 'edit', 'Enable',
'off','Position', [0.46*m_PanelWidth
0.03*m_PanelHeight 0.06*m_PanelWidth
0.043*m_PanelHeight],...
'HorizontalAlignment', 'center', 'Backgroundcolor',
Col_H3,'FontName', 'arial', 'FontWeight',
'normal','FontSize', Font_large);

% create a Static Text "Final Draw Solution
Concentration(M)"
uicontrol('Parent', Tab5_pan_Design,'Style',
'text','Position', [0.6*m_PanelWidth
0.025*m_PanelHeight 0.22*m_PanelWidth
0.04*m_PanelHeight],...
'string', 'Final Draw Solution
Concentration(M)','HorizontalAlignment',
'left','FontName', 'arial','FontWeight',
'normal','FontSize', Font_large);

% create an edit Text to display the "Final Draw
Solution Concentration(M)"
Tab5_C_D_f = uicontrol('Parent',
Tab5_pan_Design,'Style', 'edit', 'Enable',
'off','Position', [0.84*m_PanelWidth
0.03*m_PanelHeight 0.06*m_PanelWidth
0.043*m_PanelHeight],...
'HorizontalAlignment', 'center', 'Backgroundcolor',
Col_H3,'FontName', 'arial', 'FontWeight',
'normal','FontSize', Font_large);
```

```
% create a UIPanel "Element Specification"
Tab5_pan_ElSp = uipanel('Parent', Tab5_H1, 'Units',
'pixels','Visible', 'on','Backgroundcolor',
Col_H1,'BorderWidth',1.5, 'Title', 'Element
Specification',...
'FontSize', Font_large,'FontWeight', 'bold',
'Position', [0.015*m_PanelWidth 0.42*m_PanelHeight
0.4347*m_PanelWidth 0.34*m_PanelHeight]);

% create a Static Text "Element Model"
uicontrol('Parent', Tab5_pan_ElSp,'Style',
'text','Position', [0.02*m_PanelWidth
0.215*m_PanelHeight 0.1*m_PanelWidth
0.04*m_PanelHeight],...
'string', 'Element Model','HorizontalAlignment',
'left','FontName', 'arial','FontWeight',
'normal','FontSize', Font_large);

% create an edit Text to display the "Element Model"
Tab5_El_Mod = uicontrol('Parent',
Tab5_pan_ElSp,'Style', 'edit', 'Enable',
'off','Position', [0.13*m_PanelWidth
0.22*m_PanelHeight 0.08*m_PanelWidth
0.043*m_PanelHeight],...
'HorizontalAlignment', 'center', 'Backgroundcolor',
Col_H3,'FontName', 'arial', 'FontWeight',
'normal','FontSize', Font_large);

% create a Static Text "Module Type"
uicontrol('Parent', Tab5_pan_ElSp,'Style',
'text','Position', [0.23*m_PanelWidth
0.215*m_PanelHeight 0.1*m_PanelWidth
0.04*m_PanelHeight],...
'string', 'Module Type','HorizontalAlignment',
'left','FontName', 'arial','FontWeight',
'normal','FontSize', Font_large);

% create an edit Text to display the "Module Type"
Tab5_Mod_Type = uicontrol('Parent',
Tab5_pan_ElSp,'Style', 'edit', 'Enable',
'off','Position', [0.34*m_PanelWidth
0.22*m_PanelHeight 0.08*m_PanelWidth
0.043*m_PanelHeight],...
```

```
'HorizontalAlignment', 'center', 'Backgroundcolor',
Col_H3,'FontName', 'arial', 'FontWeight',
'normal','FontSize', Font_large);

% create a Static Text "Membrane Type"
uicontrol('Parent', Tab5_pan_ElSp,'Style',
'text','Position', [0.02*m_PanelWidth
0.135*m_PanelHeight 0.1*m_PanelWidth
0.04*m_PanelHeight],...
'string', 'Membrane Type','HorizontalAlignment',
'left','FontName', 'arial','FontWeight',
'normal','FontSize', Font_large);

% create an edit Text to display the "Membrane Type"
Tab5_Mem_Type = uicontrol('Parent',
Tab5_pan_ElSp,'Style', 'edit', 'Enable',
'off','Position', [0.13*m_PanelWidth
0.14*m_PanelHeight 0.06*m_PanelWidth
0.043*m_PanelHeight],...
'HorizontalAlignment', 'center', 'Backgroundcolor',
Col_H3,'FontName', 'arial', 'FontWeight',
'normal','FontSize', Font_large);

% create a Static Text "Surface Area (m2)"
uicontrol('Parent', Tab5_pan_ElSp,'Style',
'text','Position', [0.23*m_PanelWidth
0.135*m_PanelHeight 0.12*m_PanelWidth
0.04*m_PanelHeight],...
'string', 'Surface Area (m2)','HorizontalAlignment',
'left','FontName', 'arial','FontWeight',
'normal','FontSize', Font_large);

% create an edit Text to display the "Surface Area
(m2)"
Tab5_Area = uicontrol('Parent', Tab5_pan_ElSp,'Style',
'edit', 'Enable', 'off','Position', [0.34*m_PanelWidth
0.14*m_PanelHeight 0.06*m_PanelWidth
0.043*m_PanelHeight],...
'HorizontalAlignment', 'center', 'Backgroundcolor',
Col_H3,'FontName', 'arial', 'FontWeight',
'normal','FontSize', Font_large);

% create a Static Text "Spacer Type"
uicontrol('Parent', Tab5_pan_ElSp,'Style',
'text','Position', [0.02*m_PanelWidth
```

```
0.055*m_PanelHeight 0.1*m_PanelWidth
0.04*m_PanelHeight],...
'string', 'Spacer Type','HorizontalAlignment',
'left','FontName', 'arial','FontWeight',
'normal','FontSize', Font_large);

% create an edit Text to display the "Spacer Type"
Tab5_Sp_Type = uicontrol('Parent',
Tab5_pan_ElSp,'Style', 'edit', 'Enable',
'off','Position', [0.13*m_PanelWidth
0.06*m_PanelHeight 0.06*m_PanelWidth
0.043*m_PanelHeight],...
'HorizontalAlignment', 'center', 'Backgroundcolor',
Col_H3,'FontName', 'arial', 'FontWeight',
'normal','FontSize', Font_large);

% create a Static Text "Spacer Thickness (mm)"
uicontrol('Parent', Tab5_pan_ElSp,'Style',
'text','Position', [0.23*m_PanelWidth
0.055*m_PanelHeight 0.14*m_PanelWidth
0.04*m_PanelHeight],...
'string', 'Spacer Thickness
(mm)','HorizontalAlignment', 'left','FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create an edit Text to display the "Spacer Thickness
(mm)"
Tab5_Sp_thickness = uicontrol('Parent',
Tab5_pan_ElSp,'Style', 'edit', 'Enable',
'off','Position', [0.37*m_PanelWidth
0.06*m_PanelHeight 0.04*m_PanelWidth
0.043*m_PanelHeight],...
'HorizontalAlignment', 'center', 'Backgroundcolor',
Col_H3,'FontName', 'arial', 'FontWeight',
'normal','FontSize', Font_large);

% create a UIPanel "Pump Efficiencies"
Tab5_pan_PmpEff = uipanel('Parent', Tab5_H1, 'Units',
'pixels','Visible', 'on','Backgroundcolor',
Col_H1,'BorderWidth',1.5, 'Title', 'Pump
Efficiencies',...
'FontSize', Font_large,'FontWeight', 'bold',
'Position', [0.015*m_PanelWidth 0.26*m_PanelHeight
0.4347*m_PanelWidth 0.14*m_PanelHeight]);
```

```
% create a Static Text "FS Pump Efficiency (%)"
uicontrol('Parent', Tab5_pan_PmpEff,'Style',
'text','Position', [0.02*m_PanelWidth
0.035*m_PanelHeight 0.14*m_PanelWidth
0.04*m_PanelHeight],...
'string', 'FS Pump Efficiency
(%)','HorizontalAlignment', 'left','FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create an edit Text to display the "FS Pump
Efficiency (%)"
Tab5_FS_PmpEff = uicontrol('Parent',
Tab5_pan_PmpEff,'Style', 'edit','Position',
[0.17*m_PanelWidth 0.04*m_PanelHeight
0.04*m_PanelWidth 0.043*m_PanelHeight],...
'HorizontalAlignment', 'center', 'Backgroundcolor',
Col_H3,'FontName', 'arial', 'FontWeight',
'normal','FontSize', Font_large);

% create a Static Text "DS Pump Efficiency (%)"
uicontrol('Parent', Tab5_pan_PmpEff,'Style',
'text','Position', [0.23*m_PanelWidth
0.035*m_PanelHeight 0.14*m_PanelWidth
0.04*m_PanelHeight],...
'string', 'DS Pump Efficiency
(%)','HorizontalAlignment', 'left','FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create an edit Text to display the "DS Pump
Efficiency (%)"
Tab5_DS_PmpEff = uicontrol('Parent',
Tab5_pan_PmpEff,'Style', 'edit','Position',
[0.38*m_PanelWidth 0.04*m_PanelHeight
0.04*m_PanelWidth 0.043*m_PanelHeight],...
'HorizontalAlignment', 'center', 'Backgroundcolor',
Col_H3,'FontName', 'arial', 'FontWeight',
'normal','FontSize', Font_large);


% create a UIPanel "Operating Conditions"
Tab5_pan_OpCond = uipanel('Parent', Tab5_H1, 'Units',
'pixels','Visible', 'on','Backgroundcolor',
Col_H1,'BorderWidth',1.5, 'Title', 'Operating
Conditions',...
```

```
'FontSize', Font_large,'FontWeight', 'bold',
'Position', [0.47*m_PanelWidth 0.26*m_PanelHeight
0.5*m_PanelWidth 0.5*m_PanelHeight]);

% create a Static Text "Number of Elements per
Pressure Vessel"
uicontrol('Parent', Tab5_pan_OpCond,'Style',
'text','Position', [0.02*m_PanelWidth
0.395*m_PanelHeight 0.25*m_PanelWidth
0.04*m_PanelHeight],...
'string', 'Number of Elements per Pressure
Vessel','HorizontalAlignment', 'left','FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create an edit Text to display the "Number of
Pressure Vessel"
Tab5_NE = uicontrol('Parent', Tab5_pan_OpCond,'Style',
'edit', 'Enable', 'off','Position', [0.28*m_PanelWidth
0.4*m_PanelHeight 0.06*m_PanelWidth
0.043*m_PanelHeight],...
'HorizontalAlignment', 'center', 'Backgroundcolor',
Col_H3,'FontName', 'arial', 'FontWeight',
'normal','FontSize', Font_large);

% create a UIPanel "Feed Solution (Inlet)"
Tab_5_Panel_FS_In = uipanel('Parent', Tab5_pan_OpCond,
'Units', 'pixels','Visible', 'on', 'Backgroundcolor',
Col_H1,'BorderWidth',1.5, 'BorderType', ...
'beveledout','Title', 'Feed Solution
(Inlet)','FontSize', Font_large,'FontWeight', 'bold',
'Position', [0.02*m_PanelWidth 0.21*m_PanelHeight
0.46*m_PanelWidth 0.18*m_PanelHeight]);

% create a Static Text "Feed Solution"
uicontrol('Parent', Tab_5_Panel_FS_In,'Style',
'text','Position', [0.01*m_PanelWidth
0.085*m_PanelHeight 0.1*m_PanelWidth
0.04*m_PanelHeight],...
'string', 'Feed Solution','HorizontalAlignment',
'left','FontName', 'arial','FontWeight',
'normal','FontSize', Font_large);

% create an edit Text to display the "Feed Solution"
Tab5_FS_Type = uicontrol('Parent',
Tab_5_Panel_FS_In,'Style', 'edit', 'Enable',
```

```
'off','Position', [0.12*m_PanelWidth
0.09*m_PanelHeight 0.115*m_PanelWidth
0.043*m_PanelHeight],...
'HorizontalAlignment', 'center', 'Backgroundcolor',
Col_H3,'FontName', 'arial', 'FontWeight',
'normal','FontSize', Font_large);

% create a Static Text "Flowrate (LPM)"
uicontrol('Parent', Tab_5_Panel_FS_In,'Style',
'text','Position', [0.24*m_PanelWidth
0.085*m_PanelHeight 0.1*m_PanelWidth
0.04*m_PanelHeight],...
'string', 'Flowrate (LPM)','HorizontalAlignment',
'left','FontName', 'arial','FontWeight',
'normal','FontSize', Font_large);

% create an edit Text to display the "Flowrate (LPM)"
Tab5_FS_Flr = uicontrol('Parent',
Tab_5_Panel_FS_In,'Style', 'edit', 'Enable',
'off','Position', [0.39*m_PanelWidth
0.09*m_PanelHeight 0.06*m_PanelWidth
0.043*m_PanelHeight],...
'HorizontalAlignment', 'center', 'Backgroundcolor',
Col_H3,'FontName', 'arial', 'FontWeight',
'normal','FontSize', Font_large);

% create a Static Text "Concentration (M)"
uicontrol('Parent', Tab_5_Panel_FS_In,'Style',
'text','Position', [0.01*m_PanelWidth
0.015*m_PanelHeight 0.12*m_PanelWidth
0.04*m_PanelHeight],...
'string', 'Concentration (M)','HorizontalAlignment',
'left','FontName', 'arial','FontWeight',
'normal','FontSize', Font_large);

% create an edit Text to display the "Concentration
(M)"
Tab5_FS_C = uicontrol('Parent',
Tab_5_Panel_FS_In,'Style', 'edit', 'Enable',
'off','Position', [0.12*m_PanelWidth
0.02*m_PanelHeight 0.06*m_PanelWidth
0.043*m_PanelHeight],...
'HorizontalAlignment', 'center', 'Backgroundcolor',
Col_H3,'FontName', 'arial', 'FontWeight',
'normal','FontSize', Font_large);
```

```
% create a Static Text "Osmotic Pressure (Bar)"
uicontrol('Parent', Tab_5_Panel_FS_In,'Style',
'text','Position', [0.24*m_PanelWidth
0.015*m_PanelHeight 0.14*m_PanelWidth
0.04*m_PanelHeight],...
'string', 'Osmotic Pressure
(Bar)','HorizontalAlignment', 'left','FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create an edit Text to display the "Osmotic Pressure
(Bar)"
Tab5_FS_OP = uicontrol('Parent',
Tab_5_Panel_FS_In,'Style', 'edit', 'Enable',
'off','Position', [0.39*m_PanelWidth
0.02*m_PanelHeight 0.06*m_PanelWidth
0.043*m_PanelHeight],...
'HorizontalAlignment', 'center', 'Backgroundcolor',
Col_H3,'FontName', 'arial', 'FontWeight',
'normal','FontSize', Font_large);

% create a UIPanel "Draw Solution (Inlet)"
Tab_5_Panel_DS_In = uipanel('Parent', Tab5_pan_OpCond,
'Units', 'pixels','Visible', 'on', 'Backgroundcolor',
Col_H1,'BorderWidth',1.5, 'BorderType', ...
'beveledout','Title', 'Draw Solution
(Inlet)','FontSize', Font_large,'FontWeight', 'bold',
'Position', [0.02*m_PanelWidth 0.02*m_PanelHeight
0.46*m_PanelWidth 0.18*m_PanelHeight]);

% create a Static Text "Draw Solution"
uicontrol('Parent', Tab_5_Panel_DS_In,'Style',
'text','Position', [0.01*m_PanelWidth
0.085*m_PanelHeight 0.1*m_PanelWidth
0.04*m_PanelHeight],...
'string', 'Darw Solution','HorizontalAlignment',
'left','FontName', 'arial','FontWeight',
'normal','FontSize', Font_large);

% create an edit Text to display the "Draw Solution"
Tab5_DS_Type = uicontrol('Parent',
Tab_5_Panel_DS_In,'Style', 'edit', 'Enable',
'off','Position', [0.12*m_PanelWidth
0.09*m_PanelHeight 0.115*m_PanelWidth
0.043*m_PanelHeight],...
```

```
'HorizontalAlignment', 'center', 'Backgroundcolor',
Col_H3,'FontName', 'arial', 'FontWeight',
'normal','FontSize', Font_large);

% create a Static Text "Flowrate (LPM)"
uicontrol('Parent', Tab_5_Panel_DS_In,'Style',
'text','Position', [0.24*m_PanelWidth
0.085*m_PanelHeight 0.1*m_PanelWidth
0.04*m_PanelHeight],...
'string', 'Flowrate (LPM)','HorizontalAlignment',
'left','FontName', 'arial','FontWeight',
'normal','FontSize', Font_large);

% create an edit Text to display the "Flowrate (LPM)"
Tab5_DS_Flr =  uicontrol('Parent',
Tab_5_Panel_DS_In,'Style', 'edit', 'Enable',
'off','Position', [0.39*m_PanelWidth
0.09*m_PanelHeight 0.06*m_PanelWidth
0.043*m_PanelHeight],...
'HorizontalAlignment', 'center', 'Backgroundcolor',
Col_H3,'FontName', 'arial', 'FontWeight',
'normal','FontSize', Font_large);

% create a Static Text "Concentration (M)"
uicontrol('Parent', Tab_5_Panel_DS_In,'Style',
'text','Position', [0.01*m_PanelWidth
0.015*m_PanelHeight 0.12*m_PanelWidth
0.04*m_PanelHeight],...
'string', 'Concentration (M)','HorizontalAlignment',
'left','FontName', 'arial','FontWeight',
'normal','FontSize', Font_large);

% create an edit Text to display the "Concentration
(M)"
Tab5_DS_C =  uicontrol('Parent',
Tab_5_Panel_DS_In,'Style', 'edit', 'Enable',
'off','Position', [0.12*m_PanelWidth
0.02*m_PanelHeight 0.06*m_PanelWidth
0.043*m_PanelHeight],...
'HorizontalAlignment', 'center', 'Backgroundcolor',
Col_H3,'FontName', 'arial', 'FontWeight',
'normal','FontSize', Font_large);

% create a Static Text "Osmotic Pressure (Bar)"
uicontrol('Parent', Tab_5_Panel_DS_In,'Style',
'text','Position', [0.24*m_PanelWidth
```

```
0.015*m_PanelHeight 0.14*m_PanelWidth
0.04*m_PanelHeight],...
'string', 'Osmotic Pressure
(Bar)','HorizontalAlignment', 'left','FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create an edit Text to display the "Osmotic Pressure
(Bar)"
Tab5_DS_OP =  uicontrol('Parent',
Tab_5_Panel_DS_In,'Style', 'edit', 'Enable',
'off','Position', [0.39*m_PanelWidth
0.02*m_PanelHeight 0.06*m_PanelWidth
0.043*m_PanelHeight],...
'HorizontalAlignment', 'center', 'Backgroundcolor',
Col_H3,'FontName', 'arial', 'FontWeight',
'normal','FontSize', Font_large);

% create a UIPanel "Flow Configuration"
Tab5_pan_Sys = uipanel('Parent', Tab5_H1, 'Units',
'pixels','Visible', 'on','Backgroundcolor',
Col_H1,'BorderWidth',1.5, 'Title', 'Flow
Configuration',...
'FontSize', Font_large,'FontWeight', 'bold',
'Position', [0.015*m_PanelWidth 0.02*m_PanelHeight
0.98*m_PanelWidth 0.22*m_PanelHeight]);

hImageAxes = axes('Parent', Tab5_pan_Sys,'Units',
'pixels','Position', [0.02*m_PanelWidth
0.01*m_PanelHeight 0.99*m_PanelWidth
0.19*m_PanelHeight], 'visible', 'off');


%% Define Content for Tab6 (FO Simulation)

% create a Static Text for the title of the tab
"Simulation"
uicontrol('Style', 'text','Position',[0.4*m_PanelWidth
0.9*m_PanelHeight 0.25*m_PanelWidth
0.065*m_PanelHeight ],'Parent', Tab6_H1,'string',...
'Simulation','BackgroundColor',
Col_H1,'HorizontalAlignment', 'center','FontName',
'arial','FontWeight', 'bold','FontSize',
Font_tab_title);

% create a UIPanel "Simulation Outputs Graphics"
```

```
Tab_6_Panel_Sim_Out = uipanel('Parent', Tab6_H1,
'Units', 'pixels','Visible', 'on', 'Backgroundcolor',
Col_H1,'BorderWidth',1.5, 'BorderType','beveledout',
...
'Title', '','FontSize', Font_large,'FontWeight',
'normal', 'Position', [0.02*m_PanelWidth
0.4*m_PanelHeight 0.95*m_PanelWidth
0.5*m_PanelHeight]);

% create a UIPanel for "System Requirements"
Tab_6_Panel_Sys_Req = uipanel('Parent', Tab6_H1,
'Units', 'pixels','Visible', 'on', 'Backgroundcolor',
Col_H1,'BorderWidth',1.5, 'BorderType','beveledout',
...
'Title', 'System Requirements','FontSize',
Font_large,'FontWeight', 'bold', 'Position',
[0.02*m_PanelWidth 0.02*m_PanelHeight
0.35*m_PanelWidth 0.37*m_PanelHeight]);

% create a Static Text "Number of Pressure Vessels"
uicontrol('Parent', Tab_6_Panel_Sys_Req,'Style',
'text','Position', [0.04*m_PanelWidth
0.25*m_PanelHeight 0.18*m_PanelWidth
0.04*m_PanelHeight],...
'string', 'Number of Pressure
Vessels','HorizontalAlignment', 'left','FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create an edit Text to display the "Number of
Pressure Vessels"
Tab6_NPV = uicontrol('Parent',
Tab_6_Panel_Sys_Req,'Style', 'edit', 'Enable', 'off',
'Position', [0.23*m_PanelWidth 0.25*m_PanelHeight
0.06*m_PanelWidth 0.04*m_PanelHeight],...
'HorizontalAlignment', 'center', 'Backgroundcolor',
Col_H3,'FontName', 'arial', 'FontWeight',
'normal','FontSize', Font_large);

% create a Static Text "Number of Elements"
uicontrol('Parent', Tab_6_Panel_Sys_Req,'Style',
'text','Position', [0.04*m_PanelWidth
0.16*m_PanelHeight 0.18*m_PanelWidth
0.04*m_PanelHeight],...
```

```
'string', 'Number of Elements','HorizontalAlignment',
'left','FontName', 'arial','FontWeight',
'normal','FontSize', Font_large);

% create an edit Text to display the "Number of
Elements"
Tab6_TNE = uicontrol('Parent',
Tab_6_Panel_Sys_Req,'Style', 'edit', 'Enable', 'off',
'Position', [0.23*m_PanelWidth 0.16*m_PanelHeight
0.06*m_PanelWidth 0.04*m_PanelHeight],...
'HorizontalAlignment', 'center', 'Backgroundcolor',
Col_H3,'FontName', 'arial', 'FontWeight',
'normal','FontSize', Font_large);

% create a Static Text "Total Membrane Area (m2)"
uicontrol('Parent', Tab_6_Panel_Sys_Req,'Style',
'text','Position', [0.04*m_PanelWidth
0.07*m_PanelHeight 0.18*m_PanelWidth
0.04*m_PanelHeight],...
'string', 'Total Membrane Area
(m2)','HorizontalAlignment', 'left','FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create an edit Text to display the "Total Membrane
Area (m2)"
Tab6_TArea = uicontrol('Parent',
Tab_6_Panel_Sys_Req,'Style', 'edit', 'Enable', 'off',
'Position', [0.23*m_PanelWidth 0.07*m_PanelHeight
0.06*m_PanelWidth 0.04*m_PanelHeight],...
'HorizontalAlignment', 'center', 'Backgroundcolor',
Col_H3,'FontName', 'arial', 'FontWeight',
'normal','FontSize', Font_large);

% create a UIPanel for "System Performance"
Tab_6_Panel_Sys_Perf = uipanel('Parent', Tab6_H1,
'Units', 'pixels','Visible', 'on', 'Backgroundcolor',
Col_H1,'BorderWidth',1.5, 'BorderType','beveledout',
...
'Title', 'System Performance','FontSize',
Font_large,'FontWeight', 'bold', 'Position',
[0.39*m_PanelWidth 0.02*m_PanelHeight
0.42*m_PanelWidth 0.37*m_PanelHeight]);

% create a Static Text "Diluted Draw Solution
Production (m3/day)"
```

```
uicontrol('Parent', Tab_6_Panel_Sys_Perf,'Style',
'text','Position', [0.04*m_PanelWidth
0.25*m_PanelHeight 0.25*m_PanelWidth
0.04*m_PanelHeight],...
'string', 'Diluted Draw Solution Production
(m3/day)','HorizontalAlignment', 'left','FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create an edit Text to display the "Diluted Draw
Solution Production (m3/day)"
Tab6_Q_DS = uicontrol('Parent',
Tab_6_Panel_Sys_Perf,'Style', 'edit', 'Enable', 'off',
'Position', [0.3*m_PanelWidth 0.25*m_PanelHeight
0.06*m_PanelWidth 0.04*m_PanelHeight],...
'HorizontalAlignment', 'center', 'Backgroundcolor',
Col_H3,'FontName', 'arial', 'FontWeight',
'normal','FontSize', Font_large);

% create a Static Text "Total Water Permeation
(m3/day)"
uicontrol('Parent', Tab_6_Panel_Sys_Perf,'Style',
'text','Position', [0.04*m_PanelWidth
0.16*m_PanelHeight 0.25*m_PanelWidth
0.04*m_PanelHeight],...
'string', 'Total Water Permeation
(m3/day)','HorizontalAlignment', 'left','FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create an edit Text to display the "Total Water
Permeation (m3/day)"
Tab6_Q_P = uicontrol('Parent',
Tab_6_Panel_Sys_Perf,'Style', 'edit', 'Enable', 'off',
'Position', [0.3*m_PanelWidth 0.16*m_PanelHeight
0.06*m_PanelWidth 0.04*m_PanelHeight],...
'HorizontalAlignment', 'center', 'Backgroundcolor',
Col_H3,'FontName', 'arial', 'FontWeight',
'normal','FontSize', Font_large);

% create a Static Text "Final Draw Solution
Concentration (M)"
uicontrol('Parent', Tab_6_Panel_Sys_Perf,'Style',
'text','Position', [0.04*m_PanelWidth
0.07*m_PanelHeight 0.25*m_PanelWidth
0.04*m_PanelHeight],...
```

```
'string', 'Final Draw Solution Concentration
(M)','HorizontalAlignment', 'left','FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create an edit Text to display the "Final Draw
Solution Concentration (M)"
Tab6_C_D_f = uicontrol('Parent',
Tab_6_Panel_Sys_Perf,'Style', 'edit', 'Enable', 'off',
'Position', [0.3*m_PanelWidth 0.07*m_PanelHeight
0.06*m_PanelWidth 0.04*m_PanelHeight],...
'HorizontalAlignment', 'center', 'Backgroundcolor',
Col_H3,'FontName', 'arial', 'FontWeight',
'normal','FontSize', Font_large);

% create a pushbutton for "Simulate"
Tab6_button_Sim = uicontrol('Parent', Tab6_H1,'Style',
'pushbutton','Units', 'pixels','BackgroundColor',
Col_H2,'Position', [0.83*m_PanelWidth
0.22*m_PanelHeight ...
0.14*m_PanelWidth 0.1*m_PanelHeight],'String',
'Simulate','HorizontalAlignment', 'center','FontName',
'arial','FontWeight', 'bold','FontSize', Font_medium);

set(Tab6_button_Sim, 'callback',{@Sim_Callback,
Tab6_button_Sim});

% create a pushbutton for "Export Data to Excel"
Tab6_button_ExpExcl = uicontrol('Parent',
Tab6_H1,'Style', 'pushbutton','Units',
'pixels','BackgroundColor', Col_H2,'Position',
[0.83*m_PanelWidth 0.06*m_PanelHeight ...
0.14*m_PanelWidth 0.1*m_PanelHeight],'String', 'Export
Data to Excel','HorizontalAlignment',
'center','FontName', 'arial','FontWeight',
'bold','FontSize', Font_medium);

set(Tab6_button_ExpExcl,
'callback',{@ExpExcl_Callback, Tab6_button_ExpExcl});

%% Define Content for Tab7 (Economic Analysis)

% create a Static Text for the title of the tab
"Economic Analysis"
uicontrol('Style', 'text','Position',[0.4*m_PanelWidth
0.912*m_PanelHeight 0.25*m_PanelWidth
0.065*m_PanelHeight ],'Parent', Tab7_H1,'string',...
```

```
'Economic Analysis','BackgroundColor',
Col_H1,'HorizontalAlignment', 'center','FontName',
'arial','FontWeight', 'bold','FontSize',
Font_tab_title);

% create a UIPanel "Operation Life"
Tab7_Pan_OpLfe = uipanel('Parent', Tab7_H1, 'Units',
'pixels','Visible', 'on','Backgroundcolor',
Col_H1,'BorderWidth',1.5, 'Title', 'Operation
Life',...
'FontSize', Font_title,'FontWeight', 'bold',
'Position', [0.03*m_PanelWidth 0.76*m_PanelHeight
0.95*m_PanelWidth 0.17*m_PanelHeight]);

% create a Static Text "Plant Life (Years)"
uicontrol('Parent', Tab7_Pan_OpLfe,'Style',
'text','Position', [0.015*m_PanelWidth
0.075*m_PanelHeight 0.12*m_PanelWidth
0.04*m_PanelHeight],'string', 'Plant Life (Years)',...
'HorizontalAlignment', 'left','FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create an edit Text box to display the "Plant Life
(Years)"
Tab7_Plant_Life = uicontrol('Parent',
Tab7_Pan_OpLfe,'Style', 'edit', 'Position',
[0.155*m_PanelWidth 0.08*m_PanelHeight
0.06*m_PanelWidth
0.0453*m_PanelHeight],'HorizontalAlignment',...
 'center','Backgroundcolor', Col_H3,'FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create a Static Text "Element Life (Years)"
uicontrol('Parent', Tab7_Pan_OpLfe,'Style',
'text','Position', [0.235*m_PanelWidth
0.075*m_PanelHeight 0.12*m_PanelWidth
0.04*m_PanelHeight],'string', 'Element Life
(Years)',...
'HorizontalAlignment', 'left','FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create an edit Text box to display the "Element Life
(Years)"
```

```
 Tab7_El_Life = uicontrol('Parent',
Tab7_Pan_OpLfe,'Style', 'edit', 'Position',
[0.375*m_PanelWidth 0.08*m_PanelHeight
0.06*m_PanelWidth
0.0453*m_PanelHeight],'HorizontalAlignment',...
 'center','Backgroundcolor', Col_H3,'FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create a Static Text "Pressure Vessel Life (Years)"
uicontrol('Parent', Tab7_Pan_OpLfe,'Style',
'text','Position', [0.455*m_PanelWidth
0.075*m_PanelHeight 0.17*m_PanelWidth
0.04*m_PanelHeight],'string', 'Pressure Vessel Life
(Years)',...
'HorizontalAlignment', 'left','FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create an edit Text box to display the "Pressure
Vessel Life (Years)"
 Tab7_PV_Life = uicontrol('Parent',
Tab7_Pan_OpLfe,'Style', 'edit', 'Position',
[0.645*m_PanelWidth 0.08*m_PanelHeight
0.06*m_PanelWidth
0.0453*m_PanelHeight],'HorizontalAlignment',...
 'center','Backgroundcolor', Col_H3,'FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create a Static Text "Pump Life (Years)"
uicontrol('Parent', Tab7_Pan_OpLfe,'Style',
'text','Position', [0.725*m_PanelWidth
0.075*m_PanelHeight 0.12*m_PanelWidth
0.04*m_PanelHeight],'string', 'Pump Life (Years)',...
'HorizontalAlignment', 'left','FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create an edit Text box to display the "Pump Life
(Years)"
 Tab7_Pmp_Life = uicontrol('Parent',
Tab7_Pan_OpLfe,'Style', 'edit', 'Position',
[0.865*m_PanelWidth 0.08*m_PanelHeight
0.06*m_PanelWidth
0.0453*m_PanelHeight],'HorizontalAlignment',...
```

```
 'center','Backgroundcolor', Col_H3,'FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create a Static Text "Average Plant Opertation Time
Per Day (Hours)"
uicontrol('Parent', Tab7_Pan_OpLfe,'Style',
'text','Position', [0.085*m_PanelWidth
0.015*m_PanelHeight 0.27*m_PanelWidth
0.04*m_PanelHeight],'string', 'Average Plant
Opertation Time Per Day (Hours)',...
'HorizontalAlignment', 'left','FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create an edit Text box to display the "Average
Plant Opertation Time Per Day (Hours)"
 Tab7_Op_Time = uicontrol('Parent',
Tab7_Pan_OpLfe,'Style', 'edit', 'Position',
[0.375*m_PanelWidth 0.02*m_PanelHeight
0.06*m_PanelWidth
0.0453*m_PanelHeight],'HorizontalAlignment',...
 'center','Backgroundcolor', Col_H3,'FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create a UIPanel "Expenses"
Tab7_Pan_Exp = uipanel('Parent', Tab7_H1, 'Units',
'pixels','Visible', 'on','Backgroundcolor',
Col_H1,'BorderWidth',1.5, 'Title', 'Expenses',...
'FontSize', Font_title,'FontWeight', 'bold',
'Position', [0.03*m_PanelWidth 0.28*m_PanelHeight
0.95*m_PanelWidth 0.48*m_PanelHeight]);

% create a UIPanel "CAPEX"
Tab_7_Panel_Capex = uipanel('Parent', Tab7_Pan_Exp,
'Units', 'pixels','Visible', 'on', 'Backgroundcolor',
Col_H1,'BorderWidth',1.5, 'BorderType','beveledout',
...
'Title', 'CAPEX','FontSize', Font_large,'FontWeight',
'bold', 'Position', [0.015*m_PanelWidth
0.26*m_PanelHeight 0.92*m_PanelWidth
0.17*m_PanelHeight]);

% create a Static Text "Element Price ($/El)"
```

```
uicontrol('Parent', Tab_7_Panel_Capex,'Style',
'text','Position', [0.015*m_PanelWidth
0.075*m_PanelHeight 0.12*m_PanelWidth
0.04*m_PanelHeight],'string', 'Element Price
($/El)',...
'HorizontalAlignment', 'left','FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create an edit Text box to display the "Element
Price ($/El)"
 Tab7_Cap_El_Price = uicontrol('Parent',
Tab_7_Panel_Capex,'Style', 'edit', 'Position',
[0.135*m_PanelWidth 0.08*m_PanelHeight
0.06*m_PanelWidth
0.0453*m_PanelHeight],'HorizontalAlignment',...
 'center','Backgroundcolor', Col_H3,'FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create a Static Text "Pressure Vessel Price ($/PV)"
uicontrol('Parent', Tab_7_Panel_Capex,'Style',
'text','Position', [0.215*m_PanelWidth
0.075*m_PanelHeight 0.165*m_PanelWidth
0.04*m_PanelHeight],'string', 'Pressure Vessel Price
($/PV)',...
'HorizontalAlignment', 'left','FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create an edit Text box to display the "Pressure
Vessel Price ($/PV)"
 Tab7_Cap_PV_Price = uicontrol('Parent',
Tab_7_Panel_Capex,'Style', 'edit', 'Position',
[0.39*m_PanelWidth 0.08*m_PanelHeight
0.06*m_PanelWidth
0.0453*m_PanelHeight],'HorizontalAlignment',...
 'center','Backgroundcolor', Col_H3,'FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create a Static Text "DS Pump Capcity (kW)"
uicontrol('Parent', Tab_7_Panel_Capex,'Style',
'text','Position', [0.47*m_PanelWidth
0.075*m_PanelHeight 0.135*m_PanelWidth
```

```
0.04*m_PanelHeight],'string', 'DS Pump Capacity
(kW)',...
'HorizontalAlignment', 'left','FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create an edit Text box to display the "DS Pump
Capcity (kW)"
 Tab7_Cap_DS_Pmp_Cap = uicontrol('Parent',
Tab_7_Panel_Capex,'Style', 'edit', 'Enable',
'off','Position', [0.615*m_PanelWidth
0.08*m_PanelHeight 0.06*m_PanelWidth
0.0453*m_PanelHeight],'HorizontalAlignment',...
 'center','Backgroundcolor', Col_H3,'FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create a Static Text "DS Pump Price ($/Pump)"
uicontrol('Parent', Tab_7_Panel_Capex,'Style',
'text','Position', [0.695*m_PanelWidth
0.075*m_PanelHeight 0.145*m_PanelWidth
0.04*m_PanelHeight],'string', 'DS Pump Price
($/Pump)',...
'HorizontalAlignment', 'left','FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create an edit Text box to display the "DS Pump
Price ($/Pump)"
Tab7_Cap_DS_Pmp_Price = uicontrol('Parent',
Tab_7_Panel_Capex,'Style', 'edit', 'Position',
[0.85*m_PanelWidth 0.08*m_PanelHeight
0.06*m_PanelWidth
0.0453*m_PanelHeight],'HorizontalAlignment',...
 'center','Backgroundcolor', Col_H3,'FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create a Static Text "Land Cost ($)"
uicontrol('Parent', Tab_7_Panel_Capex,'Style',
'text','Position', [0.015*m_PanelWidth
0.005*m_PanelHeight 0.12*m_PanelWidth
0.04*m_PanelHeight],'string', 'Land Cost ($)',...
'HorizontalAlignment', 'left','FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);
```

```
% create an edit Text box to display the "Land Cost
($))"
 Tab7_Cap_Land_Cost = uicontrol('Parent',
Tab_7_Panel_Capex,'Style', 'edit', 'Position',
[0.135*m_PanelWidth 0.01*m_PanelHeight
0.06*m_PanelWidth
0.0453*m_PanelHeight],'HorizontalAlignment',...
 'center','Backgroundcolor', Col_H3,'FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create a Static Text "Infrastructure Cost ($)"
uicontrol('Parent', Tab_7_Panel_Capex,'Style',
'text','Position', [0.215*m_PanelWidth
0.01*m_PanelHeight 0.165*m_PanelWidth
0.04*m_PanelHeight],'string', 'Infrastructure Cost
($)',...
'HorizontalAlignment', 'left','FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create an edit Text box to display the
"Infrastructure Cost ($)"
Tab7_Cap_Inf_Cost = uicontrol('Parent',
Tab_7_Panel_Capex,'Style', 'edit', 'Position',
[0.39*m_PanelWidth 0.01*m_PanelHeight
0.06*m_PanelWidth
0.0453*m_PanelHeight],'HorizontalAlignment',...
'center','Backgroundcolor', Col_H3,'FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create a Static Text "FS Pump Capcity (kW)"
uicontrol('Parent', Tab_7_Panel_Capex,'Style',
'text','Position', [0.47*m_PanelWidth
0.01*m_PanelHeight 0.135*m_PanelWidth
0.04*m_PanelHeight],'string', 'FS Pump Capacity
(kW)',...
'HorizontalAlignment', 'left','FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create an edit Text box to display the "FS Pump
Capcity (kW)"
```

```
Tab7_FS_Pmp_Cap = uicontrol('Parent',
Tab_7_Panel_Capex,'Style', 'edit', 'Enable',
'off','Position', [0.615*m_PanelWidth
0.01*m_PanelHeight 0.06*m_PanelWidth
0.0453*m_PanelHeight],'HorizontalAlignment',...
'center','Backgroundcolor', Col_H3,'FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create a Static Text "FS Pump Price ($/Pump)"
uicontrol('Parent', Tab_7_Panel_Capex,'Style',
'text','Position', [0.695*m_PanelWidth
0.01*m_PanelHeight 0.145*m_PanelWidth
0.04*m_PanelHeight],'string', 'FS Pump Price
($/Pump)',...
'HorizontalAlignment', 'left','FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create an edit Text box to display the "FS Pump
Price ($/Pump)"
Tab7_Cap_FS_Pmp_Price =  uicontrol('Parent',
Tab_7_Panel_Capex,'Style', 'edit', 'Position',
[0.85*m_PanelWidth 0.01*m_PanelHeight
0.06*m_PanelWidth
0.0453*m_PanelHeight],'HorizontalAlignment',...
'center','Backgroundcolor', Col_H3,'FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create a UIPanel "OPEX"
Tab_7_Panel_Opex = uipanel('Parent', Tab7_Pan_Exp,
'Units', 'pixels','Visible', 'on', 'Backgroundcolor',
Col_H1,'BorderWidth',1.5, 'BorderType','beveledout',
...
'Title', 'OPEX','FontSize', Font_large,'FontWeight',
'bold', 'Position', [0.015*m_PanelWidth
0.085*m_PanelHeight 0.92*m_PanelWidth
0.17*m_PanelHeight]);

% create a Static Text "Unit Power Cost ($/kWh)"
uicontrol('Parent', Tab_7_Panel_Opex,'Style',
'text','Position', [0.015*m_PanelWidth
0.075*m_PanelHeight 0.145*m_PanelWidth
0.04*m_PanelHeight],'string', 'Unit Power Cost
($/kWh)',...
```

```
'HorizontalAlignment', 'left','FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create an edit Text box to display the "Unit Power
Cost ($/kWh)"
 Tab7_Op_Pwr_Cost = uicontrol('Parent',
Tab_7_Panel_Opex,'Style', 'edit', 'Position',
[0.215*m_PanelWidth 0.08*m_PanelHeight
0.06*m_PanelWidth
0.0453*m_PanelHeight],'HorizontalAlignment',...
 'center','Backgroundcolor', Col_H3,'FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create a Static Text "DS Pump Replacement Cost
($/Pump)"
uicontrol('Parent', Tab_7_Panel_Opex,'Style',
'text','Position', [0.295*m_PanelWidth
0.075*m_PanelHeight 0.215*m_PanelWidth
0.04*m_PanelHeight],'string', 'DS Pump Replacement
Cost ($/Pump)',...
'HorizontalAlignment', 'left','FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create an edit Text box to display the "DS Pump
Replacement Cost ($/Pump)"
 Tab7_Op_DS_Pump_Rep = uicontrol('Parent',
Tab_7_Panel_Opex,'Style', 'edit', 'Position',
[0.545*m_PanelWidth 0.08*m_PanelHeight
0.06*m_PanelWidth
0.0453*m_PanelHeight],'HorizontalAlignment',...
 'center','Backgroundcolor', Col_H3,'FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create a Static Text "FS Pump Replacement Cost
($/Pump)"
uicontrol('Parent', Tab_7_Panel_Opex,'Style',
'text','Position', [0.625*m_PanelWidth
0.075*m_PanelHeight 0.215*m_PanelWidth
0.04*m_PanelHeight],'string', 'FS Pump Replacement
Cost ($/Pump)',...
```

```
'HorizontalAlignment', 'left','FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create an edit Text box to display the "FS Pump
Replacement Cost ($/Pump)"
Tab7_Op_FS_Pump_Rep = uicontrol('Parent',
Tab_7_Panel_Opex,'Style', 'edit', 'Position',
[0.855*m_PanelWidth 0.08*m_PanelHeight
0.06*m_PanelWidth
0.0453*m_PanelHeight],'HorizontalAlignment',...
'center','Backgroundcolor', Col_H3,'FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create a Static Text "Element Replacement Cost
($/El)"
uicontrol('Parent', Tab_7_Panel_Opex,'Style',
'text','Position', [0.015*m_PanelWidth
0.005*m_PanelHeight 0.19*m_PanelWidth
0.04*m_PanelHeight],'string', 'Element Replacement
Cost ($/El)',...
'HorizontalAlignment', 'left','FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create an edit Text box to display the "Element
Replacement Cost ($/El)"
Tab7_Op_El_Rep = uicontrol('Parent',
Tab_7_Panel_Opex,'Style', 'edit', 'Position',
[0.215*m_PanelWidth 0.01*m_PanelHeight
0.06*m_PanelWidth
0.0453*m_PanelHeight],'HorizontalAlignment',...
'center','Backgroundcolor', Col_H3,'FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create a Static Text "Pressure Vessel Replacement
Cost ($/PV)"
uicontrol('Parent', Tab_7_Panel_Opex,'Style',
'text','Position', [0.295*m_PanelWidth
0.005*m_PanelHeight 0.24*m_PanelWidth
0.04*m_PanelHeight],'string', 'Pressure Vessel
Replacement Cost ($/PV)',...
```

```
'HorizontalAlignment', 'left','FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create an edit Text box to display the "Pressure
Vessel Replacement Cost ($/PV)"
Tab7_Op_PV_Rep =  uicontrol('Parent',
Tab_7_Panel_Opex,'Style', 'edit', 'Position',
[0.545*m_PanelWidth 0.01*m_PanelHeight
0.06*m_PanelWidth
0.0453*m_PanelHeight],'HorizontalAlignment',...
'center','Backgroundcolor', Col_H3,'FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create a Static Text "Employee Expenses Per Hour($)"
uicontrol('Parent', Tab_7_Panel_Opex,'Style',
'text','Position', [0.625*m_PanelWidth
0.005*m_PanelHeight 0.23*m_PanelWidth
0.04*m_PanelHeight],'string', 'Employee Expenses Per
Hour ($)',...
'HorizontalAlignment', 'left','FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create an edit Text box to display the "Annual
Employee Expenses ($)"
Tab7_Op_Emp_Exp = uicontrol('Parent',
Tab_7_Panel_Opex,'Style', 'edit', 'Position',
[0.855*m_PanelWidth 0.01*m_PanelHeight
0.06*m_PanelWidth
0.0453*m_PanelHeight],'HorizontalAlignment',...
 'center','Backgroundcolor', Col_H3,'FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create a Static Text "Net Energy Consumption (kWh)"
uicontrol('Parent', Tab7_Pan_Exp,'Style',
'text','Position', [0.03*m_PanelWidth
0.015*m_PanelHeight 0.175*m_PanelWidth
0.04*m_PanelHeight],'string', 'Net Energy Consumption
(kWh)',...
'HorizontalAlignment', 'left','FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);
```

```
% create an edit Text box to display the "Net Energy
Consumption (kWh)"
Tab7_Net_En_Cons = uicontrol('Parent',
Tab7_Pan_Exp,'Style', 'edit', 'Enable',
'off','Position', [0.23*m_PanelWidth
0.02*m_PanelHeight 0.1*m_PanelWidth
0.0453*m_PanelHeight],'HorizontalAlignment',...
 'center','Backgroundcolor', Col_H3,'FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create a Static Text "Net CAPEX ($)"
uicontrol('Parent', Tab7_Pan_Exp,'Style',
'text','Position', [0.45*m_PanelWidth
0.015*m_PanelHeight 0.09*m_PanelWidth
0.04*m_PanelHeight],'string', 'Net CAPEX ($)',...
'HorizontalAlignment', 'left','FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create an edit Text box to display the "Net CAPEX
($)"
Tab7_Net_Capex = uicontrol('Parent',
Tab7_Pan_Exp,'Style', 'edit', 'Enable',
'off','Position', [0.56*m_PanelWidth
0.02*m_PanelHeight 0.1*m_PanelWidth
0.0453*m_PanelHeight],'HorizontalAlignment',...
'center','Backgroundcolor', Col_H3,'FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create a Static Text "Net OPEX ($)"
uicontrol('Parent', Tab7_Pan_Exp,'Style',
'text','Position', [0.73*m_PanelWidth
0.015*m_PanelHeight 0.08*m_PanelWidth
0.04*m_PanelHeight],'string', 'Net OPEX ($)',...
'HorizontalAlignment', 'left','FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create an edit Text box to display the "Net OPEX
($)"
Tab7_Net_Opex =  uicontrol('Parent',
Tab7_Pan_Exp,'Style', 'edit', 'Enable', 'off',
'Position', [0.83*m_PanelWidth 0.02*m_PanelHeight
```

```
0.1*m_PanelWidth
0.0453*m_PanelHeight],'HorizontalAlignment',...
'center','Backgroundcolor', Col_H3,'FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create a UIPanel "Investment Options"
Tab7_Pan_Inv = uipanel('Parent', Tab7_H1, 'Units',
'pixels','Visible', 'on','Backgroundcolor',
Col_H1,'BorderWidth',1.5, 'Title', 'Investment
Options',...
'FontSize', Font_title,'FontWeight', 'bold',
'Position', [0.03*m_PanelWidth 0.15*m_PanelHeight
0.60*m_PanelWidth 0.12*m_PanelHeight]);

% create a Static Text "Capex Loan (%)"
uicontrol('Parent', Tab7_Pan_Inv,'Style',
'text','Position', [0.015*m_PanelWidth
0.015*m_PanelHeight 0.1*m_PanelWidth
0.04*m_PanelHeight],'string', 'Capex Loan (%)',...
'HorizontalAlignment', 'left','FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create an edit Text box to display the "Capex Loan
(%)"
Tab7_Loan_Capex = uicontrol('Parent',
Tab7_Pan_Inv,'Style', 'edit', 'Position',
[0.125*m_PanelWidth 0.02*m_PanelHeight
0.06*m_PanelWidth
0.0453*m_PanelHeight],'HorizontalAlignment',...
 'center','Backgroundcolor', Col_H3,'FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create a Static Text "Opex Loan (%)"
uicontrol('Parent', Tab7_Pan_Inv,'Style',
'text','Position', [0.205*m_PanelWidth
0.015*m_PanelHeight 0.1*m_PanelWidth
0.04*m_PanelHeight],'string', 'Opex Loan (%)',...
'HorizontalAlignment', 'left','FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create an edit Text box to display the "Opex Loan
(%)"
```

```
Tab7_Loan_Opex = uicontrol('Parent',
Tab7_Pan_Inv,'Style', 'edit', 'Position',
[0.315*m_PanelWidth 0.02*m_PanelHeight
0.06*m_PanelWidth
0.0453*m_PanelHeight],'HorizontalAlignment',...
'center','Backgroundcolor', Col_H3,'FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create a Static Text "Rate of Interest (%)"
uicontrol('Parent', Tab7_Pan_Inv,'Style',
'text','Position', [0.395*m_PanelWidth
0.015*m_PanelHeight 0.11*m_PanelWidth
0.04*m_PanelHeight],'string', 'Rate of Interest
(%)',...
'HorizontalAlignment', 'left','FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create an edit Text box to display the "Rate of
Interest (%)"
Tab7_Loan_ROI = uicontrol('Parent',
Tab7_Pan_Inv,'Style', 'edit', 'Position',
[0.52*m_PanelWidth 0.02*m_PanelHeight
0.06*m_PanelWidth
0.0453*m_PanelHeight],'HorizontalAlignment',...
 'center','Backgroundcolor', Col_H3,'FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create a pushbutton for "Analyze"
Tab7_button_Analyze = uicontrol('Parent',
Tab7_H1,'Style', 'pushbutton','Units',
'pixels','BackgroundColor', Col_H2,'Position',
[0.68*m_PanelWidth 0.155*m_PanelHeight ...
0.1*m_PanelWidth 0.1*m_PanelHeight],'String',
'Analyze','HorizontalAlignment', 'center','FontName',
'arial','FontWeight', 'bold','FontSize', Font_medium);

set(Tab7_button_Analyze,
'callback',{@Tech_Eco_Callback, Tab7_button_Analyze});

% create a pushbutton for "Save Report"
Tab7_button_SvRpt = uicontrol('Parent',
Tab7_H1,'Style', 'pushbutton','Units',
```

```
'pixels','BackgroundColor', Col_H2,'Position',
[0.83*m_PanelWidth 0.155*m_PanelHeight ...
0.1*m_PanelWidth 0.1*m_PanelHeight],'String', 'Save
Report','HorizontalAlignment', 'center','FontName',
'arial','FontWeight', 'bold','FontSize', Font_medium);

set(Tab7_button_SvRpt, 'callback',{@Sim_Callback,
Tab7_button_SvRpt});

% create a UIPanel "Techno-Economic Report"
Tab7_Pan_TcEco = uipanel('Parent', Tab7_H1, 'Units',
'pixels','Visible', 'on','Backgroundcolor',
Col_H1,'BorderWidth',1.5, 'Title', 'Techno-Economic
Report',...
'FontSize', Font_title,'FontWeight', 'bold',
'Position', [0.03*m_PanelWidth 0.02*m_PanelHeight
0.95*m_PanelWidth 0.12*m_PanelHeight]);

% create a Static Text "Sp. Energy Consumption
(kWh/m3)"
uicontrol('Parent', Tab7_Pan_TcEco,'Style',
'text','Position', [0.015*m_PanelWidth
0.015*m_PanelHeight 0.20*m_PanelWidth
0.04*m_PanelHeight],'string', 'Sp. Energy Consumption
(kWh/m3)',...
'HorizontalAlignment', 'left','FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create an edit Text box to display the "Sp. Energy
Consumption (kWh/m3)"
Tab7_Sp_En_Cons = uicontrol('Parent',
Tab7_Pan_TcEco,'Style', 'edit', 'Enable',
'off','Position', [0.22*m_PanelWidth
0.02*m_PanelHeight 0.06*m_PanelWidth
0.0453*m_PanelHeight],'HorizontalAlignment',...
'center','Backgroundcolor', Col_H3,'FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create a Static Text "Unit CAPEX ($/m3)"
uicontrol('Parent', Tab7_Pan_TcEco,'Style',
'text','Position', [0.30*m_PanelWidth
0.015*m_PanelHeight 0.11*m_PanelWidth
0.04*m_PanelHeight],'string', 'Unit CAPEX ($/m3)',...
```

```
'HorizontalAlignment', 'left','FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create an edit Text box to display the "Unit CAPEX
($/m3)"
Tab7_Unit_Capex = uicontrol('Parent',
Tab7_Pan_TcEco,'Style', 'edit', 'Enable',
'off','Position', [0.42*m_PanelWidth
0.02*m_PanelHeight 0.06*m_PanelWidth
0.0453*m_PanelHeight],'HorizontalAlignment',...
'center','Backgroundcolor', Col_H3,'FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create a Static Text "Unit OPEX ($/m3)"
uicontrol('Parent', Tab7_Pan_TcEco,'Style',
'text','Position', [0.51*m_PanelWidth
0.015*m_PanelHeight 0.11*m_PanelWidth
0.04*m_PanelHeight],'string', 'Unit OPEX ($/m3)',...
'HorizontalAlignment', 'left','FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create an edit Text box to display the "Unit OPEX
($/m3)"
Tab7_Unit_Opex =  uicontrol('Parent',
Tab7_Pan_TcEco,'Style', 'edit', 'Enable',
'off','Position', [0.63*m_PanelWidth
0.02*m_PanelHeight 0.06*m_PanelWidth
0.0453*m_PanelHeight],'HorizontalAlignment',...
'center','Backgroundcolor', Col_H3,'FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

% create a Static Text "Unit Production Cost ($/m3)"
uicontrol('Parent', Tab7_Pan_TcEco,'Style',
'text','Position', [0.71*m_PanelWidth
0.015*m_PanelHeight 0.16*m_PanelWidth
0.04*m_PanelHeight],'string', 'Unit Production Cost
($/m3)',...
'HorizontalAlignment', 'left','FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);
```

```
% create an edit Text box to display the "Unit
Production Cost ($/m3)"
Tab7_Unit_Prod_Cost =  uicontrol('Parent',
Tab7_Pan_TcEco,'Style', 'edit', 'Enable',
'off','Position', [0.88*m_PanelWidth
0.02*m_PanelHeight 0.06*m_PanelWidth
0.0453*m_PanelHeight],'HorizontalAlignment',...
'center','Backgroundcolor', Col_H3,'FontName',
'arial','FontWeight', 'normal','FontSize',
Font_large);

end

function TabSellectCallback(~,~,SelectedTab)

global Col_H1 Col_H3
global Tab1_H1 Tab1_H2 Tab2_H1 Tab2_H2 Tab3_H1 Tab3_H2
Tab4_H1 Tab4_H2 Tab5_H1 Tab5_H2
global Tab6_H1 Tab6_H2 Tab7_H1  Tab7_H2 Tab2_TDS_text
Tab4_C_F_in Tab3_TDS_text Tab4_C_D_in

%   Disable the selected tab

set(Tab1_H1, 'Visible', 'off'); set(Tab2_H1,
'Visible', 'off'); set(Tab3_H1, 'Visible', 'off');
set(Tab4_H1, 'Visible', 'off'); set(Tab5_H1,
'Visible', 'off'); set(Tab6_H1, 'Visible', 'off');
set(Tab7_H1, 'Visible', 'off');
set(Tab1_H2, 'Backgroundcolor', Col_H1); set(Tab2_H2,
'Backgroundcolor', Col_H1);
set(Tab3_H2, 'Backgroundcolor', Col_H1); set(Tab4_H2,
'Backgroundcolor', Col_H1);
set(Tab5_H2, 'Backgroundcolor', Col_H1); set(Tab6_H2,
'Backgroundcolor', Col_H1);
set(Tab7_H2, 'Backgroundcolor', Col_H1);

%   Enable the selected tab
if (SelectedTab == Tab1_H2)

    set(Tab1_H1, 'Visible', 'on');
    set(Tab1_H2, 'Backgroundcolor', Col_H3);

elseif (SelectedTab == Tab2_H2)

    set(Tab2_H1, 'Visible', 'on');
    set(Tab2_H2, 'Backgroundcolor', Col_H3);
```

```
elseif (SelectedTab == Tab3_H2)

    set(Tab3_H1, 'Visible', 'on');
    set(Tab3_H2, 'Backgroundcolor', Col_H3);

elseif (SelectedTab == Tab4_H2)

    set(Tab4_H1, 'Visible', 'on');
    set(Tab4_H2, 'Backgroundcolor', Col_H3);

    if isempty(get(Tab2_TDS_text, 'string'))

        set(Tab4_C_F_in, 'string', '');

    else

        C_F_M = ((str2double(get(Tab2_TDS_text,
'string'))/1000)/58.443);
        set(Tab4_C_F_in, 'string',
num2str(C_F_M,'%0.2f'));

    end

     if isempty(get(Tab3_TDS_text, 'string'))

        set(Tab4_C_D_in, 'string', '');

    else

        C_D_M = ((str2double(get(Tab3_TDS_text,
'string'))/1000)/58.443);
        set(Tab4_C_D_in, 'string',
num2str(C_D_M,'%0.2f'));

    end

elseif (SelectedTab == Tab5_H2)

    set(Tab5_H1, 'Visible', 'on');
    set(Tab5_H2, 'Backgroundcolor', Col_H3);

elseif (SelectedTab == Tab6_H2)

    set(Tab6_H1, 'Visible', 'on');
    set(Tab6_H2, 'Backgroundcolor', Col_H3);

elseif (SelectedTab == Tab7_H2)
```

```
    set(Tab7_H1, 'Visible', 'on');
    set(Tab7_H2, 'Backgroundcolor', Col_H3);

end

end


function new_menu_Callback(~,~,file_H2)

global Prog_Location Prog_Name Prog_ext

filepath = strcat(Prog_Location,'\',
Prog_Name,Prog_ext);

run(filepath);

end

function exit_menu_Callback(~,~,file_H5)

global Fig_H1
close (Fig_H1);

end


function Co_Flow_Call_Callback(~,~,Tab1_co_flow)

global Bin_Location

filepath = strcat(Bin_Location,
'\cocurrent_flow_model.pdf');

winopen(filepath);

end

function
Counter_Flow_Call_Callback(~,~,Tab1_counter_flow)

global Bin_Location

filepath = strcat(Bin_Location,
'\countercurrent_flow_model.pdf');

winopen(filepath);
```

```
end


function SP_FS_Callback(~,~,Tab2_chk_box_1)

global Tbl_FS Tab2_FS_disp Tab2_edit_pH Tab2_edit_pOH

if (get(Tab2_chk_box_1, 'value') ==
get(Tab2_chk_box_1, 'max'))

    set(Tbl_FS,'ColumnEditable',logical([0 1 0 0 0]));
    set(Tab2_FS_disp, 'Enable', 'on');
    set(Tab2_edit_pH, 'Enable', 'on');
    set(Tab2_edit_pOH, 'Enable', 'on');

else

    set(Tbl_FS,'ColumnEditable',logical([0 0 0 0 0]));
    set(Tab2_FS_disp, 'Enable', 'off');
    set(Tab2_edit_pH, 'Enable', 'off');
    set(Tab2_edit_pOH, 'Enable', 'off');

end

end

function FS_Load_Callback (~,~,Tab2_FS_Load)

global Tbl_FS Tab2_edit_pH Tab2_edit_pOH Tab2_FS_disp
Lib_Location

Tbl_FS_Data_ret = get(Tbl_FS, 'Data');

N_ions = 15;
N_disp = 5;

for col = 2:1:N_disp

    for row = 1:1:N_ions

            Tbl_FS_Data_ret{row,col} = '';
    end

end

set(Tbl_FS, 'Data', Tbl_FS_Data_ret);
```

```
set(Tab2_edit_pH, 'string', '');
set(Tab2_edit_pOH, 'string', '');
set(Tab2_FS_disp, 'string', '');

feed_prop_path = strcat(Lib_Location, '\*.csv;*.dat');

[filename,filepath] = uigetfile(feed_prop_path,'Feed
Solution Properties');
s = strcat(filepath,filename);
fileID = fopen(s);

read_data = textscan(fileID,'%s %f', 'Delimiter',
',');

mg_L = read_data{1,2}(:,1);

col = 2;

for row = 1:1:N_ions

       Tbl_FS_Data_ret{row,col} = mg_L(row);

end

pH = mg_L(row+1);
pOH = mg_L(row+2);

FS_type = read_data{1,1}(row+3,1);
set(Tab2_edit_pH, 'string', pH);
set(Tab2_edit_pOH, 'string', pOH);
set(Tab2_FS_disp, 'string', FS_type);
set(Tbl_FS, 'Data', Tbl_FS_Data_ret);

end

function FS_Save_Callback(~,~,Tab2_FS_Save)

global Tbl_FS Tab2_FS_disp Tab2_edit_pH Tab2_edit_pOH
Lib_Location

string_1 = 'NA';

Tbl_FS_Data_ret = get(Tbl_FS, 'Data');

cnct_NH4 = Tbl_FS_Data_ret{1,2}; cnct_K =
Tbl_FS_Data_ret{2,2}; cnct_Na = Tbl_FS_Data_ret{3,2};
```

```
cnct_Mg = Tbl_FS_Data_ret{4,2}; cnct_Ca =
Tbl_FS_Data_ret{5,2}; cnct_Sr = Tbl_FS_Data_ret{6,2};
cnct_Ba = Tbl_FS_Data_ret{7,2}; cnct_B =
Tbl_FS_Data_ret{8,2}; cnct_CO3 = Tbl_FS_Data_ret{9,2};
cnct_HCO3 = Tbl_FS_Data_ret{10,2}; cnct_NO3 =
Tbl_FS_Data_ret{11,2}; cnct_Cl =
Tbl_FS_Data_ret{12,2};
cnct_F = Tbl_FS_Data_ret{13,2}; cnct_SO4 =
Tbl_FS_Data_ret{14,2}; cnct_SiO2 =
Tbl_FS_Data_ret{15,2};

cncts_NH4 = num2str(cnct_NH4); cncts_K =
num2str(cnct_K); cncts_Na = num2str(cnct_Na);
cncts_Mg = num2str(cnct_Mg); cncts_Ca =
num2str(cnct_Ca); cncts_Sr = num2str(cnct_Sr);
cncts_Ba = num2str(cnct_Ba); cncts_B =
num2str(cnct_B); cncts_CO3 = num2str(cnct_CO3);
cncts_HCO3 = num2str(cnct_HCO3); cncts_NO3 =
num2str(cnct_NO3); cncts_Cl = num2str(cnct_Cl);
cncts_F = num2str(cnct_F); cncts_SO4 =
num2str(cnct_SO4); cncts_SiO2 = num2str(cnct_SiO2);
FSpHS = get(Tab2_edit_pH, 'string'); FSpOHS=
get(Tab2_edit_pOH, 'string');
FS = get(Tab2_FS_disp, 'string'); FSTypeS = FS{1,1};


if strcmp(string_1,cncts_NH4)||isempty(cncts_NH4)
    cnctN_NH4 = 0;
else
    cnctN_NH4  = str2double(cncts_NH4);
end

if strcmp(string_1,cncts_K)||isempty(cncts_K)
    cnctN_K = 0;
else
    cnctN_K  = str2double(cncts_K);
end

if strcmp(string_1,cncts_Na)||isempty(cncts_Na)
    cnctN_Na = 0;
else
    cnctN_Na  = str2double(cncts_Na);
end

if strcmp(string_1,cncts_Mg)||isempty(cncts_Mg)
    cnctN_Mg = 0;
```

```
else
    cnctN_Mg   = str2double(cncts_Mg);
end

if strcmp(string_1,cncts_Ca)||isempty(cncts_Ca)
    cnctN_Ca = 0;
else
    cnctN_Ca   = str2double(cncts_Ca);
end

if strcmp(string_1,cncts_Sr)||isempty(cncts_Sr)
    cnctN_Sr = 0;
else
    cnctN_Sr   = str2double(cncts_Sr);
end

if strcmp(string_1,cncts_Ba)||isempty(cncts_Ba)
    cnctN_Ba = 0;
else
    cnctN_Ba   = str2double(cncts_Ba);
end

if strcmp(string_1,cncts_B)||isempty(cncts_B)
    cnctN_B = 0;
else
    cnctN_B = str2double(cncts_B);
end

if strcmp(string_1,cncts_CO3)||isempty(cncts_CO3)
    cnctN_CO3 = 0;
else
    cnctN_CO3   = str2double(cncts_CO3);
end

if strcmp(string_1,cncts_HCO3)||isempty(cncts_HCO3)
    cnctN_HCO3 = 0;
else
    cnctN_HCO3   = str2double(cncts_HCO3);
end

if strcmp(string_1,cncts_NO3)||isempty(cncts_NO3)
    cnctN_NO3 = 0;
else
    cnctN_NO3   = str2double(cncts_NO3);
end

if strcmp(string_1,cncts_Cl)||isempty(cncts_Cl)
```

```
    cnctN_Cl = 0;
else
    cnctN_Cl  = str2double(cncts_Cl);
end

if strcmp(string_1,cncts_F)||isempty(cncts_F)
    cnctN_F = 0;
else
    cnctN_F  = str2double(cncts_F);
end

if strcmp(string_1,cncts_SO4)||isempty(cncts_SO4)
    cnctN_SO4 = 0;
else
    cnctN_SO4  = str2double(cncts_SO4);
end

if strcmp(string_1,cncts_SiO2)||isempty(cncts_SiO2)
    cnctN_SiO2 = 0;
else
    cnctN_SiO2  = str2double(cncts_SiO2);
end

if strcmp(string_1,FSpHS)||isempty(FSpHS)
    FSpHN = 0;
else
    FSpHN = str2double(FSpHS);
end

if strcmp(string_1,FSpOHS)||isempty(FSpOHS)
    FSpOHN = 0;
else
    FSpOHN = str2double(FSpOHS);
end

save_fs_data = cell(18,2);

save_fs_data{1,1} = 'NH4'; save_fs_data{2,1} = 'K' ;
save_fs_data{3,1} = 'Na'; save_fs_data{4,1} = 'Mg';
save_fs_data{5,1} = 'Ca'; save_fs_data{6,1} = 'Sr';
save_fs_data{7,1} = 'Ba'; save_fs_data{8,1} = 'B';
save_fs_data{9,1} = 'CO3'; save_fs_data{10,1} =
'HCO3'; save_fs_data{11,1} = 'NO3'; save_fs_data{12,1}
= 'Cl';
save_fs_data{13,1} = 'F'; save_fs_data{14,1} = 'SO4';
save_fs_data{15,1} = 'SiO2'; save_fs_data{16,1} =
'pH';
```

```
save_fs_data{17,1} = 'pOH'; save_fs_data{18,1} =
FSTypeS;

save_fs_data{1,2} = cnctN_NH4; save_fs_data{2,2} =
cnctN_K ; save_fs_data{3,2} = cnctN_Na;
save_fs_data{4,2} = cnctN_Mg;
save_fs_data{5,2} = cnctN_Ca; save_fs_data{6,2} =
cnctN_Sr; save_fs_data{7,2} = cnctN_Ba;
save_fs_data{8,2} = cnctN_B;
save_fs_data{9,2} = cnctN_CO3; save_fs_data{10,2} =
cnctN_HCO3; save_fs_data{11,2} = cnctN_NO3;
save_fs_data{12,2} = cnctN_Cl;
save_fs_data{13,2} = cnctN_F; save_fs_data{14,2} =
cnctN_SO4; save_fs_data{15,2} = cnctN_SiO2;
save_fs_data{16,2} = FSpHN;
save_fs_data{17,2} = FSpOHN;

feed_prop_path = strcat(Lib_Location, '\*.csv;*.dat');

[filename filepath] = uiputfile(feed_prop_path,'Feed
Solution Properties');
FS_file_path = strcat(filepath,filename);
fileID = fopen(FS_file_path, 'w');

formatSpec = '%s,%d\n';

[nrows, ncols] = size(save_fs_data);

for row = 1:nrows

    fprintf(fileID,formatSpec, save_fs_data{row,:});

end

fclose(fileID);

end

function FS_Cal1_Callback(~,~,Tab2_cal1)

global Flag
global Tbl_FS Tab2_edit_temp Tab2_edit_pH
Tab2_edit_pOH Tab2_TDS_text Tab2_OP_FS TDS
global Cations Anions Balance_cat_an Tab2_Balance_text
Tab2_Cations_text Tab2_Anions_text

Flag = 1; string_1 = 'NA';
```

```
Tbl_FS_Data_ret = get(Tbl_FS, 'Data');

T_FS = get(Tab2_edit_temp, 'string');
pH_FS = get(Tab2_edit_pH, 'string');
pOH_FS = get(Tab2_edit_pOH, 'string');

cnct_NH4 = Tbl_FS_Data_ret{1,2}; cnct_K =
Tbl_FS_Data_ret{2,2}; cnct_Na = Tbl_FS_Data_ret{3,2};
cnct_Mg = Tbl_FS_Data_ret{4,2}; cnct_Ca =
Tbl_FS_Data_ret{5,2}; cnct_Sr = Tbl_FS_Data_ret{6,2};
cnct_Ba = Tbl_FS_Data_ret{7,2}; cnct_B =
Tbl_FS_Data_ret{8,2}; cnct_CO3 = Tbl_FS_Data_ret{9,2};
cnct_HCO3 = Tbl_FS_Data_ret{10,2}; cnct_NO3 =
Tbl_FS_Data_ret{11,2}; cnct_Cl =
Tbl_FS_Data_ret{12,2};
cnct_F = Tbl_FS_Data_ret{13,2}; cnct_SO4 =
Tbl_FS_Data_ret{14,2}; cnct_SiO2 =
Tbl_FS_Data_ret{15,2};

cncts_NH4 = num2str(cnct_NH4); cncts_K =
num2str(cnct_K); cncts_Na = num2str(cnct_Na);
cncts_Mg = num2str(cnct_Mg); cncts_Ca =
num2str(cnct_Ca); cncts_Sr = num2str(cnct_Sr);
cncts_Ba = num2str(cnct_Ba); cncts_B =
num2str(cnct_B); cncts_CO3 = num2str(cnct_CO3);
cncts_HCO3 = num2str(cnct_HCO3); cncts_NO3 =
num2str(cnct_NO3); cncts_Cl = num2str(cnct_Cl);
cncts_F = num2str(cnct_F); cncts_SO4 =
num2str(cnct_SO4); cncts_SiO2 = num2str(cnct_SiO2);


if strcmp(string_1,cncts_NH4)||isempty(cncts_NH4)
    cnctN_NH4 = 0;
else
    cnctN_NH4  = str2double(cncts_NH4);
end

if strcmp(string_1,cncts_K)||isempty(cncts_K)
    cnctN_K = 0;
else
    cnctN_K  = str2double(cncts_K);
end

if strcmp(string_1,cncts_Na)||isempty(cncts_Na)
    cnctN_Na = 0;
else
```

```
    cnctN_Na   = str2double(cncts_Na);
end

if strcmp(string_1,cncts_Mg)||isempty(cncts_Mg)
    cnctN_Mg = 0;
else
    cnctN_Mg   = str2double(cncts_Mg);
end

if strcmp(string_1,cncts_Ca)||isempty(cncts_Ca)
    cnctN_Ca = 0;
else
    cnctN_Ca   = str2double(cncts_Ca);
end

if strcmp(string_1,cncts_Sr)||isempty(cncts_Sr)
    cnctN_Sr = 0;
else
    cnctN_Sr   = str2double(cncts_Sr);
end

if strcmp(string_1,cncts_Ba)||isempty(cncts_Ba)
    cnctN_Ba = 0;
else
    cnctN_Ba   = str2double(cncts_Ba);
end

if strcmp(string_1,cncts_B)||isempty(cncts_B)
    cnctN_B = 0;
else
    cnctN_B = str2double(cncts_B);
end

if strcmp(string_1,cncts_CO3)||isempty(cncts_CO3)
    cnctN_CO3 = 0;
else
    cnctN_CO3   = str2double(cncts_CO3);
end

if strcmp(string_1,cncts_HCO3)||isempty(cncts_HCO3)
    cnctN_HCO3 = 0;
else
    cnctN_HCO3   = str2double(cncts_HCO3);
end

if strcmp(string_1,cncts_NO3)||isempty(cncts_NO3)
    cnctN_NO3 = 0;
```

```
else
    cnctN_NO3   = str2double(cncts_NO3);
end

if strcmp(string_1,cncts_Cl)||isempty(cncts_Cl)
    cnctN_Cl = 0;
else
    cnctN_Cl   = str2double(cncts_Cl);
end

if strcmp(string_1,cncts_F)||isempty(cncts_F)
    cnctN_F = 0;
else
    cnctN_F   = str2double(cncts_F);
end

if strcmp(string_1,cncts_SO4)||isempty(cncts_SO4)
    cnctN_SO4 = 0;
else
    cnctN_SO4   = str2double(cncts_SO4);
end

if strcmp(string_1,cncts_SiO2)||isempty(cncts_SiO2)
    cnctN_SiO2 = 0;
else
    cnctN_SiO2   = str2double(cncts_SiO2);
end

T_FS = get(Tab2_edit_temp, 'string');
pH_FS = get(Tab2_edit_pH, 'string');
pOH_FS = get(Tab2_edit_pOH, 'string');

if strcmp(string_1,T_FS)||isempty(T_FS)
    T_N_FS = 0;
else
    T_N_FS = str2double(T_FS);
end

if strcmp(string_1,pH_FS)||isempty(pH_FS)
    pH_N_FS = 0;
else
    pH_N_FS = str2double(pH_FS);
end

if strcmp(string_1,pOH_FS)||isempty(pOH_FS)
    pOH_N_FS = 0;
else
```

```
    pOH_N_FS = str2double(pOH_FS);
end

OP_bar =
calculate(cnctN_NH4,cnctN_K,cnctN_Na,cnctN_Mg,cnctN_Ca
,cnctN_Sr,cnctN_Ba,cnctN_B,cnctN_CO3,cnctN_HCO3,cnctN_
NO3,cnctN_Cl,cnctN_F,cnctN_SO4,cnctN_SiO2, T_N_FS,
pH_N_FS, pOH_N_FS);

set(Tab2_TDS_text, 'string', num2str(TDS,'%0.2f'));
set(Tab2_OP_FS, 'string', num2str(OP_bar,'%0.2f'));
set(Tab2_Cations_text, 'string',
num2str(Cations,'%0.2f'));
set(Tab2_Anions_text, 'string',
num2str(Anions,'%0.2f'));
set(Tab2_Balance_text, 'string',
num2str(Balance_cat_an,'%0.2f'));

end

function SP_DS_Callback(~,~,Tab3_chk_box_1)

global Tbl_DS Tab3_DS_disp Tab3_edit_pH Tab3_edit_pOH

if (get(Tab3_chk_box_1, 'value') ==
get(Tab3_chk_box_1, 'max'))

    set(Tbl_DS,'ColumnEditable',logical([0 1 0 0 0]));
    set(Tab3_DS_disp, 'Enable', 'on');
    set(Tab3_edit_pH, 'Enable', 'on');
    set(Tab3_edit_pOH, 'Enable', 'on');

else

    set(Tbl_DS,'ColumnEditable',logical([0 0 0 0 0]));
    set(Tab3_DS_disp, 'Enable', 'off');
    set(Tab3_edit_pH, 'Enable', 'off');
    set(Tab3_edit_pOH, 'Enable', 'off');

end

end

function DS_Load_Callback (~,~,Tab3_DS_Load)

global Tbl_DS Tab3_DS_disp Tab3_edit_pH Tab3_edit_pOH
Lib_Location
```

```
Tbl_DS_Data_ret = get(Tbl_DS, 'Data');

N_ions = 15;
N_disp = 5;

for col = 2:1:N_disp

    for row = 1:1:N_ions

       Tbl_DS_Data_ret{row,col} = '';

    end

end

set(Tbl_DS, 'Data', Tbl_DS_Data_ret);

set(Tab3_edit_pH, 'string', '');
set(Tab3_edit_pOH, 'string', '');
set(Tab3_DS_disp, 'string', '');

draw_prop_path = strcat(Lib_Location, '\*.csv;*.dat');

[filename, filepath] = uigetfile(draw_prop_path,'Draw
Solution Properties');
s = strcat(filepath,filename);
fileID = fopen(s);

read_data = textscan(fileID,'%s %f', 'Delimiter',
',');

mg_L = read_data{1,2}(:,1);

col = 2;

for row = 1:1:N_ions

       Tbl_DS_Data_ret{row,col} = mg_L(row);

end

pH = mg_L(row+1);
pOH = mg_L(row+2);

DS_type = read_data{1,1}(row+3,1);
set(Tab3_DS_disp, 'string', DS_type);
```

```
set(Tab3_edit_pH, 'string', pH);
set(Tab3_edit_pOH, 'string', pOH);
set(Tbl_DS, 'Data', Tbl_DS_Data_ret);

end

function DS_Save_Callback(~,~,Tab3_DS_Save)

global Tbl_DS Tab3_DS_disp Tab3_edit_pH Tab3_edit_pOH
Lib_Location

string_1 = 'NA';

Tbl_DS_Data_ret = get(Tbl_DS, 'Data');

cnct_NH4 = Tbl_DS_Data_ret{1,2}; cnct_K =
Tbl_DS_Data_ret{2,2}; cnct_Na = Tbl_DS_Data_ret{3,2};
cnct_Mg = Tbl_DS_Data_ret{4,2}; cnct_Ca =
Tbl_DS_Data_ret{5,2}; cnct_Sr = Tbl_DS_Data_ret{6,2};
cnct_Ba = Tbl_DS_Data_ret{7,2}; cnct_B =
Tbl_DS_Data_ret{8,2}; cnct_CO3 = Tbl_DS_Data_ret{9,2};
cnct_HCO3 = Tbl_DS_Data_ret{10,2}; cnct_NO3 =
Tbl_DS_Data_ret{11,2}; cnct_Cl =
Tbl_DS_Data_ret{12,2};
cnct_F = Tbl_DS_Data_ret{13,2}; cnct_SO4 =
Tbl_DS_Data_ret{14,2}; cnct_SiO2 =
Tbl_DS_Data_ret{15,2};

cncts_NH4 = num2str(cnct_NH4); cncts_K =
num2str(cnct_K); cncts_Na = num2str(cnct_Na);
cncts_Mg = num2str(cnct_Mg); cncts_Ca =
num2str(cnct_Ca); cncts_Sr = num2str(cnct_Sr);
cncts_Ba = num2str(cnct_Ba); cncts_B =
num2str(cnct_B); cncts_CO3 = num2str(cnct_CO3);
cncts_HCO3 = num2str(cnct_HCO3); cncts_NO3 =
num2str(cnct_NO3); cncts_Cl = num2str(cnct_Cl);
cncts_F = num2str(cnct_F); cncts_SO4 =
num2str(cnct_SO4); cncts_SiO2 = num2str(cnct_SiO2);
DSpHS = get(Tab3_edit_pH, 'string'); DSpOHS=
get(Tab3_edit_pOH, 'string');
DS = get(Tab3_DS_disp, 'string'); DSTypeS = DS{1,1};

if strcmp(string_1,cncts_NH4)||isempty(cncts_NH4)
    cnctN_NH4 = 0;
else
    cnctN_NH4  = str2double(cncts_NH4);
end
```

```
if strcmp(string_1,cncts_K)||isempty(cncts_K)
    cnctN_K = 0;
else
    cnctN_K  = str2double(cncts_K);
end

if strcmp(string_1,cncts_Na)||isempty(cncts_Na)
    cnctN_Na = 0;
else
    cnctN_Na  = str2double(cncts_Na);
end

if strcmp(string_1,cncts_Mg)||isempty(cncts_Mg)
    cnctN_Mg = 0;
else
    cnctN_Mg  = str2double(cncts_Mg);
end

if strcmp(string_1,cncts_Ca)||isempty(cncts_Ca)
    cnctN_Ca = 0;
else
    cnctN_Ca  = str2double(cncts_Ca);
end

if strcmp(string_1,cncts_Sr)||isempty(cncts_Sr)
    cnctN_Sr = 0;
else
    cnctN_Sr  = str2double(cncts_Sr);
end

if strcmp(string_1,cncts_Ba)||isempty(cncts_Ba)
    cnctN_Ba = 0;
else
    cnctN_Ba  = str2double(cncts_Ba);
end

if strcmp(string_1,cncts_B)||isempty(cncts_B)
    cnctN_B = 0;
else
    cnctN_B = str2double(cncts_B);
end

if strcmp(string_1,cncts_CO3)||isempty(cncts_CO3)
    cnctN_CO3 = 0;
else
    cnctN_CO3  = str2double(cncts_CO3);
```

```
end

if strcmp(string_1,cncts_HCO3)||isempty(cncts_HCO3)
    cnctN_HCO3 = 0;
else
    cnctN_HCO3  = str2double(cncts_HCO3);
end

if strcmp(string_1,cncts_NO3)||isempty(cncts_NO3)
    cnctN_NO3 = 0;
else
    cnctN_NO3  = str2double(cncts_NO3);
end

if strcmp(string_1,cncts_Cl)||isempty(cncts_Cl)
    cnctN_Cl = 0;
else
    cnctN_Cl  = str2double(cncts_Cl);
end

if strcmp(string_1,cncts_F)||isempty(cncts_F)
    cnctN_F = 0;
else
    cnctN_F  = str2double(cncts_F);
end

if strcmp(string_1,cncts_SO4)||isempty(cncts_SO4)
    cnctN_SO4 = 0;
else
    cnctN_SO4  = str2double(cncts_SO4);
end

if strcmp(string_1,cncts_SiO2)||isempty(cncts_SiO2)
    cnctN_SiO2 = 0;
else
    cnctN_SiO2  = str2double(cncts_SiO2);
end

if strcmp(string_1,DSpHS)||isempty(DSpHS)
    DSpHN = 0;
else
    DSpHN = str2double(DSpHS);
end

if strcmp(string_1,DSpOHS)||isempty(DSpOHS)
    DSpOHN = 0;
else
```

```
    DSpOHN = str2double(DSpOHS);
end

save_ds_data = cell(17,2);

save_ds_data{1,1} = 'NH4'; save_ds_data{2,1} = 'K' ;
save_ds_data{3,1} = 'Na'; save_ds_data{4,1} = 'Mg';
save_ds_data{5,1} = 'Ca'; save_ds_data{6,1} = 'Sr';
save_ds_data{7,1} = 'Ba'; save_ds_data{8,1} = 'B';
save_ds_data{9,1} = 'CO3'; save_ds_data{10,1} =
'HCO3'; save_ds_data{11,1} = 'NO3'; save_ds_data{12,1}
= 'Cl';
save_ds_data{13,1} = 'F'; save_ds_data{14,1} = 'SO4';
save_ds_data{15,1} = 'SiO2'; save_ds_data{16,1} =
'pH';
save_ds_data{17,1} = 'pOH'; save_ds_data{18,1} =
DSTypeS;

save_ds_data{1,2} = cnctN_NH4; save_ds_data{2,2} =
cnctN_K ; save_ds_data{3,2} = cnctN_Na;
save_ds_data{4,2} = cnctN_Mg;
save_ds_data{5,2} = cnctN_Ca; save_ds_data{6,2} =
cnctN_Sr; save_ds_data{7,2} = cnctN_Ba;
save_ds_data{8,2} = cnctN_B;
save_ds_data{9,2} = cnctN_CO3; save_ds_data{10,2} =
cnctN_HCO3; save_ds_data{11,2} = cnctN_NO3;
save_ds_data{12,2} = cnctN_Cl;
save_ds_data{13,2} = cnctN_F; save_ds_data{14,2} =
cnctN_SO4; save_ds_data{15,2} = cnctN_SiO2;
save_ds_data{16,2} = DSpHN;
save_ds_data{17,2} = DSpOHN;

draw_prop_path = strcat(Lib_Location, '\*.csv;*.dat');

[filename filepath] = uiputfile(draw_prop_path,'Draw
Solution Properties');
FS_file_path = strcat(filepath,filename);
fileID = fopen(FS_file_path, 'w');

formatSpec = '%s,%d\n';

[nrows, ncols] = size(save_ds_data);

for row = 1:nrows

    fprintf(fileID,formatSpec, save_ds_data{row,:});
```

```
end

fclose(fileID);

end

function DS_Cal1_Callback(~,~,Tab3_cal1)

global Flag
global Tbl_DS Tab3_edit_temp Tab3_edit_pH
Tab3_edit_pOH Tab3_TDS_text Tab3_OP_DS TDS
global Cations Anions Balance_cat_an Tab3_Balance_text
Tab3_Cations_text Tab3_Anions_text

string_1 = 'NA'; Flag = 2;

Tbl_DS_Data_ret = get(Tbl_DS, 'Data');

cnct_NH4 = Tbl_DS_Data_ret{1,2}; cnct_K =
Tbl_DS_Data_ret{2,2}; cnct_Na = Tbl_DS_Data_ret{3,2};
cnct_Mg = Tbl_DS_Data_ret{4,2}; cnct_Ca =
Tbl_DS_Data_ret{5,2}; cnct_Sr = Tbl_DS_Data_ret{6,2};
cnct_Ba = Tbl_DS_Data_ret{7,2}; cnct_B =
Tbl_DS_Data_ret{8,2}; cnct_CO3 = Tbl_DS_Data_ret{9,2};
cnct_HCO3 = Tbl_DS_Data_ret{10,2}; cnct_NO3 =
Tbl_DS_Data_ret{11,2}; cnct_Cl =
Tbl_DS_Data_ret{12,2};
cnct_F = Tbl_DS_Data_ret{13,2}; cnct_SO4 =
Tbl_DS_Data_ret{14,2}; cnct_SiO2 =
Tbl_DS_Data_ret{15,2};

cncts_NH4 = num2str(cnct_NH4); cncts_K =
num2str(cnct_K); cncts_Na = num2str(cnct_Na);
cncts_Mg = num2str(cnct_Mg); cncts_Ca =
num2str(cnct_Ca); cncts_Sr = num2str(cnct_Sr);
cncts_Ba = num2str(cnct_Ba); cncts_B =
num2str(cnct_B); cncts_CO3 = num2str(cnct_CO3);
cncts_HCO3 = num2str(cnct_HCO3); cncts_NO3 =
num2str(cnct_NO3); cncts_Cl = num2str(cnct_Cl);
cncts_F = num2str(cnct_F); cncts_SO4 =
num2str(cnct_SO4); cncts_SiO2 = num2str(cnct_SiO2);


if strcmp(string_1,cncts_NH4)||isempty(cncts_NH4)
    cnctN_NH4 = 0;
else
    cnctN_NH4  = str2double(cncts_NH4);
```

```
end

if strcmp(string_1,cncts_K)||isempty(cncts_K)
    cnctN_K = 0;
else
    cnctN_K  = str2double(cncts_K);
end

if strcmp(string_1,cncts_Na)||isempty(cncts_Na)
    cnctN_Na = 0;
else
    cnctN_Na  = str2double(cncts_Na);
end

if strcmp(string_1,cncts_Mg)||isempty(cncts_Mg)
    cnctN_Mg = 0;
else
    cnctN_Mg  = str2double(cncts_Mg);
end

if strcmp(string_1,cncts_Ca)||isempty(cncts_Ca)
    cnctN_Ca = 0;
else
    cnctN_Ca  = str2double(cncts_Ca);
end

if strcmp(string_1,cncts_Sr)||isempty(cncts_Sr)
    cnctN_Sr = 0;
else
    cnctN_Sr  = str2double(cncts_Sr);
end

if strcmp(string_1,cncts_Ba)||isempty(cncts_Ba)
    cnctN_Ba = 0;
else
    cnctN_Ba  = str2double(cncts_Ba);
end

if strcmp(string_1,cncts_B)||isempty(cncts_B)
    cnctN_B = 0;
else
    cnctN_B = str2double(cncts_B);
end

if strcmp(string_1,cncts_CO3)||isempty(cncts_CO3)
    cnctN_CO3 = 0;
else
```

```
    cnctN_CO3  = str2double(cncts_CO3);
end

if strcmp(string_1,cncts_HCO3)||isempty(cncts_HCO3)
    cnctN_HCO3 = 0;
else
    cnctN_HCO3  = str2double(cncts_HCO3);
end

if strcmp(string_1,cncts_NO3)||isempty(cncts_NO3)
    cnctN_NO3 = 0;
else
    cnctN_NO3  = str2double(cncts_NO3);
end

if strcmp(string_1,cncts_Cl)||isempty(cncts_Cl)
    cnctN_Cl = 0;
else
    cnctN_Cl  = str2double(cncts_Cl);
end

if strcmp(string_1,cncts_F)||isempty(cncts_F)
    cnctN_F = 0;
else
    cnctN_F  = str2double(cncts_F);
end

if strcmp(string_1,cncts_SO4)||isempty(cncts_SO4)
    cnctN_SO4 = 0;
else
    cnctN_SO4  = str2double(cncts_SO4);
end

if strcmp(string_1,cncts_SiO2)||isempty(cncts_SiO2)
    cnctN_SiO2 = 0;
else
    cnctN_SiO2  = str2double(cncts_SiO2);
end

T_DS = get(Tab3_edit_temp, 'string');
pH_DS = get(Tab3_edit_pH, 'string');
pOH_DS = get(Tab3_edit_pOH, 'string');

if strcmp(string_1,T_DS)||isempty(T_DS)
    T_N_DS = 0;
else
    T_N_DS = str2double(T_DS);
```

```
end

if strcmp(string_1,pH_DS)||isempty(pH_DS)
    pH_N_DS = 0;
else
    pH_N_DS = str2double(pH_DS);
end

if strcmp(string_1,pOH_DS)||isempty(pOH_DS)
    pOH_N_DS = 0;
else
    pOH_N_DS = str2double(pOH_DS);
end

OP_bar =
calculate(cnctN_NH4,cnctN_K,cnctN_Na,cnctN_Mg,cnctN_Ca
,cnctN_Sr,cnctN_Ba,cnctN_B,cnctN_CO3,cnctN_HCO3,cnctN_
NO3,cnctN_Cl,cnctN_F,cnctN_SO4,cnctN_SiO2, T_N_DS,
pH_N_DS, pOH_N_DS);

set(Tab3_TDS_text, 'string', num2str(TDS,'%0.2f'));
set(Tab3_OP_DS, 'string', num2str(OP_bar,'%0.2f'));
set(Tab3_Cations_text, 'string',
num2str(Cations,'%0.2f'));
set(Tab3_Anions_text, 'string',
num2str(Anions,'%0.2f'));
set(Tab3_Balance_text, 'string',
num2str(Balance_cat_an,'%0.2f'));

end


function OP_bar = calculate(DS_Ammonium, DS_Potassium,
DS_Sodium, DS_Magnesium, DS_Calcium,
DS_Strontium,DS_Barium,DS_Boron,DS_Carbonate,DS_Bicarb
onate,DS_Nitrate,DS_Chloride, DS_Fluoride, DS_Sulfate,
DS_Silica, Temperature, pH, pOH)

global Flag

global TDS Cations Anions Balance_cat_an Tbl_FS Tbl_DS

global VC_Ammonium VC_Potassium VC_Sodium VC_Magnesium
VC_Calcium VC_Strontium VC_Barium
global VC_Boron VC_Carbonate VC_Bicarbonate VC_Nitrate
VC_Chloride VC_Fluoride VC_Sulfate VC_Silica
```

```
global MW_Ammonium MW_Potassium MW_Sodium MW_Magnesium
MW_Calcium MW_Strontium MW_Barium  MW_Boron
global MW_Carbonate MW_Bicarbonate MW_Nitrate
MW_Chloride MW_Fluoride MW_Sulfate MW_Silica

Tbl_FS_Data_ret = get(Tbl_FS, 'Data');
Tbl_DS_Data_ret = get(Tbl_DS, 'Data');

TDS =  DS_Ammonium + DS_Potassium + DS_Sodium +
DS_Magnesium + DS_Calcium + DS_Strontium + DS_Barium +
DS_Boron + DS_Carbonate + DS_Bicarbonate + DS_Nitrate
+ DS_Chloride + DS_Fluoride + DS_Sulfate + DS_Silica;
%mg/L

if (DS_Ammonium == 0)||(VC_Ammonium == 0)

    EW_Ammonium = 0; MPL_Ammonium = 0; EPL_Ammonium =
0; MW_EPL_Ammonium = 0;  IS_Ammonium = 0;
mg_L_CaCo3_Ammonium = 0;
else
    EW_Ammonium = MW_Ammonium/abs(VC_Ammonium);
MPL_Ammonium = DS_Ammonium/(MW_Ammonium*1000);
    EPL_Ammonium =
DS_Ammonium*abs(VC_Ammonium)/(MW_Ammonium*1000);
MW_EPL_Ammonium = MW_Ammonium*EPL_Ammonium;
    IS_Ammonium = MPL_Ammonium*(abs(VC_Ammonium))^2;
mg_L_CaCo3_Ammonium = IS_Ammonium*100*1000;

end

if (DS_Potassium == 0)

    EW_Potassium = 0;  MPL_Potassium = 0;
EPL_Potassium = 0; MW_EPL_Potassium = 0;  IS_Potassium
= 0; mg_L_CaCo3_Potassium  = 0;

else
    EW_Potassium = MW_Potassium/abs(VC_Potassium);
MPL_Potassium = DS_Potassium/(MW_Potassium*1000);
    EPL_Potassium =
DS_Potassium*abs(VC_Potassium)/(MW_Potassium*1000);
MW_EPL_Potassium = MW_Potassium*EPL_Potassium;
    IS_Potassium =
MPL_Potassium*(abs(VC_Potassium))^2;
mg_L_CaCo3_Potassium = IS_Potassium*100*1000;
end
```

```
if (DS_Sodium == 0)

    EW_Sodium = 0;    MPL_Sodium = 0;    EPL_Sodium =
0;    MW_EPL_Sodium = 0;    IS_Sodium = 0;
mg_L_CaCo3_Sodium  = 0;

else
    EW_Sodium = MW_Sodium/abs(VC_Sodium);
MPL_Sodium = DS_Sodium/(MW_Sodium*1000);    EPL_Sodium
= DS_Sodium*abs(VC_Sodium)/(MW_Sodium*1000);
    MW_EPL_Sodium = MW_Sodium*EPL_Sodium;    IS_Sodium
= MPL_Sodium*(abs(VC_Sodium))^2; mg_L_CaCo3_Sodium =
IS_Sodium*100*1000;

end

if (DS_Magnesium == 0)

    EW_Magnesium = 0;    MPL_Magnesium = 0;
EPL_Magnesium = 0;    MW_EPL_Magnesium = 0;
IS_Magnesium = 0; mg_L_CaCo3_Magnesium = 0;

else

    EW_Magnesium = MW_Magnesium/abs(VC_Magnesium);
MPL_Magnesium = DS_Magnesium/(MW_Magnesium*1000);
    EPL_Magnesium =
DS_Magnesium*abs(VC_Magnesium)/(MW_Magnesium*1000);
MW_EPL_Magnesium = MW_Magnesium*EPL_Magnesium;
    IS_Magnesium =
MPL_Magnesium*(abs(VC_Magnesium))^2;
mg_L_CaCo3_Magnesium = IS_Magnesium*100*1000;

end

if (DS_Calcium == 0)

    EW_Calcium = 0;    MPL_Calcium = 0;    EPL_Calcium
= 0;    MW_EPL_Calcium = 0;    IS_Calcium = 0;
mg_L_CaCo3_Calcium = 0;
else
    EW_Calcium = MW_Calcium/abs(VC_Calcium);
MPL_Calcium = DS_Calcium/(MW_Calcium*1000);
EPL_Calcium =
DS_Calcium*abs(VC_Calcium)/(MW_Calcium*1000);
```

```
    MW_EPL_Calcium = MW_Magnesium*EPL_Calcium;
IS_Calcium = MPL_Calcium*(abs(VC_Calcium))^2;
mg_L_CaCo3_Calcium = IS_Calcium*100*1000;
end

if (DS_Strontium == 0)

    EW_Strontium = 0;    MPL_Strontium = 0;
EPL_Strontium = 0;    MW_EPL_Strontium = 0;
IS_Strontium = 0; mg_L_CaCo3_Strontium = 0;

else

    EW_Strontium = MW_Strontium/abs(VC_Strontium);
MPL_Strontium = DS_Strontium/(MW_Strontium*1000);
EPL_Strontium =
DS_Strontium*abs(VC_Strontium)/(MW_Strontium*1000);
    MW_EPL_Strontium = MW_Strontium*EPL_Strontium;
IS_Strontium = MPL_Strontium*(abs(VC_Strontium))^2;
mg_L_CaCo3_Strontium = IS_Strontium*100*1000;

end

if (DS_Barium == 0)

    EW_Barium = 0;    MPL_Barium = 0;    EPL_Barium =
0;    MW_EPL_Barium = 0;    IS_Barium = 0;
mg_L_CaCo3_Barium = 0;

else

    EW_Barium = MW_Barium/abs(VC_Barium);
MPL_Barium = DS_Barium/(MW_Barium*1000);    EPL_Barium
= DS_Barium*abs(VC_Barium)/(MW_Barium*1000);
    MW_EPL_Barium = MW_Barium*EPL_Barium;    IS_Barium
= MPL_Barium*(abs(VC_Barium))^2; mg_L_CaCo3_Barium =
IS_Barium*100*1000;

end

if (DS_Boron == 0)||(VC_Boron == 0)

    EW_Boron = 0;    MPL_Boron = 0;    EPL_Boron = 0;
MW_EPL_Boron = 0;    IS_Boron = 0; mg_L_CaCo3_Boron =
0;

else
```

```
    EW_Boron = MW_Boron/abs(VC_Boron);    MPL_Boron =
DS_Boron/(MW_Boron*1000);    EPL_Boron =
DS_Boron*abs(VC_Boron)/(MW_Boron*1000);
    MW_EPL_Boron = MW_Boron*EPL_Boron;    IS_Boron =
MPL_Boron*(abs(VC_Boron))^2; mg_L_CaCo3_Boron =
IS_Boron*100*1000;
end

if (DS_Carbonate == 0)

    EW_Carbonate = 0;    MPL_Carbonate  = 0;
EPL_Carbonate  = 0;    MW_EPL_Carbonate  = 0;
IS_Carbonate  = 0; mg_L_CaCo3_Carbonate = 0;

else

    EW_Carbonate = MW_Carbonate/abs(VC_Carbonate);
MPL_Carbonate = DS_Carbonate/(MW_Carbonate*1000);
EPL_Carbonate =
DS_Carbonate*abs(VC_Carbonate)/(MW_Carbonate*1000);
    MW_EPL_Carbonate = MW_Carbonate*EPL_Carbonate;
IS_Carbonate = MPL_Carbonate*(abs(VC_Carbonate))^2;
mg_L_CaCo3_Carbonate = IS_Carbonate*100*1000;

end

if (DS_Bicarbonate == 0)

    EW_Bicarbonate = 0;    MPL_Bicarbonate = 0;
EPL_Bicarbonate= 0;    MW_EPL_Bicarbonate = 0;
IS_Bicarbonate = 0; mg_L_CaCo3_Bicarbonate = 0;

else

    EW_Bicarbonate =
MW_Bicarbonate/abs(VC_Bicarbonate);    MPL_Bicarbonate
= DS_Bicarbonate/(MW_Bicarbonate*1000);
    EPL_Bicarbonate =
DS_Bicarbonate*abs(VC_Bicarbonate)/(MW_Bicarbonate*100
0);    MW_EPL_Bicarbonate =
MW_Bicarbonate*EPL_Bicarbonate
    IS_Bicarbonate =
MPL_Bicarbonate*(abs(VC_Bicarbonate))^2;
mg_L_CaCo3_Bicarbonate = IS_Bicarbonate*100*1000;

end
```

```
if (DS_Nitrate == 0)

    EW_Nitrate = 0;     MPL_Nitrate = 0;
EPL_Nitrate= 0;    MW_EPL_Nitrate = 0;    IS_Nitrate =
0; mg_L_CaCo3_Nitrate = 0;

else

    EW_Nitrate = MW_Nitrate /abs(VC_Nitrate);
MPL_Nitrate = DS_Nitrate/(MW_Nitrate*1000);
EPL_Nitrate =
DS_Nitrate*abs(VC_Nitrate)/(MW_Nitrate*1000);
    MW_EPL_Nitrate = MW_Nitrate*EPL_Nitrate;
IS_Nitrate = MPL_Nitrate*(abs(VC_Nitrate))^2;
mg_L_CaCo3_Nitrate = IS_Nitrate*100*1000;

end

if (DS_Chloride == 0)

    EW_Chloride = 0;    MPL_Chloride  = 0;
EPL_Chloride = 0;    MW_EPL_Chloride  = 0;
IS_Chloride  = 0; mg_L_CaCo3_Chloride = 0;

else

    EW_Chloride = MW_Chloride/abs(VC_Chloride);
MPL_Chloride = DS_Chloride/(MW_Chloride*1000);
    EPL_Chloride =
DS_Chloride*abs(VC_Chloride)/(MW_Chloride*1000);
MW_EPL_Chloride = MW_Chloride*EPL_Chloride;
    IS_Chloride = MPL_Chloride*(abs(VC_Chloride))^2;
mg_L_CaCo3_Chloride = IS_Chloride*100*1000;

end

if (DS_Fluoride == 0)

    EW_Fluoride = 0;    MPL_Fluoride  = 0;
EPL_Fluoride = 0;    MW_EPL_Fluoride  = 0;
IS_Fluoride  = 0; mg_L_CaCo3_Fluoride = 0;

else

    EW_Fluoride = MW_Fluoride/abs(VC_Fluoride);
MPL_Fluoride = DS_Fluoride/(MW_Fluoride*1000);
```

```
    EPL_Fluoride =
DS_Fluoride*abs(VC_Fluoride)/(MW_Fluoride*1000);
MW_EPL_Fluoride = MW_Fluoride*EPL_Fluoride;
    IS_Fluoride = MPL_Fluoride*(abs(VC_Fluoride))^2;
mg_L_CaCo3_Fluoride = IS_Fluoride*100*1000;
end

if (DS_Sulfate == 0)

    EW_Sulfate = 0;    MPL_Sulfate  = 0;
EPL_Sulfate = 0;    MW_EPL_Sulfate  = 0;    IS_Sulfate
= 0; mg_L_CaCo3_Sulfate = 0;

else

    EW_Sulfate = MW_Sulfate/abs(VC_Sulfate);
MPL_Sulfate = DS_Sulfate/(MW_Sulfate*1000);
    EPL_Sulfate =
DS_Sulfate*abs(VC_Sulfate)/(MW_Sulfate*1000);
MW_EPL_Sulfate = MW_Sulfate*EPL_Sulfate;
    IS_Sulfate = MPL_Sulfate*(abs(VC_Sulfate))^2;
mg_L_CaCo3_Sulfate = IS_Sulfate*100*1000;

end

if (DS_Silica == 0)||(VC_Silica == 0)

    EW_Silica = 0;    MPL_Silica  = 0;    EPL_Silica =
0;    MW_EPL_Silica  = 0;    IS_Silica  = 0;
mg_L_CaCo3_Silica = 0;

else

    EW_Silica = MW_Silica/abs(VC_Silica);
MPL_Silica = DS_Silica/(MW_Silica*1000);    EPL_Silica
= DS_Silica*abs(VC_Silica)/(MW_Silica*1000);
    MW_EPL_Silica = MW_Silica*EPL_Silica;    IS_Silica
= MPL_Silica*(abs(VC_Silica))^2; mg_L_CaCo3_Silica =
IS_Silica*100*1000;

end

MPL_pH = pH/1000; MPL_pOH = pOH/1000; EPL_pH =
pH/1000; EPL_pOH = pOH/1000; MW_EPL_pOH = EPL_pOH;
IS_pH = MPL_pH;  IS_pOH = MPL_pOH;

if (Flag == 1)
```

```
Tbl_FS_Data_ret{1,3} = num2str(EPL_Ammonium,'%0.2e');
Tbl_FS_Data_ret{2,3} = num2str(EPL_Potassium,'%0.2e');
Tbl_FS_Data_ret{3,3} = num2str(EPL_Sodium,'%0.2e');
Tbl_FS_Data_ret{4,3} = num2str(EPL_Magnesium,'%0.2e');
Tbl_FS_Data_ret{5,3} = num2str(EPL_Calcium,'%0.2e');
Tbl_FS_Data_ret{6,3} = num2str(EPL_Strontium,'%0.2e');
Tbl_FS_Data_ret{7,3} = num2str(EPL_Barium,'%0.2e');
Tbl_FS_Data_ret{8,3} = num2str(EPL_Boron,'%0.2e');
Tbl_FS_Data_ret{9,3} = num2str(EPL_Carbonate,'%0.2e');
Tbl_FS_Data_ret{10,3} =
num2str(EPL_Bicarbonate,'%0.2e');
Tbl_FS_Data_ret{11,3} = num2str(EPL_Nitrate,'%0.2e');
Tbl_FS_Data_ret{12,3} = num2str(EPL_Chloride,'%0.2e');
Tbl_FS_Data_ret{13,3} = num2str(EPL_Fluoride,'%0.2e');
Tbl_FS_Data_ret{14,3} = num2str(EPL_Sulfate,'%0.2e');
Tbl_FS_Data_ret{15,3} = num2str(EPL_Silica,'%0.2e');

Tbl_FS_Data_ret{1,4} = num2str(IS_Ammonium,'%0.2e');
Tbl_FS_Data_ret{2,4} = num2str(IS_Potassium,'%0.2e');
Tbl_FS_Data_ret{3,4} = num2str( IS_Sodium,'%0.2e');
Tbl_FS_Data_ret{4,4} = num2str(IS_Magnesium ,'%0.2e');
Tbl_FS_Data_ret{5,4} = num2str( IS_Calcium,'%0.2e');
Tbl_FS_Data_ret{6,4} = num2str(IS_Strontium,'%0.2e');
Tbl_FS_Data_ret{7,4} = num2str(IS_Barium,'%0.2e');
Tbl_FS_Data_ret{8,4} = num2str(IS_Boron,'%0.2e');
Tbl_FS_Data_ret{9,4} = num2str(IS_Carbonate,'%0.2e');
Tbl_FS_Data_ret{10,4} =
num2str(IS_Bicarbonate,'%0.2e'); Tbl_FS_Data_ret{11,4}
= num2str(IS_Nitrate,'%0.2e'); Tbl_FS_Data_ret{12,4} =
num2str(IS_Chloride,'%0.2e');
Tbl_FS_Data_ret{13,4} = num2str(IS_Fluoride,'%0.2e');
Tbl_FS_Data_ret{14,4} = num2str(IS_Sulfate,'%0.2e');
Tbl_FS_Data_ret{15,4} = num2str(IS_Silica,'%0.2e');

Tbl_FS_Data_ret{1,5} =
num2str(mg_L_CaCo3_Ammonium,'%0.2f');
Tbl_FS_Data_ret{2,5} =
num2str(mg_L_CaCo3_Potassium,'%0.2f');
Tbl_FS_Data_ret{3,5} =
num2str(mg_L_CaCo3_Sodium,'%0.2f');
Tbl_FS_Data_ret{4,5} =
num2str(mg_L_CaCo3_Magnesium,'%0.2f');
Tbl_FS_Data_ret{5,5} =
num2str(mg_L_CaCo3_Calcium,'%0.2f');
Tbl_FS_Data_ret{6,5} = num2str(mg_L_CaCo3_Strontium
```

```
,'%0.2f'); Tbl_FS_Data_ret{7,5} =
num2str(mg_L_CaCo3_Barium,'%0.2f');
Tbl_FS_Data_ret{8,5} =
num2str(mg_L_CaCo3_Boron,'%0.2f');
Tbl_FS_Data_ret{9,5} =
num2str(mg_L_CaCo3_Carbonate,'%0.2f');
Tbl_FS_Data_ret{10,5} =
num2str(mg_L_CaCo3_Bicarbonate,'%0.2f');
Tbl_FS_Data_ret{11,5} =
num2str(mg_L_CaCo3_Nitrate,'%0.2f');
Tbl_FS_Data_ret{12,5} =
num2str(mg_L_CaCo3_Chloride,'%0.2f');
Tbl_FS_Data_ret{13,5} =
num2str(mg_L_CaCo3_Fluoride,'%0.2f');
Tbl_FS_Data_ret{14,5} =
num2str(mg_L_CaCo3_Sulfate,'%0.2f');
Tbl_FS_Data_ret{15,5} =
num2str(mg_L_CaCo3_Silica,'%0.2f');

set(Tbl_FS,'data',Tbl_FS_Data_ret);

elseif (Flag == 2)

Tbl_DS_Data_ret{1,3} = num2str(EPL_Ammonium,'%0.2e');
Tbl_DS_Data_ret{2,3} = num2str(EPL_Potassium,'%0.2e');
Tbl_DS_Data_ret{3,3} = num2str(EPL_Sodium,'%0.2e');
Tbl_DS_Data_ret{4,3} = num2str(EPL_Magnesium,'%0.2e');
Tbl_DS_Data_ret{5,3} = num2str(EPL_Calcium,'%0.2e');
Tbl_DS_Data_ret{6,3} = num2str(EPL_Strontium,'%0.2e');
Tbl_DS_Data_ret{7,3} = num2str(EPL_Barium,'%0.2e');
Tbl_DS_Data_ret{8,3} = num2str(EPL_Boron,'%0.2e');
Tbl_DS_Data_ret{9,3} = num2str(EPL_Carbonate,'%0.2e');
Tbl_DS_Data_ret{10,3} =
num2str(EPL_Bicarbonate,'%0.2e');
Tbl_DS_Data_ret{11,3} = num2str(EPL_Nitrate,'%0.2e');
Tbl_DS_Data_ret{12,3} = num2str(EPL_Chloride,'%0.2e');
Tbl_DS_Data_ret{13,3} = num2str(EPL_Fluoride,'%0.2e');
Tbl_DS_Data_ret{14,3} = num2str(EPL_Sulfate,'%0.2e');
Tbl_DS_Data_ret{15,3} = num2str(EPL_Silica,'%0.2e');

Tbl_DS_Data_ret{1,4} = num2str(IS_Ammonium,'%0.2e');
Tbl_DS_Data_ret{2,4} = num2str(IS_Potassium,'%0.2e');
Tbl_DS_Data_ret{3,4} = num2str( IS_Sodium,'%0.2e');
Tbl_DS_Data_ret{4,4} = num2str(IS_Magnesium ,'%0.2e');
Tbl_DS_Data_ret{5,4} = num2str( IS_Calcium,'%0.2e');
Tbl_DS_Data_ret{6,4} = num2str(IS_Strontium,'%0.2e');
```

```
Tbl_DS_Data_ret{7,4} = num2str(IS_Barium,'%0.2e');
Tbl_DS_Data_ret{8,4} = num2str(IS_Boron,'%0.2e');
Tbl_DS_Data_ret{9,4} = num2str(IS_Carbonate,'%0.2e');
Tbl_DS_Data_ret{10,4} =
num2str(IS_Bicarbonate,'%0.2e'); Tbl_DS_Data_ret{11,4}
= num2str(IS_Nitrate,'%0.2e'); Tbl_DS_Data_ret{12,4} =
num2str(IS_Chloride,'%0.2e');
Tbl_DS_Data_ret{13,4} = num2str(IS_Fluoride,'%0.2e');
Tbl_DS_Data_ret{14,4} = num2str(IS_Sulfate,'%0.2e');
Tbl_DS_Data_ret{15,4} = num2str(IS_Silica,'%0.2e');

Tbl_DS_Data_ret{1,5} =
num2str(mg_L_CaCo3_Ammonium,'%0.2f');
Tbl_DS_Data_ret{2,5} =
num2str(mg_L_CaCo3_Potassium,'%0.2f');
Tbl_DS_Data_ret{3,5} =
num2str(mg_L_CaCo3_Sodium,'%0.2f');
Tbl_DS_Data_ret{4,5} =
num2str(mg_L_CaCo3_Magnesium,'%0.2f');
Tbl_DS_Data_ret{5,5} =
num2str(mg_L_CaCo3_Calcium,'%0.2f');
Tbl_DS_Data_ret{6,5} = num2str(mg_L_CaCo3_Strontium
,'%0.2f'); Tbl_DS_Data_ret{7,5} =
num2str(mg_L_CaCo3_Barium,'%0.2f');
Tbl_DS_Data_ret{8,5} =
num2str(mg_L_CaCo3_Boron,'%0.2f');
Tbl_DS_Data_ret{9,5} =
num2str(mg_L_CaCo3_Carbonate,'%0.2f');
Tbl_DS_Data_ret{10,5} =
num2str(mg_L_CaCo3_Bicarbonate,'%0.2f');
Tbl_DS_Data_ret{11,5} =
num2str(mg_L_CaCo3_Nitrate,'%0.2f');
Tbl_DS_Data_ret{12,5} =
num2str(mg_L_CaCo3_Chloride,'%0.2f');
Tbl_DS_Data_ret{13,5} =
num2str(mg_L_CaCo3_Fluoride,'%0.2f');
Tbl_DS_Data_ret{14,5} =
num2str(mg_L_CaCo3_Sulfate,'%0.2f');
Tbl_DS_Data_ret{15,5} =
num2str(mg_L_CaCo3_Silica,'%0.2f');

set(Tbl_DS,'data',Tbl_DS_Data_ret);


end
```

```
%==============================Charge
Balance==============================%

Cations = MPL_Ammonium + MPL_Potassium + MPL_Sodium +
MPL_Magnesium + MPL_Calcium + MPL_Strontium +
MPL_Barium + MPL_Boron + MPL_pH;
Anions = EPL_Carbonate + EPL_Bicarbonate + EPL_Nitrate
+ EPL_Chloride + EPL_Fluoride + EPL_Sulfate + EPL_pOH;

Balance_cat_an = Cations/Anions;

Total_EPL = EPL_Ammonium + EPL_Potassium + EPL_Sodium
+ EPL_Magnesium + EPL_Calcium + EPL_Strontium +
EPL_Barium + EPL_Boron + EPL_Carbonate +
EPL_Bicarbonate + EPL_Nitrate + EPL_Chloride +
EPL_Fluoride + EPL_Sulfate + EPL_Silica;

Avg_EPL = (MW_EPL_Ammonium + MW_EPL_Potassium +
MW_EPL_Sodium + MW_EPL_Magnesium + MW_EPL_Calcium +
MW_EPL_Strontium + MW_EPL_Barium + MW_EPL_Boron +
MW_EPL_Carbonate + MW_EPL_Bicarbonate + MW_EPL_Nitrate
+ MW_EPL_Chloride + MW_EPL_Fluoride + MW_EPL_Sulfate +
MW_EPL_Silica)/Total_EPL;

Total_IS = 0.5*(IS_Ammonium + IS_Potassium + IS_Sodium
+ IS_Magnesium + IS_Calcium + IS_Strontium + IS_Barium
+ IS_Boron + IS_Carbonate + IS_Bicarbonate +
IS_Nitrate + IS_Chloride + IS_Fluoride + IS_Sulfate +
IS_Silica + IS_pH + IS_pOH);

Avg_MW = (DS_Ammonium + DS_Potassium + DS_Sodium +
DS_Magnesium + DS_Calcium + DS_Strontium + DS_Barium +
DS_Boron + DS_Carbonate + DS_Bicarbonate + DS_Nitrate
+ DS_Chloride + DS_Fluoride + DS_Sulfate +
DS_Silica)/((EPL_Ammonium + EPL_Potassium + EPL_Sodium
+ EPL_Magnesium + EPL_Calcium + EPL_Strontium +
EPL_Barium + EPL_Boron + EPL_Carbonate +
EPL_Bicarbonate + EPL_Nitrate + EPL_Chloride +
EPL_Fluoride + EPL_Sulfate + EPL_Silica)*1000);

Total_MPL = MPL_Ammonium + MPL_Potassium + MPL_Sodium
+ MPL_Magnesium + MPL_Calcium + MPL_Strontium +
MPL_Barium + MPL_Boron + MPL_Carbonate +
MPL_Bicarbonate + MPL_Nitrate + EPL_Chloride +
MPL_Fluoride + MPL_Sulfate + MPL_Silica;

OP_psi = 1.19*(273+Temperature)*Total_MPL;
```

```
OP_bar = OP_psi*6.8949*0.01;

C_M = (TDS/1000)/58.443;

%OP_bar = (3.76*C_M^2+42.52*C_M+0.41)*1.01325;

end

function El_Model_Callback(~,~,Tab4_popup_ElModel)

global El_Model Tab4_edit_MemType Tab4_edit_ModType
Tab4_edit_SpacerType El_model_gui W_Perm S_Perm S
sp_th Area

El_Model = get(Tab4_popup_ElModel, 'value');

if (El_Model  == 1)

    set(Tab4_edit_MemType, 'string', '');
    set(Tab4_edit_ModType, 'string', '');
    set(Tab4_edit_SpacerType, 'string', '');
    El_model_gui = 'Toray 8040';
    W_Perm = 0; %(L/m2-hr-bar)
    S_Perm = 0; %(L/m2-hr)
    S      = 0; %(mm)
    sp_th = 0; %(mm)
    Area  = 0;  %(m2)

elseif(El_Model  == 2)

    set(Tab4_edit_MemType, 'string', 'TFC');
    set(Tab4_edit_ModType, 'string', 'Spiral wound');
    set(Tab4_edit_SpacerType, 'string', 'Diamond');
    El_model_gui = 'Toray 8040';
    W_Perm = 5.54; %(L/m2-hr-bar)
    S_Perm = 2.26; %(L/m2-hr)
    S      = 0.46; %(mm)
    sp_th = 1.19; %(mm)
    Area  = 15.3;  %(m2)

elseif(El_Model  == 3)

    set(Tab4_edit_MemType, 'string', 'TFC');
    set(Tab4_edit_ModType, 'string', 'Spiral wound');
    set(Tab4_edit_SpacerType, 'string', 'Diamond');
    El_model_gui = 'Toray 4040';
```

```
    W_Perm = 4.54; %(L/m2-hr-bar)
    S_Perm = 1.26; %(L/m2-hr)
    S      = 0.36; %(mm)
    sp_th = 2.19; %(mm)
    Area  = 14.3;  %(m2)

 elseif(El_Model  == 4)

    set(Tab4_edit_MemType, 'string', 'CTA');
    set(Tab4_edit_ModType, 'string', 'Flat plate');
    set(Tab4_edit_SpacerType, 'string', 'Corrugated');
    El_model_gui = 'HTI CTA';
    W_Perm = 2.54; %(L/m2-hr-bar)
    S_Perm = 0.26; %(L/m2-hr)
    S      = 0.46; %(mm)
    sp_th = 2.19; %(mm)
    Area  = 14.3;  %(m2)


  elseif(El_Model  == 5)

    set(Tab4_edit_MemType, 'string', 'CTA');
    set(Tab4_edit_ModType, 'string', 'Hollow Fibre');
    set(Tab4_edit_SpacerType, 'string', '');
    El_model_gui = 'HTI CTA';
    W_Perm = 0.7; %(L/m2-hr-bar)
    S_Perm = 0.53; %(L/m2-hr)
    S      = 0.46; %(mm)
    Area  = 31.5;  %(m2)

  elseif(El_Model  == 6)

    set(Tab4_edit_MemType, 'string', 'TFC');
    set(Tab4_edit_ModType, 'string', 'Plate & Frame');
    set(Tab4_edit_SpacerType, 'string', 'Fishnet');
    El_model_gui = 'Porifera TFC';
    W_Perm = 2.2; %(L/m2-hr-bar)
    S_Perm = 0.49; %(L/m2-hr)
    S      = 0.215; %(mm)
    Area  = 7;  %(m2)

end

end

function Flow_Config_Callback(~,~,Tab4_popup_FlCon)
```

```
global Bin_Location Tab5_pan_Sys m_PanelWidth
m_PanelHeight hImageAxes

Flow_Con = get(Tab4_popup_FlCon, 'value');

if (Flow_Con == 1)

     if ishandle(hImageAxes)

            delete(hImageAxes);      % Delete the
previous image
     end



elseif(Flow_Con == 2)

    if ishandle(hImageAxes)

        delete(hImageAxes);      % Delete the previous
image

        PicFilePath =
strcat(Bin_Location,'\CO_Config.png');
        InitPicRGB = imread(PicFilePath);
        hImageAxes = axes('Parent',
Tab5_pan_Sys,'Units', 'pixels','Position',
[0.02*m_PanelWidth 0.01*m_PanelHeight
0.99*m_PanelWidth 0.19*m_PanelHeight], 'visible',
'off');
        imshow(InitPicRGB,'Parent', hImageAxes);

    else

        PicFilePath =
strcat(Bin_Location,'\CO_Config.png');
        InitPicRGB = imread(PicFilePath);
        hImageAxes = axes('Parent',
Tab5_pan_Sys,'Units', 'pixels','Position',
[0.02*m_PanelWidth 0.01*m_PanelHeight
0.99*m_PanelWidth 0.19*m_PanelHeight], 'visible',
'off');
        imshow(InitPicRGB,'Parent', hImageAxes);

   end
```

```
elseif(Flow_Con  == 3)


    if ishandle(hImageAxes)

        delete(hImageAxes);     % Delete the previous
image

        PicFilePath =
strcat(Bin_Location,'\Counter_Config.png');
        InitPicRGB = imread(PicFilePath);
        hImageAxes = axes('Parent',
Tab5_pan_Sys,'Units', 'pixels','Position',
[0.02*m_PanelWidth 0.01*m_PanelHeight
0.99*m_PanelWidth 0.19*m_PanelHeight], 'visible',
'off');
        imshow(InitPicRGB,'Parent', hImageAxes);

    else

        PicFilePath =
strcat(Bin_Location,'\Counter_Config.png');
        InitPicRGB = imread(PicFilePath);
        hImageAxes = axes('Parent',
Tab5_pan_Sys,'Units', 'pixels','Position',
[0.02*m_PanelWidth 0.01*m_PanelHeight
0.99*m_PanelWidth 0.19*m_PanelHeight], 'visible',
'off');
        imshow(InitPicRGB,'Parent', hImageAxes);

   end

end


end

function Opt_Callback(~,~,Tab4_button_Opt)

global Tab2_FS_disp Tab3_DS_disp Tab3_OP_DS Tab2_OP_FS
global Tab4_edit_Rec Tab4_edit_Rec_Flex Tab4_edit_Conc
Tab4_edit_Conc_Flex
global Tab4_Q_D_in_i Tab4_Q_D_in_f Tab4_del_Q_D_in
Tab4_Q_F_in_i Tab4_C_D_in Tab4_C_F_in
global Tab4_status_disp
global Tab4_opt_Q_FS_in Tab4_opt_Q_DS_in Tab4_opt_NE
Tab4_opt_Rec Tab4_opt_C_DS Tab4_opt_OPR
```

```
global opt_Q_FS_gui opt_Q_DS_gui Opt_NE_gui
Opt_Rec_gui Opt_C_DS_gui Opt_OPR_gui
global El_model_gui Tab4_edit_MemType
Tab4_edit_ModType Tab4_edit_SpacerType W_Perm S_Perm S
sp_th Area
global Tab5_Rec Tab5_C_D_f Tab5_El_Mod Tab5_Mod_Type
Tab5_Mem_Type Tab5_A Tab5_B Tab5_S Tab5_Area
Tab5_Sp_Type Tab5_Sp_thickness
global Tab5_NE Tab5_FS_Type Tab5_FS_Flr Tab5_FS_C
Tab5_FS_OP Tab5_DS_Type Tab5_DS_Flr Tab5_DS_C
Tab5_DS_OP

set(Tab4_status_disp , 'string', 'Optimization in
progress');
drawnow

Rec_D = str2double(get(Tab4_edit_Rec, 'string'));
Rec_flex = str2double(get(Tab4_edit_Rec_Flex,
'string')); C_D_f_Design =
str2double(get(Tab4_edit_Conc, 'string'));
C_D_f_Design_flex =
str2double(get(Tab4_edit_Conc_Flex, 'string'));

Q_F_in_Ui = str2double(get(Tab4_Q_F_in_i, 'string'));
Q_D_in_Ui = str2double(get(Tab4_Q_D_in_i, 'string'));
Q_D_in_Uf = str2double(get(Tab4_Q_D_in_f, 'string'));
del_Q_D_in = str2double(get(Tab4_del_Q_D_in,
'string'));
C_D_U_in = str2double(get(Tab4_C_D_in, 'string'));
C_F_U_in = str2double(get(Tab4_C_F_in, 'string'));

Opt_OPR_gui = FO_Opt(Rec_D, Rec_flex, C_D_f_Design,
C_D_f_Design_flex, Q_F_in_Ui, Q_D_in_Ui, Q_D_in_Uf,
del_Q_D_in, C_D_U_in, C_F_U_in);

set(Tab4_status_disp , 'string', 'Optimization is
done');
set(Tab4_opt_Q_FS_in, 'string',
num2str(opt_Q_FS_gui,'%0.2f'));
set(Tab4_opt_Q_DS_in, 'string',
num2str(opt_Q_DS_gui,'%0.2f'));
set(Tab4_opt_NE, 'string', num2str(Opt_NE_gui));
set(Tab4_opt_Rec, 'string',
num2str(Opt_Rec_gui,'%0.2f'));
set(Tab4_opt_C_DS, 'string',
num2str(Opt_C_DS_gui,'%0.2f'));
```

```
set(Tab4_opt_OPR, 'string',
num2str(Opt_OPR_gui,'%0.2f'));

Mod_type = get(Tab4_edit_ModType, 'string');
Mem_type = get(Tab4_edit_MemType, 'string');
Spacer_type = get(Tab4_edit_SpacerType, 'string');
FS_type = get(Tab2_FS_disp, 'string');
FS_OP = get(Tab2_OP_FS, 'string');
DS_type = get(Tab3_DS_disp, 'string');
DS_OP = get(Tab3_OP_DS, 'string');

set(Tab5_Rec, 'string', num2str(Rec_D, '%0.2f'));
set(Tab5_C_D_f, 'string', num2str(C_D_f_Design,
'%0.2f'));
set(Tab5_El_Mod, 'string', El_model_gui);
set(Tab5_Mod_Type, 'string', Mod_type);
set(Tab5_Mem_Type, 'string', Mem_type);
set(Tab5_A, 'string', num2str(W_Perm, '%0.2f'));
set(Tab5_B, 'string', num2str(S_Perm, '%0.2f'));
set(Tab5_S, 'string', num2str( S, '%0.2f'));
set(Tab5_Area, 'string', num2str(Area, '%0.2f'));
set(Tab5_Sp_Type, 'string', Spacer_type);
set(Tab5_Sp_thickness, 'string', num2str(sp_th,
'%0.2f'));
set(Tab5_NE, 'string', num2str(Opt_NE_gui));
set(Tab5_FS_Type, 'string', FS_type);
set(Tab5_FS_Flr, 'string', num2str(opt_Q_FS_gui,
'%0.2f'));
set(Tab5_FS_C, 'string', num2str(C_F_U_in, '%0.2f'));
set(Tab5_FS_OP, 'string', num2str(FS_OP, '%0.2f'));
set(Tab5_DS_Type, 'string', DS_type);
set(Tab5_DS_Flr, 'string', num2str(opt_Q_DS_gui,
'%0.2f'));
set(Tab5_DS_C, 'string', num2str(C_D_U_in, '%0.2f'));
set(Tab5_DS_OP, 'string', num2str(DS_OP, '%0.2f'));

end

function Sim_Callback(~,~,Tab6_button_Sim)

global Tab4_Q_D_in_i Tab5_Area Opt_NE_gui
Tab5_Prod_Cap Tab6_NPV Tab6_TNE Tab6_TArea Tab6_Q_DS
Tab6_Q_P Tab6_C_D_f
global Prod_Per_PV Opt_C_DS_gui

Q_D_in = str2double(get(Tab4_Q_D_in_i, 'string'));
```

```
Mem_Area = str2double(get(Tab5_Area, 'string'));
Prod_Cap_Req = str2double(get(Tab5_Prod_Cap,
'string'));

NPV =
round(((Prod_Cap_Req*1000)/(24*60))/Prod_Per_PV);
TNE = NPV*Opt_NE_gui;
TArea = round(TNE*Mem_Area);
T_Q_DS = round(((NPV*Prod_Per_PV)/1000)*60*24);
T_Q_P = round(((NPV*(Prod_Per_PV -
Q_D_in))/1000)*24*60);
C_D_f = Opt_C_DS_gui;

set(Tab6_NPV, 'string', num2str(NPV));
set(Tab6_TNE, 'string', num2str(TNE));
set(Tab6_TArea, 'string', num2str(TArea));
set(Tab6_Q_DS, 'string', num2str(T_Q_DS));
set(Tab6_Q_P, 'string', num2str(T_Q_P));
set(Tab6_C_D_f, 'string', num2str(C_D_f, '%0.2f'));

El_dist = Sim_Drawing(Opt_NE_gui);

end

function Prod_Cost =
Tech_Eco_Callback(~,~,Tab7_button_Analyze)

global Tab7_Net_Capex Tab7_Net_Opex Tab7_Net_En_Cons
Tab7_Unit_Prod_Cost Tab7_Unit_Opex Tab7_Unit_Capex
Tab7_Sp_En_Cons Tab7_Loan_ROI
global Tab7_Loan_Capex Tab7_Loan_Opex Tab7_Op_Pwr_Cost
Tab7_Op_FS_Pump_Rep Tab7_Op_DS_Pump_Rep Tab7_Op_El_Rep
Tab7_Op_PV_Rep Tab7_Op_Emp_Exp
global Tab7_Cap_El_Price Tab7_Cap_PV_Price
Tab7_Cap_DS_Pmp_Cap Tab7_Cap_DS_Pmp_Price
Tab7_Cap_Land_Cost Tab7_Cap_Inf_Cost Tab7_FS_Pmp_Cap
global Tab7_Cap_FS_Pmp_Price Tab7_Plant_Life
Tab7_El_Life Tab7_PV_Life Tab7_Pmp_Life Tab7_Op_Time

%==================Invesment Options
(Input)==============================%

Loan_Capex = str2double(get(Tab7_Loan_Capex,
'string'));
Loan_Opex = str2double(get(Tab7_Loan_Opex, 'string'));
Loan_ROI = str2double(get(Tab7_Loan_ROI, 'string'));
```

```
%===================Operation Time
(Input)==================================%

Plant_Life = str2double(get(Tab7_Plant_Life,
'string'));
El_Life = str2double(get(Tab7_El_Life, 'string'));
PV_Life = str2double(get(Tab7_PV_Life, 'string'));
Pmp_Life = str2double(get(Tab7_Pmp_Life, 'string'));
Op_Time = str2double(get(Tab7_Op_Time, 'string'));

%===================Capex Cost
(Input)======================================%

Cap_El_Price = str2double(get(Tab7_Cap_El_Price,
'string'));
Cap_PV_Price = str2double(get(Tab7_Cap_PV_Price,
'string'));
Cap_DS_Pmp_Cap = str2double(get(Tab7_Cap_DS_Pmp_Cap,
'string'));
Cap_DS_Pmp_Price =
str2double(get(Tab7_Cap_DS_Pmp_Price, 'string'));
FS_Pmp_Cap = str2double(get(Tab7_FS_Pmp_Cap,
'string'));
Cap_FS_Pmp_Price =
str2double(get(Tab7_Cap_FS_Pmp_Price, 'string'));
Cap_Land_Cost = str2double(get(Tab7_Cap_Land_Cost,
'string'));
Cap_Inf_Cost = str2double(get(Tab7_Cap_Inf_Cost,
'string'));

%====================Opex Cost
(Input)======================================%

Op_Pwr_Cost = str2double(get(Tab7_Op_Pwr_Cost,
'string'));
Op_FS_Pump_Rep = str2double(get(Tab7_Op_FS_Pump_Rep,
'string'));
Op_DS_Pump_Rep = str2double(get(Tab7_Op_DS_Pump_Rep,
'string'));
Op_El_Rep = str2double(get(Tab7_Op_El_Rep, 'string'));
Op_PV_Rep = str2double(get(Tab7_Op_PV_Rep, 'string'));
Op_Emp_Exp = str2double(get(Tab7_Op_Emp_Exp,
'string'));

%====================TechnoEconomic(Output)===========
=========================%
```

```
set(Tab7_Net_Capex, 'string', num2str(Net_Capex,
'%0.2f'));
set(Tab7_Net_Opex, 'string', num2str(Net_Opex,
'%0.2f'));
set(Tab7_Net_En_Cons, 'string', num2str(Net_En_Cons,
'%0.2f'));
set(Tab7_Unit_Opex, 'string', num2str(Unit_Opex,
'%0.2f'));
set(Tab7_Unit_Capex, 'string', num2str(Unit_Capex,
'%0.2f'));
set(Tab7_Sp_En_Cons, 'string', num2str(Sp_En_Cons,
'%0.2f'));
set(Tab7_Unit_Prod_Cost, 'string',
num2str(Unit_Prod_Cost, '%0.2f'));



El_dist = Sim_Drawing(Opt_NE_gui);


end


function El_dist = Sim_Drawing(N_El)

global m_PanelHeight m_PanelWidth Tab_6_Panel_Sim_Out
Q_D_in_E Q_F_in_E Q_D_ind C_F_in_E C_D_in_E Recovery_E
Avg_WF_LMH
global Col_H1 Col_H2 Col_H3 Col_H4 Col_H5 Col_H6
Col_H7 Col_H8
global Font_tab_title Font_title Font_large
Font_medium Font_small

El_height = (0.5*m_PanelHeight)/2;
text_height = 0.04*m_PanelHeight;

El_dist = (0.95*m_PanelWidth)/((N_El+1)+N_El);
El_width = El_dist;
El_elv = (0.5*m_PanelHeight)/4;
FS_ax_elv = El_elv + El_height;
FS_ax_width = El_dist + El_width/2;
FS_ax_V_width = El_width;
DS_FR_width = El_dist/2;
DS_C_width = El_dist/2;
FS_FR_width = El_dist/2;
FS_C_width = El_dist/2;
```

```
Rec_el_width = El_width/2;
Flx_el_width = El_width/2;
FS_ax_V_height = El_elv/2;

FS_ax_V_elv = FS_ax_elv;
DS_FR_elv = El_elv + (El_height/2) + (El_elv/4) -
text_height/2;
DS_C_elv = El_elv + (El_height/2) - (El_elv/4) -
text_height/2;
FS_FR_elv = El_elv + El_height + 3*El_elv/4 -
text_height/2;
FS_C_elv = El_elv + El_height + El_elv/4 -
text_height/2;
Rec_el_elv = El_elv + 3*El_height/4 - text_height/2;
Flx_el_elv = El_elv + El_height/4 - text_height/2;

El_dist_i = El_dist;
DS_Line_i = 0;
FS_Line_i = 0;
FS_Line_V_i = El_dist;
DS_FR_i = El_dist/4;
DS_C_i = El_dist/4;
FS_FR_i = El_dist/4;
Rec_el_i = El_dist + El_width/4;
Flx_el_i = El_dist + El_width/4;
FS_C_i = El_dist/4;

for N = 1:1:N_El+1

ax_DS = axes('Parent', Tab_6_Panel_Sim_Out,'Units',
'pixels','Position', [DS_Line_i El_elv El_width
El_height],'visible', 'off');
line([0,1],[0.5,0.5],'parent',ax_DS, 'color',
Col_H5,'LineWidth',3);

% create an edit Text to display "DS FR"
Tab_6_DS_FR = uicontrol('Parent',
Tab_6_Panel_Sim_Out,'Style', 'edit', 'Enable', 'off',
'Position', [DS_FR_i DS_FR_elv DS_FR_width
text_height],'Backgroundcolor', Col_H3,...
'string', '','HorizontalAlignment',
'center','FontName', 'arial','FontWeight',
'normal','FontSize', Font_large);

DS_FR = Q_D_in_E(N, Q_D_ind);
set(Tab_6_DS_FR, 'string', num2str(DS_FR, '%0.2f'));
```

```
% create an edit Text to display "DS C"
Tab_6_DS_C = uicontrol('Parent',
Tab_6_Panel_Sim_Out,'Style', 'edit', 'Enable',
'off','Position', [DS_C_i DS_C_elv DS_C_width
text_height],'Backgroundcolor', Col_H3,...
'string', '','HorizontalAlignment',
'center','FontName', 'arial','FontWeight',
'normal','FontSize', Font_large);

DS_C = C_D_in_E(N, Q_D_ind);
set(Tab_6_DS_C, 'string', num2str(DS_C, '%0.2f'));

% create an edit Text for "FS FR"
Tab_6_FS_FR = uicontrol('Parent',
Tab_6_Panel_Sim_Out,'Style', 'edit', 'Enable',
'off','Position', [FS_FR_i FS_FR_elv FS_FR_width
text_height],'Backgroundcolor', Col_H3,...
'string', '','HorizontalAlignment',
'center','FontName', 'arial','FontWeight',
'normal','FontSize', Font_large);

FS_FR = Q_F_in_E(N, Q_D_ind);
set(Tab_6_FS_FR, 'string', num2str(FS_FR, '%0.2f'));

% create an edit Text for "FS C"
Tab_6_FS_C = uicontrol('Parent',
Tab_6_Panel_Sim_Out,'Style', 'edit', 'Enable',
'off','Position', [FS_C_i FS_C_elv FS_C_width
text_height],'Backgroundcolor', Col_H3,...
'string', '','HorizontalAlignment',
'center','FontName', 'arial','FontWeight',
'normal','FontSize', Font_large);

FS_C = C_F_in_E(N, Q_D_ind);
set(Tab_6_FS_C, 'string', num2str(FS_C, '%0.2f'));

ax_FS = axes('Parent', Tab_6_Panel_Sim_Out,'Units',
'pixels','Position', [FS_Line_i FS_ax_elv FS_ax_width
El_elv],'visible', 'off');
line([0,1],[0,0],'parent',ax_FS,'color',
Col_H7,'LineWidth',3)


if (N <= N_El)

    ax_el = axes('Parent',
Tab_6_Panel_Sim_Out,'Units', 'pixels','Position',
```

```
[El_dist_i El_elv El_width El_height],'visible',
'off');
    rectangle('Position', [0 0 1 1],
'FaceColor',Col_H8,'EdgeColor','b','LineWidth',3);

    % create an edit Text for "Recovery"
    Tab_6_Rec_E = uicontrol('Parent',
Tab_6_Panel_Sim_Out,'Style', 'edit', 'Enable',
'off','Position', [Rec_el_i Rec_el_elv Rec_el_width
text_height],'Backgroundcolor', Col_H3,...
    'string', '','HorizontalAlignment',
'center','FontName', 'arial','FontWeight',
'normal','FontSize', Font_large);

    Rec_E = Recovery_E(N+1, Q_D_ind);
    set(Tab_6_Rec_E, 'string', num2str(Rec_E,
'%0.2f'));

    % create an edit Text for "Flux"
    Tab_6_AWF_E = uicontrol('Parent',
Tab_6_Panel_Sim_Out,'Style', 'edit', 'Enable',
'off','Position', [Flx_el_i Flx_el_elv Flx_el_width
text_height],'Backgroundcolor', Col_H3,...
    'string', '','HorizontalAlignment',
'center','FontName', 'arial','FontWeight',
'normal','FontSize', Font_large);

    AWF_E = Avg_WF_LMH(N+1, Q_D_ind);
    set(Tab_6_AWF_E, 'string', num2str(AWF_E,
'%0.2f'));

    FS_ax_V1 = axes('Parent',
Tab_6_Panel_Sim_Out,'Units', 'pixels','Position',
[FS_Line_V_i FS_ax_V_elv FS_ax_V_width FS_ax_V_height
],'visible', 'off');
    line([0,0],[1,0],'parent',FS_ax_V1,'color',
Col_H7,'LineWidth',3);

    FS_ax_V2 = axes('Parent',
Tab_6_Panel_Sim_Out,'Units', 'pixels','Position',
[FS_Line_V_i FS_ax_V_height FS_ax_V_width
FS_ax_V_height ],'visible', 'off');
    line([0,0],[1,0],'parent',FS_ax_V2,'color',
Col_H7,'LineWidth',3);

end
```

```
El_dist_i = El_dist_i + El_dist + El_width;
DS_Line_i = DS_Line_i + El_dist + El_width;
DS_FR_i = DS_FR_i + El_dist + El_width;
DS_C_i = DS_C_i + El_dist + El_width;
Rec_el_i = Rec_el_i + El_dist + El_width;
Flx_el_i = Flx_el_i + El_dist + El_width;
FS_FR_i = FS_FR_i + El_dist + El_width;
FS_C_i = FS_C_i + El_dist + El_width;

if (N == 1)||(rem(N,2)==1)

    FS_ax_elv = 0;
    FS_FR_elv = 3*El_elv/4 - text_height/2;
    FS_C_elv = El_elv/4 - text_height/2;
else

    FS_ax_elv = El_elv + El_height;
    FS_FR_elv = El_elv + El_height + 3*El_elv/4 -
text_height/2;
    FS_C_elv = El_elv + El_height + El_elv/4 -
text_height/2;

end

if (N == 1)

    FS_Line_i = FS_Line_i + El_dist + El_width/2;

else

    FS_Line_i = FS_Line_i + El_dist + El_width;

end

FS_Line_V_i = FS_Line_V_i + + El_dist + El_width;

FS_ax_width = El_dist + El_width;

end


end

function Opt_OPR_gui = FO_Opt(Rec_D, Rec_flex,
C_D_f_Design, C_D_f_Design_flex, Q_F_in_Ui, Q_D_in_Ui,
Q_D_in_Uf, del_Q_D_in, C_D_U_in, C_F_U_in)
```

```
global El_Model A_eff_SW opt_Q_FS_gui opt_Q_DS_gui
Opt_NE_gui Opt_Rec_gui Opt_C_DS_gui Prod_Per_PV
Q_D_in_E Q_F_in_E C_F_in_E C_D_in_E Recovery_E
Avg_WF_LMH Q_D_ind
global Q_D_int_gui C_F_Avg C_D_Avg ttl_perm A_eff_HF
A_eff_PF Summary_excl

%%                        Process Constants

Q_F_inlet = Q_F_in_Ui;          %Feed solution inlet
flow rate (lpm)
C_F_inlet = C_F_U_in;        %Feed Solution
concentration (M)
C_D_inlet = C_D_U_in;         %Draw solution to feed
solution concentration ratio

Q_D_V_i = Q_D_in_Ui;              %Initial Draw solution
inlet flow rate value of range
Q_D_V_f = Q_D_in_Uf;             %Final Draw solution
inlet flow rate value of range
del_Q_D = del_Q_D_in;            %Step change of Draw
solution inlet flow rate value

%%                        Design Constraints

C_D_F = C_D_f_Design;% Required Final Draw Solution
Concentration(M)
Recovery_Final = (Rec_D/100); % Maximum Recovery
del_C_D_E = 0.001; % Minimum Concentration Difference
Between Two Elements (M)

RF_Cut_Off = 1- (Rec_flex/100);
CNF_Cut_Off = 1 - (C_D_f_Design_flex/100);

%%                        Process Inputs

flag_D = 1; FF = 1; Prog_ind = 1;

while(flag_D == 1)

progress = ProgBar_Drawing(Prog_ind);

Q_F_in_i = Q_F_inlet; %(LPM)
Q_D_in_i = Q_D_V_i; %(LPM)

C_F_in_i =  C_F_inlet; % (mole/L)
C_D_in_i = C_D_inlet; %(LPM)
```

```
E = 1;

Q_F_in_E(E,FF) = Q_F_in_i;
Q_D_in_E(E,FF) = Q_D_in_i;

C_F_in_E(E,FF) = C_F_in_i;
C_D_in_E(E,FF) = C_D_in_i;

Flag_Sim = 1;

if(C_D_in_i <= C_D_F)

    Flag_Sim = 0;

end

Recovery_E(E,FF)= 0;

while (Flag_Sim == 1)


%%              Element Initialization

if((El_Model == 1)||(El_Model == 2)||(El_Model == 3))

    Q_F_int_gui = J_w_FO_SW(Q_F_in_i, Q_D_in_i,
C_F_in_i, C_D_in_i);

elseif ((El_Model == 4)||(El_Model == 5))

    Q_F_int_gui = J_w_FO_HF(Q_F_in_i, Q_D_in_i,
C_F_in_i, C_D_in_i);


elseif(El_Model == 6)

    Q_F_int_gui = J_w_FO_PF(Q_F_in_i, Q_D_in_i,
C_F_in_i, C_D_in_i);

end

Q_F_in_i = Q_F_int_gui*1000*60;
Q_D_in_i = Q_D_int_gui*1000*60;

C_F_in_i = C_F_Avg;
C_D_in_i = C_D_Avg;
```

```
E = E+1;

EA((E-1),FF) = E-1;

Q_F_in_E(E, FF) = Q_F_in_i;
Q_D_in_E(E, FF) = Q_D_in_i;

C_F_in_E(E, FF) = C_F_in_i;
C_D_in_E(E, FF) = C_D_in_i;

del_C_D_i = C_D_in_E((E-1), FF) - C_D_in_E(E, FF);

Recovery_E(E, FF) = ((Q_F_in_E(1, FF)-Q_F_in_E(E,
FF))/Q_F_in_E(1, FF))*100;

if((El_Model == 1)||(El_Model == 2)||(El_Model == 3))

    A_eff = A_eff_SW;

elseif ((El_Model == 4)||(El_Model == 5))

    A_eff = A_eff_HF;

elseif (El_Model == 6)

    A_eff = A_eff_PF;

end

Avg_WF_LMH(E,FF) =  (ttl_perm)/A_eff;

RF_C = (Recovery_E(E, FF)/100)/Recovery_Final;

if (RF_C >= 1)

    RF(E, FF) = 1;

elseif (RF_C < 1)&& (RF_C >= RF_Cut_Off )

    RF(E, FF) = RF_C;

else

    RF(E, FF) = 0;

end

CNF_C = (C_D_F/C_D_in_E(E, FF));
```

```
if(CNF_C >= 1)

    CNF(E, FF) = 1;

elseif (CNF_C < 1)&&(CNF_C >= CNF_Cut_Off)

    CNF(E, FF) = CNF_C;

else

   CNF(E, FF) = 0;

end

OPR(E, FF) = ((RF(E, FF)*CNF(E, FF))*100)/EA((E-1),
FF);

if((RF_C >= 1)&& (CNF_C >= 1))||(del_C_D_i <=
del_C_D_E)

    Flag_Sim = 0;

end


end

[Opt_OPR_FR(FF), I] = max(OPR(:,FF));

Opt_FR(FF) = Q_D_V_i;

OPt_Rec_FR(FF) = Recovery_E(I, FF);
OPt_C_D_FR(FF) = C_D_in_E(I, FF);
Opt_Avg_WF_FR(FF) = Avg_WF_LMH(I,FF);

if (I == 1)

    I = 2;

end

OPt_NE_FR(FF) = EA((I-1),FF);

Q_D_V_i = Q_D_V_i + del_Q_D;

if (del_Q_D == 0)||(Q_D_V_i == (Q_D_V_f + del_Q_D))
```

```
    flag_D = 0;
    Prog_ind = Prog_ind + 1;

else

    FF = FF + 1;
    Prog_ind = Prog_ind + 1;
    flag_D = 1;

end

end

progress = ProgBar_Drawing(Prog_ind);

[Opt_OPR, I] = max(Opt_OPR_FR);

Opt_FlR = Opt_FR(I);
OPt_Rec = OPt_Rec_FR(I);
OPt_C_D = OPt_C_D_FR(I);
Opt_WF = Opt_Avg_WF_FR(I);
OPt_NE = OPt_NE_FR(I);

Opt_OPR_gui = Opt_OPR;
opt_Q_FS_gui = Q_F_inlet;
opt_Q_DS_gui = Opt_FlR;
Opt_NE_gui = OPt_NE;
Opt_Rec_gui = OPt_Rec;
Opt_C_DS_gui = OPt_C_D;

if (del_Q_D == 0)

    Q_D_ind = 1;

else

    Q_D_ind = ((Opt_FlR - Q_D_in_Ui)/del_Q_D)+1;

end

Prod_Per_PV = Q_D_in_E((OPt_NE+1), Q_D_ind);

Summary_excl = {'Optimum_DS_FR', 'Optimum_Recovery',
'Optimum_Final Concentration', 'Optimum_Water_Flux',
'Optimum_Element', 'Optimum_OPR'; Opt_FlR, OPt_Rec,
OPt_C_D, Opt_WF, OPt_NE, Opt_OPR};
```

```
end

function J_w = flux_PF_SW(Jw_i, C_F_b, C_D_b, pi_F_b,
pi_D_b, k_F, K_D)

global RSF B pi_F_e pi_D_e k_F_e K_D_e M_NaCl rho_NaCl

pi_F_e = pi_F_b;
pi_D_e = pi_D_b;
k_F_e = k_F;
K_D_e = K_D;

%Osmotoic Equilibrium check

F_m = C_F_b*exp(Jw_i/k_F);
D_m = F_m - (F_m - C_D_b*exp(-K_D*Jw_i))/(1-
(B/Jw_i)*(exp(-K_D*Jw_i)-1));

if (F_m >= D_m)

    fprintf('Osmotic Equilibrium is Reached')

end

J_w = wflux_PF_SW(Jw_i);

C_F_m = C_F_b*exp(J_w/k_F);
C_D_m = C_F_m - (C_F_m - C_D_b*exp(-K_D*J_w))/(1-
(B/J_w)*(exp(-K_D*J_w)-1));

RSF = B*((C_D_m - C_F_m)*M_NaCl)/rho_NaCl;

end

function y = wflux_PF_SW(Jw_i)

global A B k_F_e K_D_e pi_F_e pi_D_e

k_p = 0.25;
k_i = 0.05;
k_d = 0.05;
lmd = 0.1;

error_tolerance = 1e-20;

i = 1;
```

```
Itr(i) = 1;

if (Jw_i == 0)

    Jw_i = Jw_i + error_tolerance;

end

solution(i) = Jw_i;
LHS = solution(i);
RHS = A*((pi_D_e*exp(-solution(i)*K_D_e))-
(pi_F_e*exp(solution(i)/k_F_e)))/(1+(B/solution(i))*(e
xp(solution(i)/k_F_e)-exp(-solution(i)*K_D_e)));
error = RHS-LHS;
Err(i) = error;
P_act = k_p*error;
I_act = k_i*(0+(lmd^i*(i-0)*error));
D_act = 0;

while((error > error_tolerance)||(error < -
error_tolerance))

i = i+1;
Itr(i) = i;

Jw_i = solution(i-1)+ P_act + I_act + D_act;

if (Jw_i == 0)

    Jw_i = Jw_i + error_tolerance;

end

solution(i) = Jw_i;
LHS = solution(i);
RHS = A*((pi_D_e*exp(-solution(i)*K_D_e))-
(pi_F_e*exp(solution(i)/k_F_e)))/(1+(B/solution(i))*(e
xp(solution(i)/k_F_e)-exp(-solution(i)*K_D_e)));
error = RHS-LHS;
Err(i) = error;
P_act = k_p*error;
I_act = k_i*(I_act+(lmd^i*(i-(i-1))*error));
D_act = k_d*(Err(i) - Err(i-1))/(i-(i-1));

end

y = solution(i);
```

```
e = error;

end


function pos = position(h,v,n,o,g_gap)

if (v > h)

    d_1 = v/h;

    if (d_1 > 2)

        fprintf('Please check the value of dL and dW.')


    else

        p = rem(v,h);
        r = rem(v,p);
        s = (h-r)/p;

    end

elseif (q == p)

            s = 0;

elseif  (q < p)

            fprintf('Please check the value of dL and dW.')
            beep on
            beep
            beep
            beep
            beep off
end

beep off


i = 1;
P = n;

for I = 1:1:v
```

```
if (I == 1)

      pos(i) = P;
      counter = 1;
      flag = 0;


else

  if (I <= r)

      P = P-1;
      pos(i) = P;
      counter = 1;
      flag = 0;

  elseif (I > r)

    if (flag == 0)

    if (counter == 1)

      P = P - 1;
      pos(i) = P;
      flag = 0;
      counter = 2;

    elseif (counter == 2)

      P = P;
      pos(i) = P;
      flag = 1;
      counter = 1;
      counter_2 = 1;
    end

   elseif (flag == 1)

          P = P -1;
          pos(i) = P;
          flag = 1;
          counter_2 = counter_2 + 1;

       if (counter_2 == s )

          P = P;
```

```
          pos(i) = P;
          counter = 1;
          flag = 0;

       end

   end


  end


end

i = i + 1;

end

GP = o;
i = v+1;

while(i <= (v+g_gap))

    pos(i) = GP+1;
    i = i+1;
end

F = i;
P = GP+1;

a = F+v-1;

for I = F:1:a

if (I == F)

      pos(i) = P;
      counter = 1;
      flag = 0;


else


  if (I > F)&& (I < (a-r))

    if (flag == 0)
```

```
     if (counter == 1)

       P = P + 1;
       pos(i) = P;
       flag = 0;
       counter = 2;

     elseif (counter == 2)

       P = P;
       pos(i) = P;
       flag = 1;
       counter = 1;
       counter_2 = 1;
     end

    elseif (flag == 1)

           P = P+1;
           pos(i) = P;
           flag = 1;
           counter_2 = counter_2 + 1;

        if (counter_2 == s )

           P = P;
           pos(i) = P;
           counter = 1;
           flag = 0;

        end

     end

   elseif (I >= (a-r))

       P = P+1;
       pos(i) = P;
       counter = 1;
       flag = 0;

   end


end

i = i + 1;
```

```
end


end

function print =
xlswrite3(filename,data,sheet,first,option)


warning off MATLAB:xlswrite:AddSheet

% --Check for input errors
if nargin < 5
    error('This function requires that all five (5)
parameters be defined.  See help for more info.')
end
if ~ischar(filename)
    error('MATLAB:xlswrite:InputClass','Filename must
be a string');
end
[Directory,file,ext]=fileparts(filename);
if isempty(ext)
    ext = '.xls';
end
if isempty(data)
    error('MATLAB:xlswrite:EmptyInput','Input array is
empty.');
end
if ndims(data) < 3
    disp(' ');
    %disp(cat(2,'Your array is only
',num2str(ndims(data)),'-dimensional.  You could also
use XLSWRITE.'))
    disp(' ');
end
if ~(iscell(data) || isnumeric(data) || ischar(data))
&& ~islogical(data)
    error('MATLAB:xlswrite:InputClass',...
        'Input data must be a numeric, cell, or
logical array.');
end
if (~isnumeric(option) || option < 1 || option > 3)
    error('Invalid selection for parameter OPTION.
Select either 1, 2 or 3.')
end
```

```
% --End error check

if option == 1  %--------------------array(:,:,n) will
be placed underneath array(:,:,n-1) with a blank row
in between the two
    spacing = size(data,1) + 1;
    number = 0;
    y = 0;
    while number == 0
        y = y + 1;
        num = ~isletter(first(y));
        if num == 1
            number = 1;
        end
    end
    row = str2double(first(y:end));
    for t = 1:size(data,3)
        range = cat(2,first(1:y-1),num2str(row));
        xlswrite(filename,data(:,:,t),sheet,range);
        row = row + spacing;
    end

elseif option == 2  %----------------array(:,:,n) will
be placed to the right of array(:,:,n-1) with a blank
column in between the two
    spacing = size(data,2) + 1;
    alphabet = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ';
    letter = 1;
    y = 0;
    while letter == 1
        y = y + 1;
        let = isletter(first(y));
        if let == 0
            letter = 0;
        end
    end
    col = first(1:y-1);
    if length(col) == 1  %---------Turn string portion
of FIRST into a number to determine which column
        for z = 1:length(alphabet)
            if col == alphabet(z)
                column = z;
            end
        end
    elseif length(col) == 2
        for z = 1:length(alphabet)
```

```
            if col(1) == alphabet(z)
                column1 = z * length(alphabet);
            end
            if col(2) == alphabet(z)
                column2 = z;
            end
        end
        column = column1 + column2;
    end
    for t = 1:size(data,3)
        if column/length(alphabet) < 1  %---------
Convert column number back into a string
            columnstr = alphabet(column);
        else
            temp = num2str(column/length(alphabet));
            columnstr(1) =
num2str(alphabet(str2double(temp(1))));
            columnstr(2) = num2str(alphabet(column-
str2double(temp(1))*length(alphabet)));
        end
        range =
cat(2,columnstr,num2str(first(y:end)));
        xlswrite(filename,data(:,:,t),sheet,range);
        column = column + spacing;
    end

elseif option == 3  %----------------array(:,:,n) will
be placed in a separate worksheet from array(:,:,n-1),
named 'SHEETn'
    for t = 1:size(data,3)

xlswrite(filename,data(:,:,t),cat(2,sheet,num2str(t)),
first);
    end
end

end

function progress = ProgBar_Drawing(Prog_ind)

global m_PanelWidth m_PanelHeight Tab4_Q_D_in_i
Tab4_Q_D_in_f Tab4_del_Q_D_in Tab_4_Panel_Status
global Col_H5 Col_H7

Q_D_in_Ui = str2double(get(Tab4_Q_D_in_i, 'string'));
Q_D_in_Uf = str2double(get(Tab4_Q_D_in_f, 'string'));
```

```
del_Q_D_in = str2double(get(Tab4_del_Q_D_in,
'string'));

if (del_Q_D_in == 0)

    N_bar = 1;

else

    N_bar = ((Q_D_in_Uf - Q_D_in_Ui)/del_Q_D_in) + 1;

end

L_bar = 1/N_bar;

L_bar_i = 0;

ax_status = axes('Parent', Tab_4_Panel_Status,'Units',
'pixels','Position', [0.35*m_PanelWidth
0.027*m_PanelHeight 0.14*m_PanelWidth
0.05*m_PanelHeight],'XLim',[0 1],'YLim',[0 1],'Box',
'on', 'visible', 'off');

if (Prog_ind == 1)

    h = ax_status;
    rectangle('Position', [L_bar_i 0 L_bar 1],
'FaceColor',Col_H5,'EdgeColor','b','LineWidth',0.25);
    drawnow

elseif (Prog_ind > 1) && (Prog_ind <= N_bar)

    for N = 1:1:Prog_ind

        if (N < Prog_ind)

            h = ax_status;
            rectangle('Position', [L_bar_i 0 L_bar 1],
'FaceColor',Col_H7,'EdgeColor','b','LineWidth',0.25);
            drawnow

        else

            h = ax_status;
            rectangle('Position', [L_bar_i 0 L_bar 1],
'FaceColor',Col_H5,'EdgeColor','b','LineWidth',0.25);
            drawnow
```

```
            end

            L_bar_i = L_bar_i + L_bar;

        end

    else
            L_bar_i = L_bar*(N_bar-1);
            h = ax_status;
            rectangle('Position', [L_bar_i 0 L_bar 1],
'FaceColor',Col_H7,'EdgeColor','b','LineWidth',0.25);
            drawnow

    end

    progress = (Prog_ind/N_bar)*100;

    end
```