



Treball Final de Grau

**Design of a software executable from the "Wolfram Player" to
simulate Low-salt-rejection reverse osmosis (LSRRO) processes**

Andreu Navarro Balaguer

June 2024



UNIVERSITAT DE
BARCELONA

Aquesta obra està subjecta a la llicència de:
Reconeixement–NoComercial–SenseObraDerivada



<http://creativecommons.org/licenses/by-nc-nd/3.0/es/>

*The sea... a salt lake so big no merchant could
ever deplete. Saltwater that stretches all the way to
the horizon!*

Armin Arlert

This work could not have been possible without the help of Dr Joan Llorens Llacuna, whose experience in the sector goes without saying and guided me through the multiple iterations of the presented simulation software.

In this regard, I am also deeply indebted to German Fañanás from LEF Ingenieros, who provided me with an internship opportunity where I gained a lot of knowledge of the water treatment field. I am also grateful to my parents and sister, for their continuous support. Finally, I'd like to acknowledge all the professors and teachers that I have had during my entire studies, whose combined efforts resulted in leading me to where I currently am.

CONTENTS

SUMMARY	I
RESUM	III
SUSTAINABLE DEVELOPMENT GOALS	V
1. INTRODUCTION	1
1.1. Semipermeable membrane	2
1.2. Concentration polarization	3
1.3. Industry usage	5
1.4. Membrane scaling, fouling, and refurbishment	5
1.5. Low-salt-rejection reverse osmosis	7
2. OBJECTIVES	9
3. SIMULATION STATE OF THE ART REVIEW	11
4. DESIGN OF A LSRRO SIMULATION SOFTWARE	15
4.1. Simulation scope and limitations	15
4.2. Mathematical model	16
4.2.1. Mass balance model	17
4.2.1.1. Mass balance across each element	17
4.2.1.2. Mass balance in each node	17
4.2.2. Phenomenological equations	18
4.3. Solving philosophy	19
4.4. Defining system performance metrics	20
4.5. UI and controls	21
4.5.1. Control options	21
4.5.1.1. Water source characteristics	22
4.5.1.2. System configuration	22
4.5.1.3. Operating conditions	23
4.5.1.4. Equipment characteristics	23
4.5.1.5. Membrane characteristics	24

4.5.2.	Results screen	24
4.5.2.1.	Streams table	24
4.5.2.2.	KPI section	25
4.5.2.3.	System diagram	25
5.	RESULTS COMPARISON TO WAVE SOFTWARE	27
5.1.	Pressure and surface area variables	27
5.2.	Temperature variable	29
6.	OPTIMIZATION OF A LSRRO SYSTEM	33
6.1.	Definition of the system	33
6.2.	Simulation results	34
6.3.	Results discussion	39
6.4.	Future research proposals	40
7.	CONCLUSIONS	43
	REFERENCES AND NOTES	45
	ACRONYMS	47
	SYMBOLS	48
	APPENDICES	49
	APPENDIX 1: USER MANUAL	51
	APPENDIX 2: MATHEMATICA CODE FOR THE SIMULATION PROGRAM	55

SUMMARY

A new approach to improve efficiency of reverse osmosis systems by adding a low salt rejection stage, which can be retrofitted into existing water treatment configurations, is investigated by creating simulation in Wolfram Mathematica with a user-friendly UI.

The creation of this software is warranted as currently no other free alternatives exist which can simulate these complex systems with high accuracy.

Firstly, to ensure that the mathematical model in the presented software is accurate enough, it's compared against commercial reverse osmosis simulation software for validation, for which it shows agreement on average of 97%.

Then the program is used to optimize the low salt rejection system, for which the second stage's parameters are optimized, decreasing specific energy consumption by up to 2.8% for sea water reverse osmosis. It is proposed that further research be focused on brackish water treatment, which shows much better prospects.

Keywords: Reverse osmosis, desalination, water treatment, membrane, simulation, optimization, Mathematica.

RESUM

S'investiga una nova estratègia per millorar l'eficiència dels sistemes d'osmosi inversa afegint una etapa de baix rebuig de sals, que es pot adaptar a les configuracions existents de tractament d'aigua, mitjançant la creació d'una simulació en Wolfram Mathematica amb una interfície d'usuari fàcil d'usar.

La creació d'aquest programa ve justificada per la falta d'alternatives gratuïtes que puguin simular aquests sistemes complexos amb exactitud.

És per això que primerament, per assegurar l'exactitud del model presentat en aquesta obra, aquest es compara amb programari comercial de simulació d'osmosi inversa per a la seva validació, mostrant una concordança mitjana del 97%.

Aleshores, el programa s'utilitza per optimitzar el sistema d'osmosi inversa de baix rebuig de sals. Els paràmetres d'aquesta etapa addicional s'optimitzen, disminuint el consum específic d'energia fins a un 2,8% per a l'osmosi inversa d'aigua de mar. Per a la vinent recerca, es proposa d'optimitzar de cara a sistemes de tractament d'aigua salobre, que mostra molt millors pronòstics.

Paraules clau: Osmosis inversa, dessalació, tractament d'aigües, membrana, simulació, optimització, Mathematica.

SUSTAINABLE DEVELOPMENT GOALS

Drinking water supply has been a subject of increasing concern in the last few decades. According to the United Nations, water scarcity affects more than 40% of the world's population [1]. Especially in third-world countries, communities that not too long ago depended exclusively on surface water for their needs are now finding their rivers extremely contaminated, thus rendering them unsafe to drink from. Groundwater found in wells also suffers a similar problem, as excessive fertilizer drains into the soil to aquifers. Solving hygiene and agriculture problems is also the first step in improving the lives of these communities.

Reverse osmosis plants can process this water for reasonable cost, while being very down scalable and up scalable to fit the application's needs.

In addition, sea water desalination plants built around the globe in coastal cities also combat river overexploitation, alleviating the stress in these ecosystems. Catalonia currently has two installations; one in El Prat del Llobregat, which has 60 hm³/year capacity and the other in Tordera, with 20 hm³/year capacity, which is to be expanded to 80 hm³/year in the following years. Another plant is commissioned in Cunit, which will have 20 hm³/year capacity. By 2029, the Catalan Water Agency expects a total capacity of desalinated water of 160 hm³/year [2].

Many industries also face challenges when it comes to disposing of process water. Reverse osmosis for industrial wastewater treatment is gaining a lot of traction as a cost-effective method to suffice regulations for dumping water to the nature. In some processes, this purification is so effective that the water can be reused in the same plant.

1. INTRODUCTION

Reverse osmosis is a unit operation focused on water treatment. The process was developed in the 1950s [3] and it's based on generating a concentration difference across a semipermeable membrane using high pressure, thus reversing physical osmotic equilibrium [4].

Without an applied external pressure, the system tends to compensate for the osmotic pressure on both sides of the semipermeable membrane, that selectively only lets water molecules through.

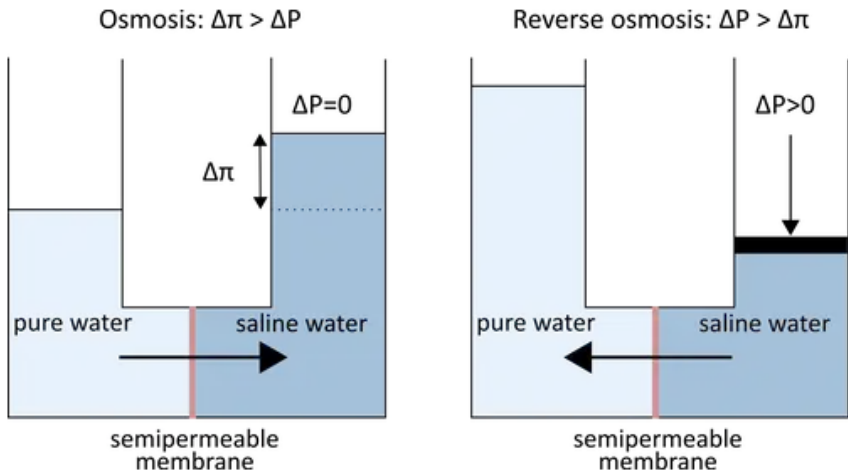


Figure 1. Effect of the semipermeable membrane, the OMD [7].

The breakthrough in the development came with John Cadotte's discovery of high flux and low salt passage membranes made by interfacial polymerization of m-phenylene diamine and trimesoyl chloride [5]. Even to this day, this is the preferred membrane for commercial applications. [6]

RO finds extensive use due to its energy efficiency when compared to now obsolete processes such as mechanical vapor compression or evaporation. Now more than ever,

desalination plants are being commissioned to provide a reliable alternative source of drinking water to combat drought and alleviate the impact on rivers.

1.1. SEMIPERMEABLE MEMBRANE

The main building block in the RO process is the thin film composite (TFC), where a dense layer of polyamide (50 to 200 nm thick, ~ 5 Å free volume size) is supported on a polysulfone layer (100 mm thick, 10 nm pore size). [7]

The TFC sheets are folded on themselves with a permeate spacer in-between, which leads to the perforated center permeate collector pipe. Each of these membrane leaves is spiral-wound around the permeate tube, separated with another channel spacer for the feed. [8]

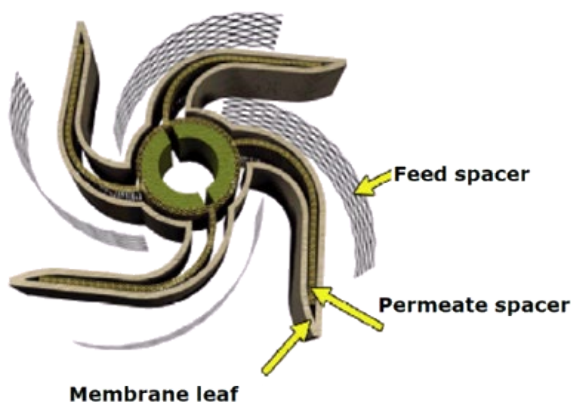


Figure 2. RO membrane construction, Lenntech [8]

This membrane assembly is finally compacted and encased in a cylindrical outer shell, producing modular elements that are housed and capped off in a longer pressure vessel made from fiber reinforced plastics to withstand the enormous pressures that these systems operate at. Commercially available PV contain 6 elements, but 7 and 8-element PV are also manufactured for special scenarios.

Feed water is forced through the membrane, which favors water molecules through to the permeate side so that solutes are retained in the concentrate side. It relies on the relative sizes of the solute molecules in comparison to water, and the lower the free volume size the more salt rejection, but water permeability also decreases.

Frequently, only a salt rejection parameter is reported to describe a membrane's performance, but actually, the salt rejection is not inherently specific to the membrane; it depends extensively on operation conditions.

Instead, two membrane-specific parameters describe membrane salt rejection in a rigorous way: A, the water permeability coefficient [LMH/bar] and B, the salt permeability coefficient [LMH].

These parameters define water and salt fluxes through the membrane, which are proportional to the respective driving force; difference in pressure for water flux, and difference in concentration for salt flux.

$$J_w = A \cdot (\Delta P - \Delta \Pi) \quad (1)$$

$$J_s = B \cdot \Delta c \quad (2)$$

With $\Delta \Pi$ being the osmotic pressure difference across the membrane:

$$\Delta \Pi = i \cdot R \cdot T \cdot \Delta c \quad (3)$$

Where i is the ionization constant, specific to each salt (i.e. 2 for NaCl), R being the ideal gas constant in [L·bar·K⁻¹·mol⁻¹], T being the absolute temperature [K] and Δc being the concentration difference across the membrane [mol/L].

1.2. CONCENTRATION POLARIZATION

The concentration on the feed side of the membrane cannot always be approximated to the feed stream concentration itself due to concentration polarization.

Concentration polarization is a natural phenomenon that is observed on the membrane surface, where, due to the accumulation of rejected solutes, a stationary layer forms which is more concentrated than the feed. The same applies to the permeate side of the membrane. The effect of CP can be approximated using film theory: [9]

$$c_{F,wall} = c_F \cdot e^{\frac{J_w}{k_F}} - c_P \left(e^{\frac{J_w}{k_F}} - 1 \right) \quad (4)$$

Where c_F is the bulk concentration of the feed stream, c_P the bulk concentration of the permeate stream, $c_{F,wall}$ the concentration on the membrane wall surface, J_w the local water flux through the membrane, and k_F the water mass transfer coefficient.

Considering that during normal operation, the concentration in the permeate side is negligible when compared with the one in the feed side, the equation can be simplified, and the concentration polarization modulus factor can be defined:

$$c_{F,wall} = c_F \cdot e^{\frac{J_w}{k_F}} \quad (5)$$

$$CPF = \frac{c_{F,wall}}{c_F} = e^{\frac{J_w}{k_F}} \quad (6)$$

Given this formula, it checks out that for a very large mass transfer coefficient or a very small water flux through the membrane, no solute accumulates, so there is no CP.

The water mass transfer coefficient k_F can be calculated from the Sherwood-Reynolds-Schmidt correlation: [9]

$$k_F = Sh \left(\frac{D}{d_h} \right) \quad (7)$$

$$Sh = 0.16 Re^{0.605} Sc^{0.42} \quad (8)$$

$$Re = \frac{d_h u \rho}{\mu} \quad (9)$$

$$Sc = \frac{\mu}{\rho D} \quad (10)$$

The hydraulic diameter of the spiral wound membrane, d_h , is taken to be a typical value of 0.95 mm [7] [9]. Both the fluid density ρ and fluid viscosity μ can be approximated to those of pure water, so 1000 kg/m³ and 10⁻⁶ Pa·s respectively.

The crossflow velocity u can be assumed to be between 0.05 and 2 m/s which is usual for RO operating conditions. [10]

At 25 °C, the water diffusivity [m²/s] can be approximated using the following experimental correlation:

$$D = 6.73 \times 10^{-6} \exp(0.00903c - 8.43) \quad (11)$$

Where c is the molar concentration, which for a sea water desalination plant is approximately 0.6 M.

For 0.05 m/s CFV, a k_F of 10⁻⁵ m/s is obtained, and for a 2 m/s CFV, 10⁻⁴. These results give an average k_F of 5·10⁻⁵ m/s, which is taken to be constant so simplify simulations, as big fluctuations in CFV in the feed-concentrate side are not expected during operation.

1.3. INDUSTRY USAGE

In an industrial setting, multiple pressure vessels in series can be used. Two options are possible depending on after which stream the additional PV are placed.

Staging is the term used to describe concentrate streams leading to other PV as feed, instead of using a longer PV.

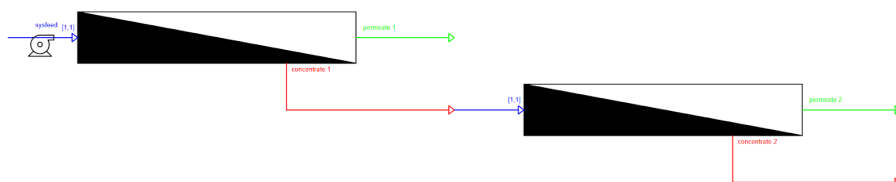


Figure 3. Box diagram of a 2-stage RO configuration

On top of stages, setups may also implement passes, especially when aiming for a high salt rejection, such as in seawater desalination plants for drinking water. In a pass, the PV permeate stream is fed to another PV as feed.

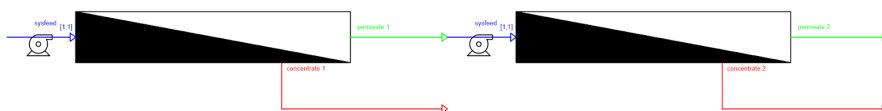


Figure 4. Box diagram of a 2-pass RO configuration

In essence, staging is akin to concentrate recycle and passing is similar to permeate recycling.

Stream recycling does not necessarily need to feed into the same PV; a stream can also be recycled back to a previous one. Such is the basis of LSRRO; combining staging with permeate recycling to a previous stage.

1.4. MEMBRANE SCALING, FOULING, AND REFURBISHMENT

Even with pretreatment steps such as nanofiltration before the RO system, after continuous operation, all membranes eventually need replacement due to fouling and scaling.

Fouling occurs continuously when particulate contaminants in the feed accumulate on the membrane surface, reducing effective membrane area and leading to other problems. Biofouling is by far the most recurring type of fouling faced in the industry, the main cause being RO

membranes' lack of chlorine tolerance, which prevents systems from efficiently and economically preventing algae growth. [11]

On the other hand, scaling is due to the precipitation of poorly soluble inorganic compounds such as CaCO_3 , CaSO_4 , BaSO_4 , SrSO_4 or SiO_2 in spots where the concentration exceeds solubility limits, such as near the concentrate stream exit. The formed hard crystalline layer is impossible to remove mechanically and can only be eliminated by dissolving it away. Low pH cleaning solution at a warm temperature is used industrially for this purpose during the CIP procedure.

Membrane CIP allows cleaning without the need to disassemble the water treatment system by passing specialized solutions through the system. These include antiscalants and biocides among other chemicals. At the end of the CIP, the system is rinsed thoroughly. [12]

Overall, membrane scaling and fouling decrease membrane water and salt permeability, which greatly increases pressure drop over time, despite regularly cleaning the membranes. Using DuPont's WAVE software, the performance of a system can be simulated over time. According to Dupont's FilmTec membranes operation manual, for a new membrane a flow factor is 1.00 which drops to 0.85 with some use, and for old systems it's as low as 0.6.

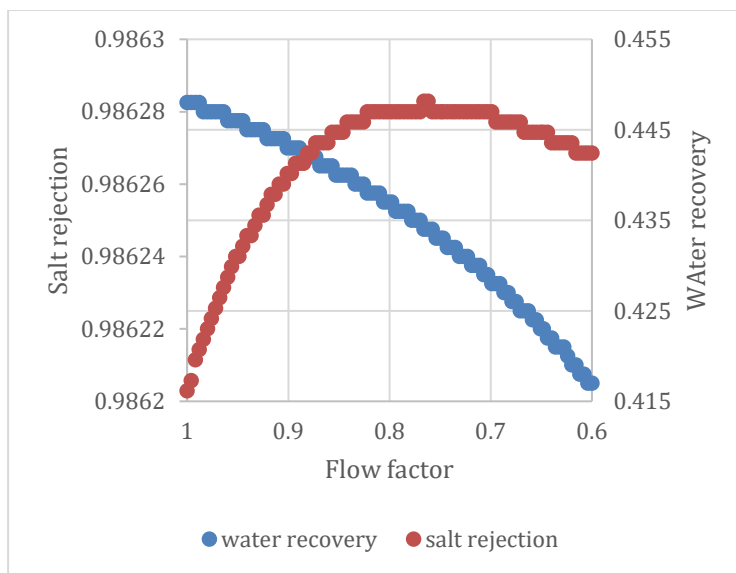


Fig 5. Simulated effect of membrane fouling in 10PVx6 Seamaxx membranes at 54 bar, 100 m³/h 35 g/L TDS (WAVE)

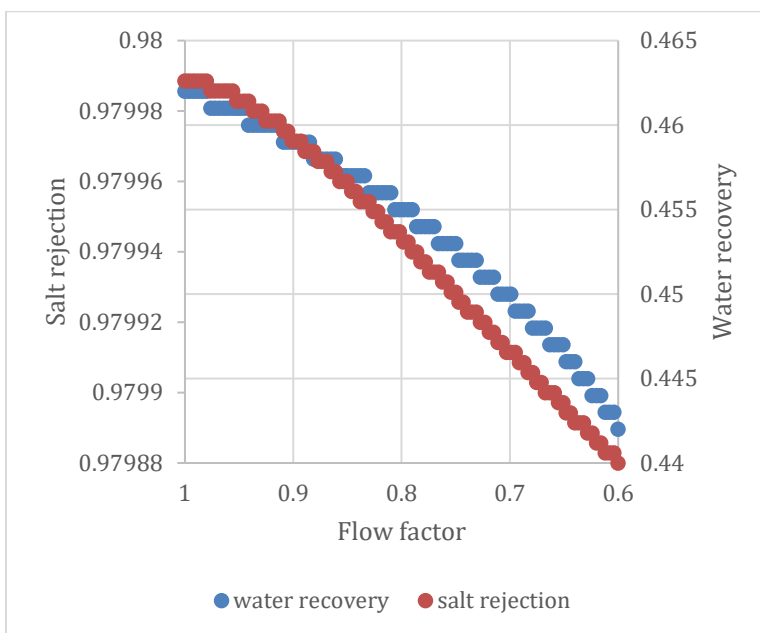


Fig 6. Simulated effect of membrane fouling in 15PVx6 Seamaxx membranes at 52 bar, 100 m³/h 35 g/L TDS (WAVE)

From the simulated performance of these systems, operating in recommended conditions, a total lifetime decrease of water recovery of 6.9% and 4.3% respectively is observed before these membranes are to be replaced. They can be salvaged and refurbished to use in LSRRO systems.

The refurbishing process consists of passing an oxidizing solution through the membrane in a controlled manner. Oxidizing compounds punch holes in the thin polyamide layer, increasing its water and salt permeability, to counteract the fouling and scaling.

1.5. LOW-SALT-REJECTION REVERSE OSMOSIS

LSRRO (Low-Salt-Rejection Reverse Osmosis) is an emerging technology that allows fouled RO membranes to be reused after undergoing a special treatment.

Reused RO membranes are to be placed after the initial stage, in a LSRRO stage. Due to their lower salt rejection, the remaining pressure from the concentrate of the previous stage is

enough to drive forward the separation without the need for booster pumps. The permeate of the LSRRO stage is returned as feed to the first RO stage.

More than one LSRRO stage can be used, improving the overall system performance, similar to how more plates in a rectification column improve separation.

In this scenario, the membranes go down a stage after they reach their end of life and are treated. Membranes must be ordered from higher to lower permeability so that the system operates optimally.

Nevertheless, a 2-stage system is the most practical setup, as no additional booster pumps are needed.

2. OBJECTIVES

The main goal of this work is to produce a simulation in Wolfram Mathematica language for reverse osmosis systems from scratch, that is generalized for a wide range of applications, and that is accurate, that is, that gives results that agree with commercial simulation software.

Then, using this simulation, optimum parameters for the system which maximize the system power efficiency will be determined.

3. SIMULATION STATE OF THE ART REVIEW

Out of all the commercially available simulation software, Dupont's WAVE (Water Application Value Engine), formerly known as ROSA, is the state of the art when it comes to dedicated RO simulating software. It is unrivaled in the water treatment industry, with its main appeal being its user-friendly interface combined with its high simulation accuracy validated with real-life results.

The screenshot displays the WAVE software interface for Reverse Osmosis simulation. The top navigation bar includes 'Home', 'Feed Water', 'Reverse Osmosis', and 'Summary Report'. The 'Reverse Osmosis' tab is active, showing the 'Reverse Osmosis Pass Configuration' section.

Reverse Osmosis Pass Configuration

Configuration for Pass 1

Number of Stages: ☐ 1 ☒ 2 ☐ 3 ☐ 4 ☐ 5

Flow Factor:

Temperature: Design °C

Pass Permeate Back Pressure: bar

Flows

Feed Flow: m³/h

Recovery: %

Permeate Flow: m³/h

Flux: LMH

Conc. Recycle Flow: m³/h

Bypass Flow: m³/h

Stages

	Stage 1	Stage 2
# PV per stage	10	1
# Els per PV	6	
Element Type	SW30XLE-400	ISD (Internally Staged Design)
Specs		EdL ISD
Total Els per Stage	60	6
Pre-stage ΔP (bar)	0.31	0.20
Stage Back Press (bar)	0.00	0.00
Boost Press (bar)	N/A	0
Feed Press (bar)	0	N/A
% Conc to Feed	0.00	0.00
Flow Factor	0.85	0.85

System Configuration

The diagram shows a two-pass system. Feed water enters a pump and is distributed to two stages. Each stage produces permeate and concentrate. The permeate from the first stage is recycled back to the feed of the second stage. The final permeate and concentrate streams are shown at the bottom.

© 2022 DuPont de Nemours Inc. All rights reserved. Water Application Value Engine Water Solutions

Figure 7. WAVE main UI.

Its ability to customize systems down to stages with internally staged designs (different elements in the same pressure vessel) and to define feed water ion-specific composition and temperature, adding pH adjustment, degassing, and feed pretreating with ultrafiltration is impressive, while allowing the user to optimize the process to meet specified criteria.

Its computation speed is almost instant even with a very loaded design such as a two-pass with 5 stage each configuration. The summary reports produce detailed pressure losses in each PV and alert the user when element design warnings are detected.

However, this program is limited to simulating DuPont's brands of RO membranes. Given that its code is not open-source, modification of the software to allow custom membranes with user-defined parameters is impossible. These membranes are described in the interface in terms of their salt recovery, which is known to not be a membrane-specific parameter. LSRRO modeling requires A and B parameters to be inputted, as its optimization is based on finding the appropriate treatment for the refurbished stage.

Element	Active Area	Pressure	Flow	Rejection(%)	Conc. (ppm)	Salt	Recovery (%)	Diameter
Eco Pro-400	400 (37.2)	150 (10.3)	11,500 (43.5)	99.7	2000	NaCl	15	8
Eco Pro-400i	400 (37.2)	150 (10.3)	11,500 (43.5)	99.7	2000	NaCl	15	8
Eco Pro-440	440 (40.9)	150 (10.3)	12,650 (47.9)	99.7	2000	NaCl	15	8
Eco Pro-440i	440 (40.9)	150 (10.3)	12,650 (47.9)	99.7	2000	NaCl	15	8
Eco Platinum-440	440 (40.9)	150 (10.3)	12,650 (47.9)	99.7	2000	NaCl	15	8
Eco Platinum-440i	440 (40.9)	150 (10.3)	12,650 (47.9)	99.7	2000	NaCl	15	8
ECO-400i (obsolete)	400 (37.2)	150 (10.3)	11,500 (43.5)	99.7	2000	NaCl	15	8
ECO-440i (obsolete)	440 (40.9)	150 (10.3)	12,650 (47.9)	99.7	2000	NaCl	15	8
XLE-440	440 (40.9)	125 (8.6)	14,000 (53)	99.0	2000	NaCl	15	8
XLE-B-440i (france pota)	440 (40.9)	125 (8.6)	14,000 (53)	99.0	2000	NaCl	15	8
XLE-440i	440 (40.9)	125 (8.6)	14,000 (53)	99.0	2000	NaCl	15	8
XLE-4040	87 (8.1)	100 (6.9)	2,600 (9.8)	99.0	500	NaCl	15	4
XLE-2540	28 (2.6)	100 (6.9)	850 (3.2)	99.0	500	NaCl	15	2.5
HRLE-440 (obsolete)	440 (40.9)	150 (10.3)	12,650 (47.9)	99.5	2000	NaCl	15	8
HRLE-440i (obsolete)	440 (40.9)	150 (10.3)	12,650 (47.9)	99.5	2000	NaCl	15	8
BW30HRLE-440	440 (40.9)	150 (10.3)	12,650 (47.9)	99.3	2000	NaCl	15	8
BW30HRLE-440i	440 (40.9)	150 (10.3)	12,650 (47.9)	99.3	2000	NaCl	15	8
BW30FR-365 (obsolete)	365 (33.9)	225 (15.5)	9,500 (36)	99.5	2000	NaCl	15	8
BW30XFR-400/34	400 (37.2)	225 (15.5)	11,500 (43)	99.65	2000	NaCl	15	8

Active Area units: square feet (square meter)
Pressure units: psi (bar)
Flow units: gallons per day (cubic meters per day)
Double-click on a row above to select the appropriate product

OK Cancel

Figure 8. Available membranes for simulation in WAVE, with their specifications.

Besides, this program presents few configuration options for complex RO configurations other than simple staging, passes, feed bypasses and concentrate recycles. Specifically, it doesn't allow for total permeate recycles, which is crucial for the design of LSRRO systems.

RO System Feed Flow Rate

☒ Automatic m³/h

RO System Summary

Feed Flow m³/h

Product Flow m³/h

Concentrate Flow m³/h

System Recovery %

Pass 1

Net Feed Flow m³/h

Permeate Flow m³/h

Flux LMH

Conc. Flowrate m³/h

Recovery %

Conc. Recycle to head of: % m³/h

Pass Size Optimization

☐ Bypass % of feed to head of Pass 2 m³/h

☐ Permeate Split % of permeate drawn from front and bypassed to Product m³/h

☒ None

Pass 2

Net Feed Flow m³/h

Permeate Flow m³/h

Flux LMH

Conc. Flowrate m³/h

Recovery %

Conc. Recycle to head of: % m³/h

Concentrate Recycle Split

to Pass1:(0.00 % to Pass2:(0.00 %)

% % m³/h

Pass Size Optimization

☐ Bypass % of feed to RO Product m³/h

☒ None

Figure 9. RO system configuration options UI.

Other honorable mentions of free RO simulating software include Winflows, from Veolia Water Technologies and Solutions, which while not as trusted as WAVE by the industry, it's similarly capable. However, much like WAVE, it also doesn't allow the user to input A B parameters. [13]

Another free program is WaterTAP, which open source, based in Python language, and allows for a lot of customization but ultimately lacks a proper UI. [14]

On the other hand, ROSIM is a toolkit developed at the University of Valladolid (Spain) and validated with experimental data from a pilot plant in the Research and Technology Center of Energy (CRTE), Tunisia. [15]

Given the numerous available options, none satisfy all the previously mentioned requirements, which is the reason why businesses designing processes consisting of an LSRRO setup with custom membranes, which is the subject of this paper, require similarly accurate free simulation software with a user-friendly UI.

4. DESIGN OF A LSRRO SIMULATION SOFTWARE

An interactive application was developed in Mathematica language, which allows for the simulation of an LSRRO system given its membranes A B parameters and its operating conditions, such as feed TDS and pressure. Mathematica was the language of choice due to the vast number of prebuilt powerful functions and its optimized nature. It has very efficient built-in functions to find exact solutions to large non-linear systems of equations and to build interactive applications with easily customizable controls.

It must also be noted that while Mathematica is not free to write code with, a Player version is available at no cost. This version allows the use of Mathematica exported notebooks without making changes to the code.

Other options have been considered, such as Python language, which although powerful, can be hard to optimize and to choose appropriate libraries to import. SMath software was also on the table but ultimately discarded due to the cluttered user interface.

Mathematica stands as the simplest yet most powerful coding language, perfect for the needs of this project.

4.1. SIMULATION SCOPE AND LIMITATIONS

The produced software can simulate LSRRO systems of an arbitrary number of stages thanks to the generalized way in which it is coded. However, the front-end UI is limited to up to 3 stages maximum, to keep the UI simple and focused on real-world applications.

The results output includes flow rates and TDS of all streams in an organized table and the main streams over a system configuration diagram. In addition, observed salt rejection and water recovery and SEC values are presented, which are essential parameters for quickly designing systems. It must be noted though, that this simulation does not account for pressure drop in the concentrate streams.

Besides the main controls, the user is also given the option to change the salt in the feed. A specific salt can be selected, or an average different molecular mass can be inputted. This simulation cannot be used to determine the distribution of different salts.

The user can also increase the accuracy of the simulation at the cost of computation time by increasing the stage element discretization parameter and increasing the number of digits used in internal calculations.

4.2. MATHEMATICAL MODEL

The mathematical model is the equations that need to be solved for each case. It combines both material balance equations and phenomenological equations describing mass transfer.

This simulation uses a perfect plug flow approximation, for which each real stage is divided into elements in series, which behave as a discretized stage. The concentrate flows from one stage element to the next one, while the permeate is collected in a central tube. Equations for each stage element are based on the flow and concentration of itself, rather than the bulk concentration in the collector tube.

The nomenclature for the streams is as follows:

$$[i, j, k]$$

With i being the stage number, j the element number inside that stage, and k the stream type: 1 for feed, 2 for permeate, and 3 for concentrate. Having the same nomenclature for all the stages facilitates the generalization of the problem.

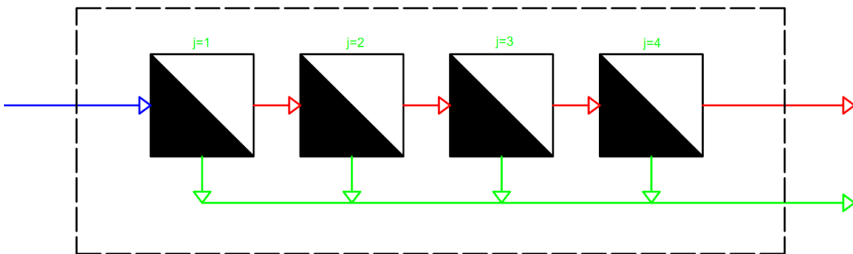


Figure 10. Discretization of a stage into four virtual stages.

4.2.1. Mass balance model

Two components are to be considered: the water and the salt. For each of these two components, two different sets of mass balance equations are used:

4.2.1.1. Mass balance across each element

The feed is divided into the permeate stream and concentrate stream. For $i=1$ to $i=Nstages$ and $j=1$ to $j=Nelements$, the following mass balances apply:

$$w[i, j, 1] = w[i, j, 2] + w[i, j, 3] \quad (12)$$

$$w[i, j, 1] \cdot c[i, j, 1] = w[i, j, 2] \cdot c[i, j, 2] + w[i, j, 3] \cdot c[i, j, 3] \quad (13)$$

The mass balance for the salt could have been simplified by assuming that all the salt in the feed ends in the concentrate stream, as the salt concentration in the permeate is negligible. However, such simplification was observed not to significantly speed up computation, so to preserve simulation accuracy, the non-simplified salt mass balance is used.

4.2.1.2. Mass balance in each node

A node in this context is defined as the mixing point of a concentrate stream and a permeate stream of the next stage. This set of equations really only apply to the first element in each stage, for the rest it's merely in place to simplify and generalize the model as much as possible.

For each stage, for its initial element $j=1$:

$$w[i, 1, 1] = w[i - 1, Nelements, 3] + \sum_{j=1}^{Nelements} w[i + 1, j, 2] \quad (14)$$

$$w[i, 1, 1] \cdot c[i, 1, 1] = w[i - 1, Nelements, 3] \cdot c[i - 1, Nelements, 3] + \sum_{j=1}^{Nelements} w[i + 1, j, 2] \cdot c[i + 1, j, 2] \quad (15)$$

This applies for all stages except the last one, because that one does not have a permeate recirculation leading back to it. For the last stage, the trivial equation is used instead (still only for its initial element, $j=1$):

$$w[i, 1, 1] = w[i - 1, Nelements, 3] \quad (16)$$

$$w[i, 1, 1] \cdot c[i, 1, 1] = w[i - 1, Nelements, 3] \cdot c[i - 1, Nelements, 3] \quad (17)$$

Then for each element that is not the first in its stage:

$$w[i, j, 1] = w[i, j - 1, 3] \quad (18)$$

$$w[i, j, 1] \cdot c[i, j, 1] = w[i, j - 1, 3] \cdot c[i, j - 1, 3] \quad (19)$$

The addition of these trivial last equations does not substantially affect the computation time of the simulation, but are useful because they allow the system to be very generalized with the same index notation.

4.2.2. Phenomenological equations

The water and salt fluxes can be described with equations 20 and 21 Which applies to all elements in all stages.

$$J_w[i, j] = A[i] \cdot (\Delta P - (\Pi[c[i, j, wall]] - \Pi[c[i, j, 2]])) \quad (20)$$

$$J_s[i, j] = B[i] \cdot (c[i, j, wall] - c[i, j, 2]) \quad (21)$$

The osmotic pressure is determined by the usual expression, which for moderate concentrations is deemed valid:

$$\Pi = iRT \cdot c \quad (22)$$

These equations use the concentration at the wall of the membrane, which because of polarization, cannot be considered equal to the bulk concentration of the feed.

$$c[i, j, wall] = CPF[i, j] \cdot c[i, j, 1] \quad (23)$$

This expression is already substituted in, to avoid working with another k index value. The concentration polarization modulus is calculated with equation...

$$CPF[i, j] = e^{J_w[i, j]/k_f} \quad (24)$$

Where k_f is the mass transfer coefficient, which is assumed to be constant at $5 \cdot 10^{-5}$ m/s (which is 180 LMH) for all normal range of operation conditions, as outlined in section 1.2.

To establish a relationship between fluxes and the actual streams, the membrane active area is used:

$$w[i, j, 2] = J_w[i, j] \cdot S[i]/Nelements \quad (25)$$

$$w[i, j, 2] \cdot c[i, j, 2] = J_s[i, j] \cdot S[i]/Nelements \quad (26)$$

Because these equations are defined for each element, but the total membrane area of the stage is used, it is crucial to divide it by the number of elements in each stage.

4.3. SOLVING PHILOSOPHY

Solving the system of equations traditionally for a large number of stage elements using Mathematica's NSolve functionality is not viable in a reasonable amount of time so that is not jarring to the user. Instead, FindRoot function is used. FindRoot finds a solution by numerical iteration provided a starting value for each variable.

The philosophy is that by solving the system first with NSolve without discretizing each stage (Nelements set to 1), a reasonable solution can be obtained. This solution is then used as a starting value for all the elements in the discretized stages in the FindRoot function.

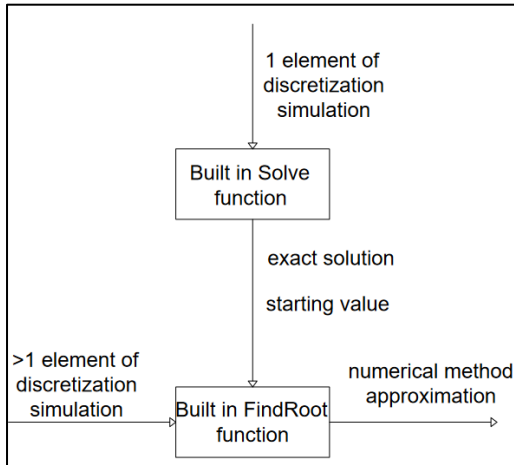


Figure 11. Initial value adquisition for the numerical method pseudocode.

Results appear to differ considerably when switching from 1 element of discretization to 2. 3 or more elements have negligible impact. The more elements, the harder it is for the solution to converge, and more often than not, nonsensical solutions are obtained.

To speed up the convergence and accuracy of this internal iteration process, several settings can be changed. Setting the iteration method to AffineCovariantNewton instead of the Newton default seems to improve accuracy substantially. The working precision and precision goal can also be increased.

Another layer to the solving philosophy is the iteration around the CPF. Due to the exponential equation defining CPF, introducing it as another equation to solve in the system of

equations adds a lot of instability, leading to nonsensical results or the simulation not converging at all.

To avoid these problems, the program solves the system for an initial set of CPF values, and then uses the results to recalculate the CPF. It keeps iterating until the newly calculated CPF differs less than 0.001 to the previous one.

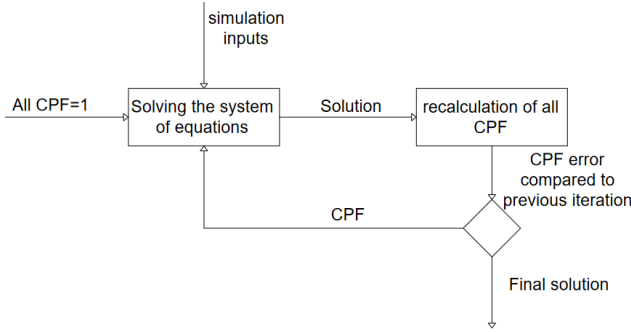


Figure 12. Iteration around the CPF values pseudocode.

A limit to the CPF of 2 has been hard-coded, else, the simulation loops endlessly without converging. A CPF value this high is unlikely in normal operation. Otherwise, this value can be modified in the code for special cases.

4.4. DEFINING SYSTEM PERFORMANCE METRICS

To evaluate the performance of different configurations and operating conditions, system performance metrics or KPI need to be defined.

For this study, the focus is in the specific energy recovery. Salt rejection is another important metric, but any RO system operating in usual conditions, it is always approximately equal to 100%, so it is not very distinctive and useful to optimize. [15]

SEC is a function of applied hydraulic pressure and water recovery, which is in turn also function of applied hydraulic pressure. For simple 1-stage configurations, water recovery is approximately linearly dependent on applied hydraulic pressure, so the SEC has an asymptotic shape:

$$SEC = \frac{P}{w_p} = \frac{w_F \cdot \Delta P}{w_p \cdot \eta} = \frac{\Delta P}{R_w \cdot \eta} = \frac{\Delta P}{b \cdot \Delta P - a} \quad (27)$$

$$\lim_{\Delta P \rightarrow \infty} SEC = \frac{1}{b} = SEC_{min} \quad (28)$$

Theoretically, a minimum SEC is possible, provided a large enough applied hydraulic pressure. In this limit, the SEC does not depend on the applied hydraulic pressure. The slope b depends only on the total membrane surface area and pump efficiency.

On the other hand, for more complex configurations, like 2-stage systems, the relation is not as straight-forward:

$$SEC = \frac{P}{w_P} = \frac{(w_F + w_{2,P}) \cdot \Delta P}{w_{1,P} \cdot \eta} = \frac{\Delta P}{\eta} \cdot \left(\frac{1}{R_w} + \frac{w_{2,P}}{w_{1,P}} \right) \quad (29)$$

For 3-stage and more configurations, the power consumption of additional pumps needs to be considered. For example, for a 3-stage system:

$$SEC = \frac{P}{w_P} = \frac{(w_F + w_{2,P}) \cdot \Delta P_1 / \eta_1 + w_{2,C} \cdot \Delta P_2 / \eta_2}{w_{1,P}} \quad (30)$$

The fact that 3-stage systems are not common industrially, and the added complexity of having another variable to optimize around, these configurations are out of the scope of this study.

4.5. UI AND CONTROLS

The program has been outfitted with a visually appealing UI that presents the results in an intuitive way. Two distinct sections can be observed: the first one containing the control options for inputs, and the second one presenting the results.

4.5.1. Control options

The control options are further divided into smaller subsections, each serving a specific purpose. It is expected that the user will rarely interact with controls pertaining to different subsections. The expected usage is that the user fixes some controls and only simulates variations of another control.

Thanks to Mathematica's implementation, by pressing the plus sign (+) beside the slider controls, a menu appears where the user can also start a simulation that goes through a wide range of inputs automatically, with an option to control the step size and speed.



Figure 13. Expanded slider control in the UI.

Additionally, all controls come with usual default values upon opening the program, this way the user can focus on changing only the variables of interest.

4.5.1.1. Water source characteristics

This subsection contains controls intrinsic to the water source, that are unlikely to be changed continuously during simulations, hence why they have been separated from the rest.

An input field is used to input the water source temperature in Celsius degrees, and a setter bar is used to choose the salt. The predefined list of NaCl and MgSO₄ can be easily expanded to contain other common salts.

If the user desires to simulate a salt not present in the list, or an average molecular weight accounting for several different salts in the water source, an input field is offered. The user must also input the Van't Hoff coefficient in the adjacent input field.

Figure 14. Salt and temperature inputs in the UI.

4.5.1.2. System configuration

The user can change the number of stages of the LSRRO system here. Setting the option to 1 is the same as a simple single stage RO configuration. The interface limits this option to up to 3 stages, as more stages are slow to simulate, on top of not being industrially useful.

The option to increase stage discretization is also found here. Again, this option has been limited to 4, as further increasing the number of elements requires exponentially more computational power and introduces a lot of numerical instability. It has also been found that the simulation results do not considerably change when using a discretization bigger than 4 elements.

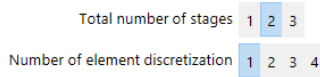


Figure 15. System configuration selection in the UI.

4.5.1.3. Operating conditions

Three sliders can be used to adjust operating conditions, namely system feed flow rate and TDS, and pump applied hydraulic pressure. Sliders have been chosen for these variables as it's likely the user will want to, for example, quickly check how an existing system behaves under different operating conditions.

However, the drawback to using sliders is that they have a limited input range by design. This limitation can be overcome by directly entering the input in the small box right to the slider. This way valid inputs outside the slider range can be used for simulation.



Figure 16. Operating conditions slider section, with the system feed TDS outside the range.

When using three or more stages, another pump is needed. This pump's applied hydraulic pressure cannot be separately defined, it's set to be equal to the first pump.

The pressure is to be inputted as the pressure difference between the feed and the permeate stream. The program uses gauge pressures, so in most cases, because the permeate stream exits to ambient pressure then the applied hydraulic pressure across the system is the same as the gauge pressure at the feed. In case where the permeate has some backpressure, this must be considered, and the subtraction performed.

$$\Delta P = P_{feed} - P_{permeate} \quad (31)$$

4.5.1.4. Equipment characteristics

In this subsection, input fields for the pump efficiency, pump motor efficiency and energy recovery device efficiency are presented. For systems with multiple pumps, such as 3 or more stage configurations, individual pump efficiencies cannot be defined.

Pump Efficiency	<input type="text" value="0.84"/>	Motor Efficiency	<input type="text" value="0.95"/>	ERD Efficiency	<input type="text" value="1."/>
-----------------	-----------------------------------	------------------	-----------------------------------	----------------	---------------------------------

Figure 17. Equipment efficiencies input boxes in the UI.

4.5.1.5. Membrane characteristics

The membrane specific parameters A and B are to be defined here. An input box for each stage's membrane is provided. The option to specify the membrane active surface area is also included for each stage.

A1 [LMH/bar]	<input type="text" value="0.7"/>	A2 [LMH/bar]	<input type="text" value="7."/>	A3 [LMH/bar]	<input type="text" value="60."/>
B1 [LMH]	<input type="text" value="0.2"/>	B2 [LMH]	<input type="text" value="4.5"/>	B3 [LMH]	<input type="text" value="25."/>
S1 [m ²]	<input type="text" value="2230."/>	S2 [m ²]	<input type="text" value="350."/>	S3 [m ²]	<input type="text" value="100."/>

Figure 18. Membrane parameters input boxes in the UI.

When using a system configuration of less than 3 stages, the unused stages parameter input boxers have no effect on simulation.

4.5.2. Results screen

In the results section, three different areas are presented in an organized manner. The results update in real time when the user makes changes to the inputs.

4.5.2.1. Streams table

The flow rate and TDS for all the streams is presented in a table, with columns as stream types (feed, permeate or concentrate) and rows as stage number. The permeate of the first stage and the concentrate of the last stage are highlighted as they are the out streams when treating the whole system as a black box.

Streams as {flow rate[m³/h], TDS[g/L]}

stage	1-feed	2-permeate	3-concentrate
1	{119.314, 30.2553}	{54.5738, 0.3334}	{64.7397, 55.4787}
2	{64.7397, 55.4787}	{19.3135, 5.6885}	{45.4262, 76.6476}

Figure 19. Streams table in the UI.

4.5.2.2. KPI section

The performance of the system is shown here. The first row underlines the system power consumption and SEC without an ERD in use, while the second row includes an ERD with the previously defined efficiency in the calculations.

Lastly, the overall system observed salt rejection and water recovery are presented. While not a good measure of performance, it might still be useful to the user to be provided these values, otherwise easy to determine from the information in the streams table.

Power consumption [kW]:	228.43	SEC [kWh/m ³]:	4.19
Net Power consumption (w/ ERD) [kW]:	159.03	SEC [kWh/m ³]:	2.91
Observed salt rejection:	0.9905	water recovery:	0.5457

Figure 20. KPI section in the UI.

The number of significant digits in these results have been limited in order to preserve repeatability, else the user might have found that the same input gives different results. This is since the program uses different starting values for solving with numerical methods depending on the previous result.

4.5.2.3. System diagram

A graphical representation of the system configuration is shown, with the primary system streams information included. The stream type is color coded; blue for feed, green for permeate, and red for concentrate. Each stream is identified with an index corresponding to the one in the streams table.

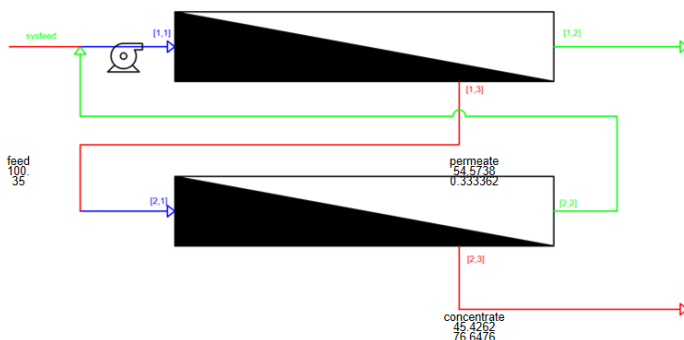


Figure 21. 2-stage configuration diagram in the UI.

This diagram also updates based on the number of stages selected and includes the position of the pumps accordingly. If the source Mathematica code for this program was to be modified to allow the simulation of more than three stages, additional diagram drawings would have to be made.

5. RESULTS COMPARISON TO WAVE SOFTWARE

While not a real-world validation of results can be conducted due to the nature of the project, which aims to simulate a new technology, the mathematical model can be tested by comparing results with WAVE software for 1-stage system configurations. As outlined in section 2, WAVE stands as the most trusted water treatment simulation software by the industry, so that's why it will be used as a source of accurate data. It is clear however that DuPont itself doesn't guarantee that their products will perform exactly the same as in the simulations.

In order to use the same equipment in both simulations for a proper comparison, the correct inputs must be given in the Mathematica model from what little can be known of the membrane that is selected inside WAVE. The exact A B parameters used for each membrane inside WAVE are initially unknown, but can be determined indirectly using the result data provided in the simulation report.

Additionally, it has been observed that the calculated A B parameters change from one simulation to another. Therefore it is suspected that WAVE does not actually use constant membrane specific parameters for its internal calculations.

For the sake of comparison, the calculated A B parameters from the wave simulation are given as inputs in the Mathematica model, along with the applied hydraulic pressure and total membrane surface area.

5.1. PRESSURE AND SURFACE AREA VARIABLES

For a feed stream of 100 m³/h and 35 g/L NaCl TDS, a wide range of system operations has been simulated with both WAVE and the Mathematica model.

In this configuration, each PV is composed of 6 Seamaxx membrane element modules, each of which has a 440 ft active surface area, and the total efficiency of the pump is assumed to be a constant 0.798.

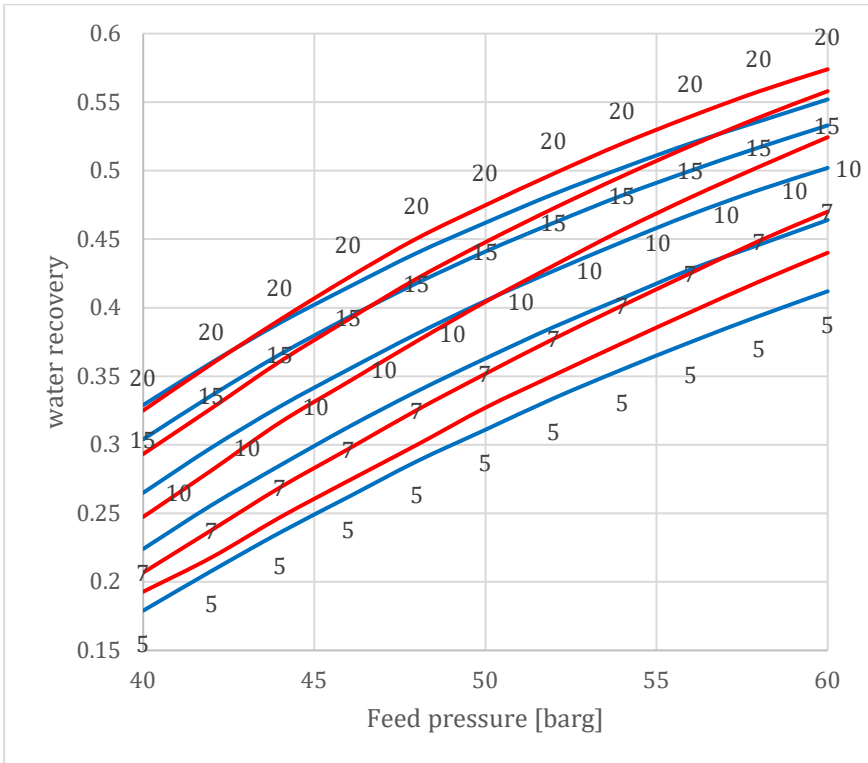


Figure 22. Simulation water recovery results comparison. WAVE in blue and Mathematica model in red.

From the water recovery simulation data, it seems that for lower PV count, that is, less total active surface area and higher permeate fluxes, the developed Mathematica model underestimates the system performance, but at higher PV count, the opposite happens, and the Mathematica model gives higher water recovery results, specially at higher operating pressures. Overall, the agreement on simulation results with WAVE is best in the 45 to 55 barg range, which coincidently is near the usual operating range for SWRO systems.

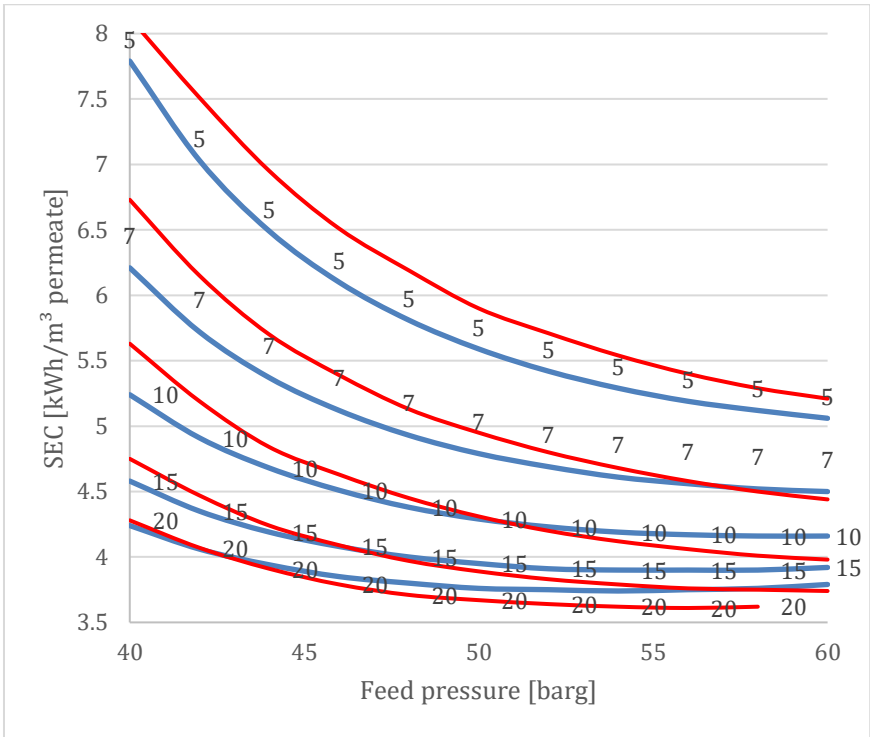


Figure 23. Simulation SEC results comparison. WAVE in blue and Mathematica model in red.

Comparing specifically the SEC results, which in section 3.4 was decided that was the most important performance metric, an agreement on average of 96.65% is observed for the SEC values taking into account all the simulated range of pressures. Considering only a more realistic operation range of 50 to 60 barg, the results agreement increases to 97.55%

5.2. TEMPERATURE VARIABLE

When it comes to simulating the effect of temperature on the system, a clear difference exists between WAVE and MATHEMATICA's results. As explained in the limitations section of the MATHEMATICA model, temperature is only taken into account for the osmotic pressure calculation, and not for the diffusion constants. The diffusion constants for both water and salt depend extensively on operating temperature.

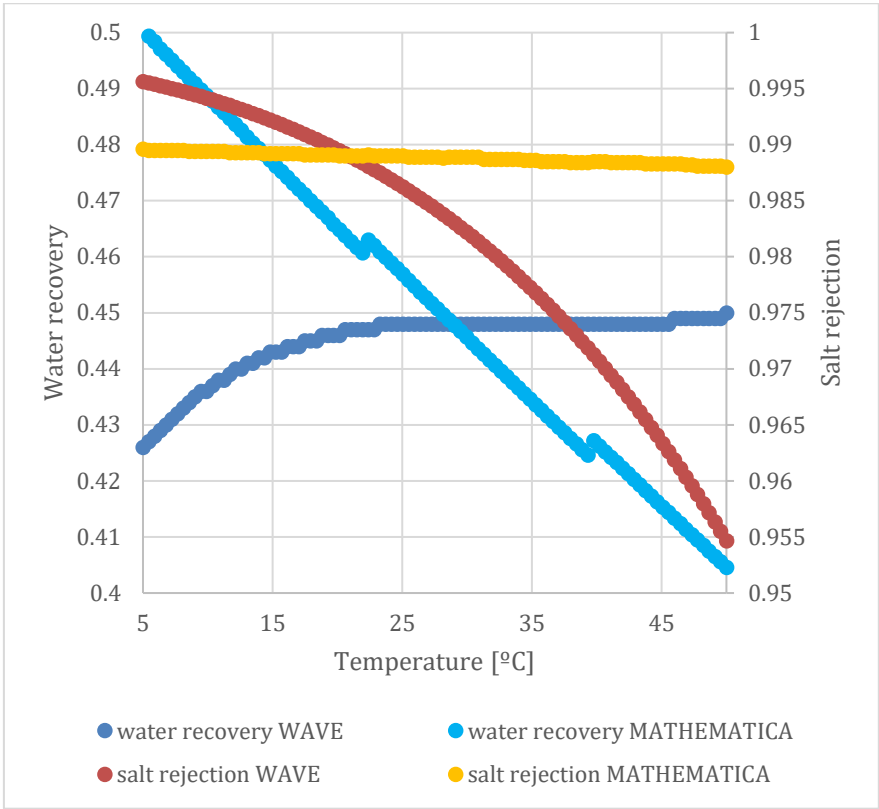


Figure 24. Simulation water recovery results comparison with temperature variation,

10PVx6 Seamaxx at 54 bar, 100m3/h 35g/L TDS feed

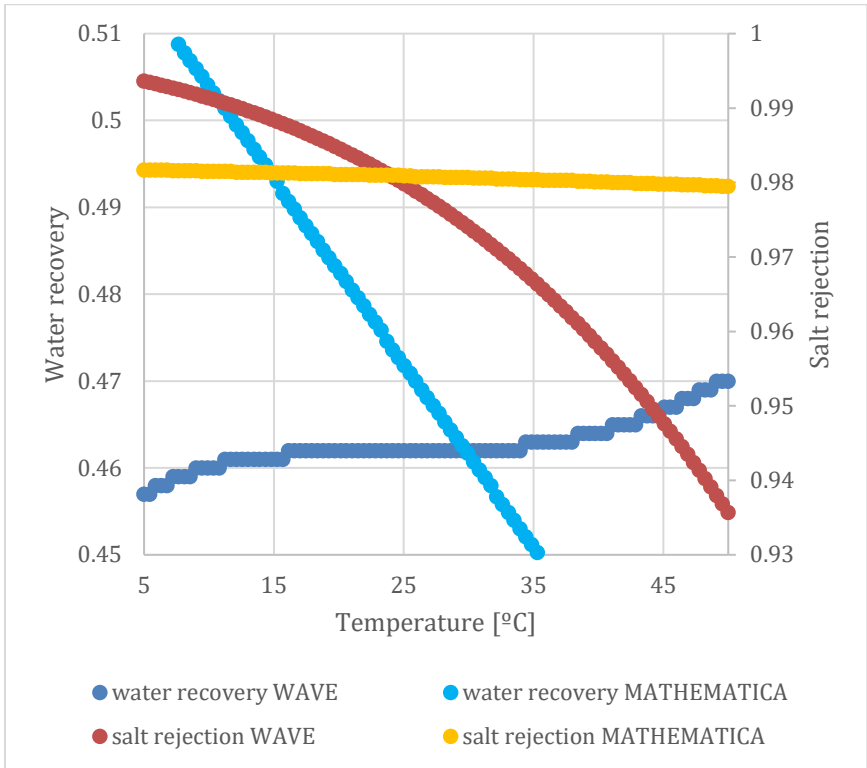


Figure 25. Simulation water recovery results comparison with temperature variation,

15PVx6 Seamaxx at 52 bar, 100m³/h 35g/L TDS feed

As can be seen in figure 13 and 14, the MATHEMATICA model does not accurately represent the drastic effect of high temperatures when it comes to salt rejection, and assumes that temperature affects water recovery linearly, when according to WAVE, it's constant within the recommended operating temperature range.

Given this limitation, the user must input the corresponding water and salt diffusivity constants at the desired operating temperature.

6. OPTIMIZATION OF A LSRRO SYSTEM

For 1-stage RO systems, given some fixed feed stream and existing membranes (defined A and B parameters), the only optimizable process variables to minimize SEC are applied hydraulic pressure, and PV number, which are a discrete way of increasing or decreasing total active membrane surface area. [17]

SEC does not present a mathematical optimum with respect to these variables; instead, an optimum based on the LCOW is to be determined, which takes into consideration membrane module costs, labor, maintenance and replacement rates, etc. In any case, this cost optimization is outside the scope of this work.

However, for 2-stage LSRRO, the SEC can be mathematically optimized with regards to the second stage's parameters, which in practice, are made to specification during the membrane refurbishment.

6.1. DEFINITION OF THE SYSTEM

From the simulation data of the previous section, for a 1 stage system, 15 PV seems like a good compromise between energy efficiency (related to operational costs) and initial capital expense, as increasing the number of PVs to 20, for example, brings only a marginal decrease in SEC. 15 PVs also results in an average water flux density of 18.3 LMH, which falls under the recommended range of 15 to 20 LMH for this specific equipment. [18]

For this reason, 15 PVs is going to be fixed for the first stage (2453 m²), and the optimization will revolve around finding an optimum surface area ratio for the second stage

$$S_{1st\ stage} : S_{2nd\ stage} \quad (32)$$

The other parameters to optimize are the A and B values. To remove a degree of freedom, the B value is estimated from an experimental relationship to the A value, which applies to polyamide membranes. [19]

$$B = 0.0133 \cdot A^3 \quad (33)$$

The second stage permeability will be evaluated in relation to the first stage, which is composed of the same Seamaxx membrane elements used in the previous section, of 1.3 LMH/bar water permeability and 0.2 LMH salt permeability.

However, the literature also suggests that new polyamide TFC membranes are deviating from the proposed water and salt permeability relationship presented in equation 33, notably the membranes with high permeability used un LSRRO, so an optimization of both A and B parameters independently from each other might have to be studied further. [19]

6.2. SIMULATION RESULTS

With a 100 m³/h 35 g/L feed, for a surface ratio of 2:1, the following results are obtained:

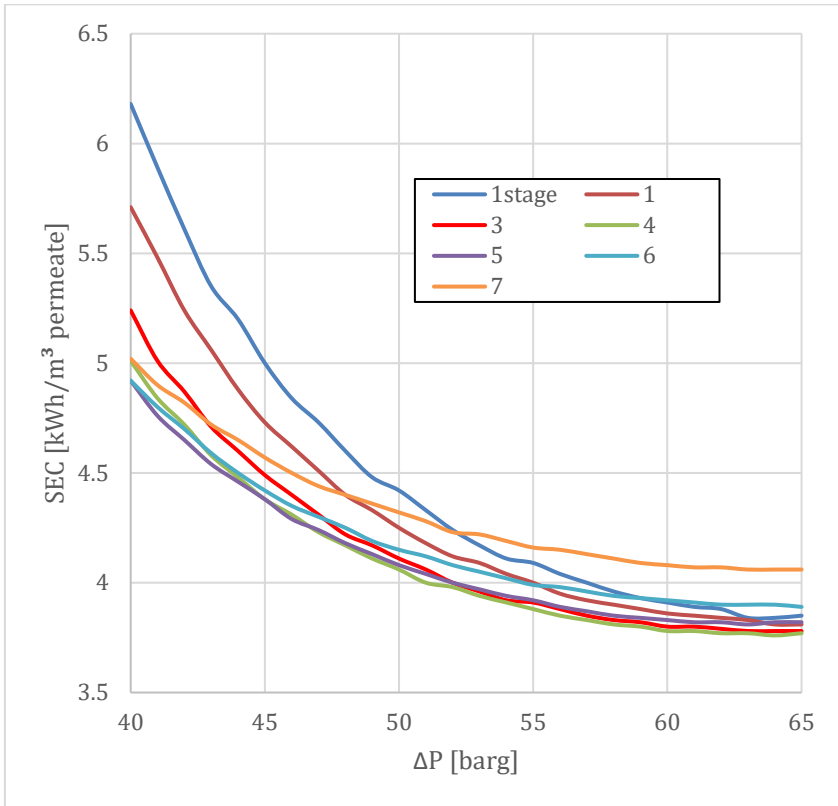


Figure 26. Surface area relation 2:1 SEC simulation results for several second stage permeabilities across the operation range.

As can be seen in the graph, figure 26, a considerable decrease in SEC can be achieved with the added LSRRO stage, especially at lower operating pressures. However, after a certain point, by further increasing the second stage's water permeability the SEC increases. This means that an optimum must exist.

Focusing on the range of interest of operation range, 50 to 65 barg, and showing only the candidates for the optimum:

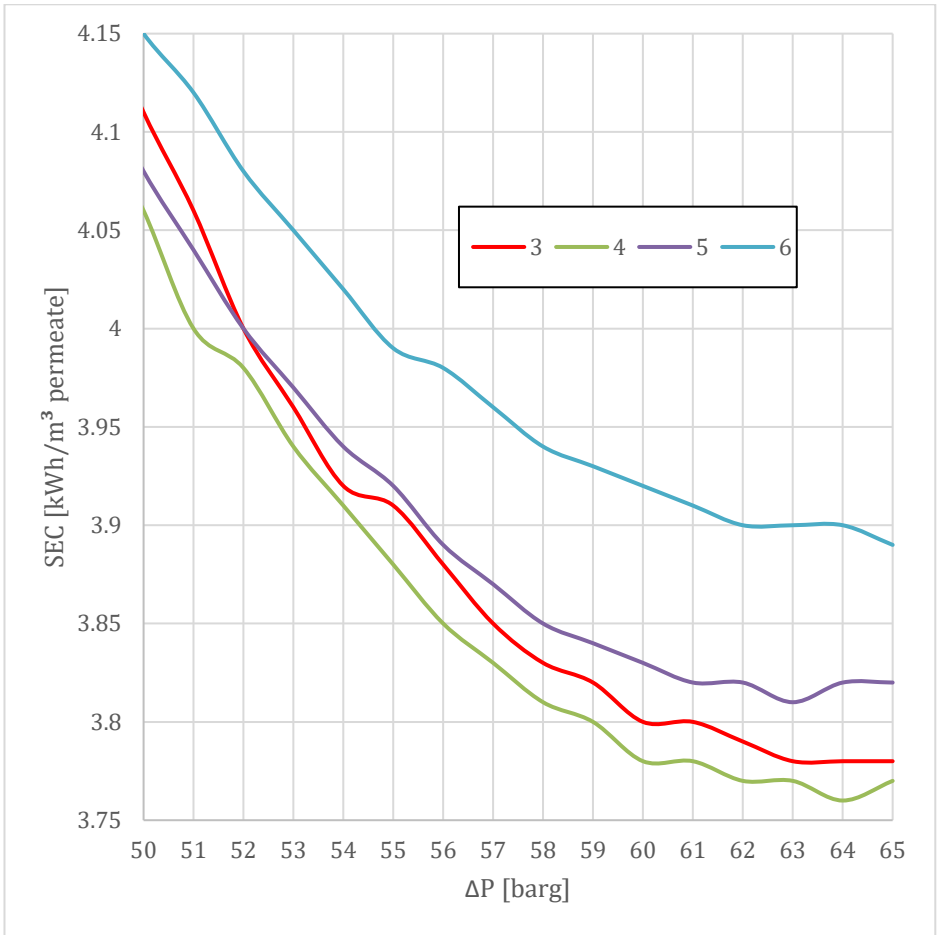


Figure 27. Surface area relation 2:1 second stage permeability optimization for minimum SEC.

The optimum second stage permeability in this configuration is about 4 times the first stage. When operating at a reasonable 60 barg, the point at which the curve starts flattens out, the

energy saving totals to 0.13 kW/m³ permeate, so 2.8%, when compared to a single stage system.

If the system were to operate at lower pressures, the overall SEC would be larger, but the difference in SEC between the single stage RO and the two stage LSRRO would be more considerable. For example, for a system operating at 55 barg, the energy savings would be 0.21 kW/m³ permeate.

For a 3:1 surface area relation and the same feed conditions of 100 m³/h and 35 g/L TDS, similar results are obtained:

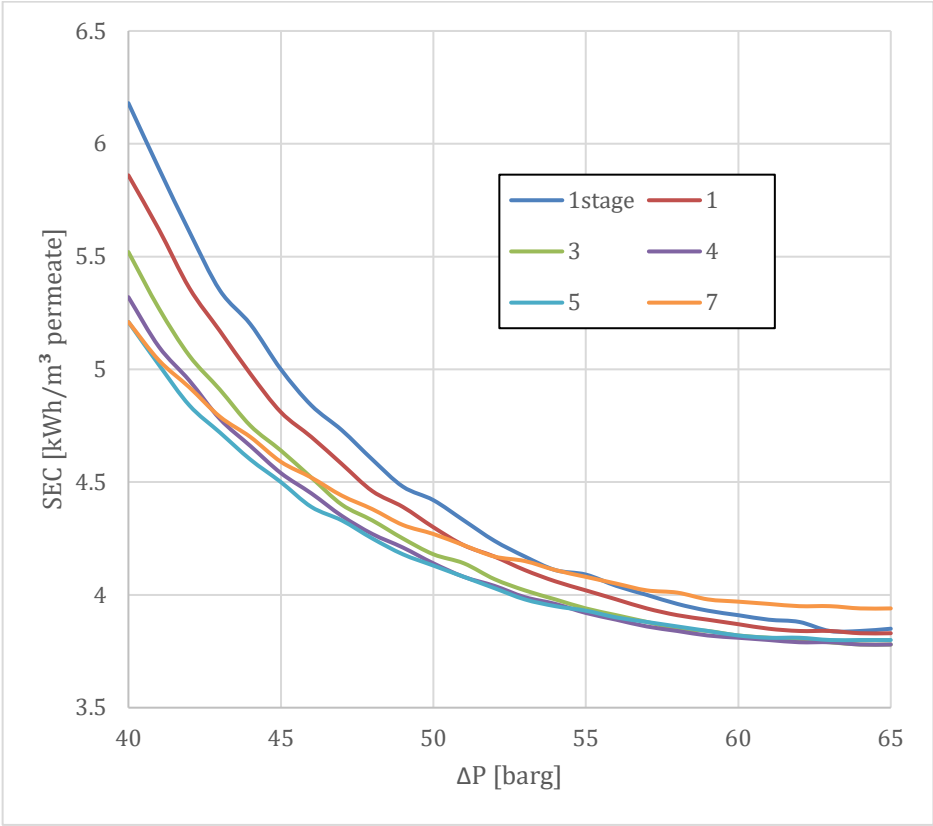


Figure 28. Surface area relation 3:1 SEC simulation results for several second stage permeabilities across the operation range.

Narrowing down the search for the optimum, it leaves the following candidates for optimum:

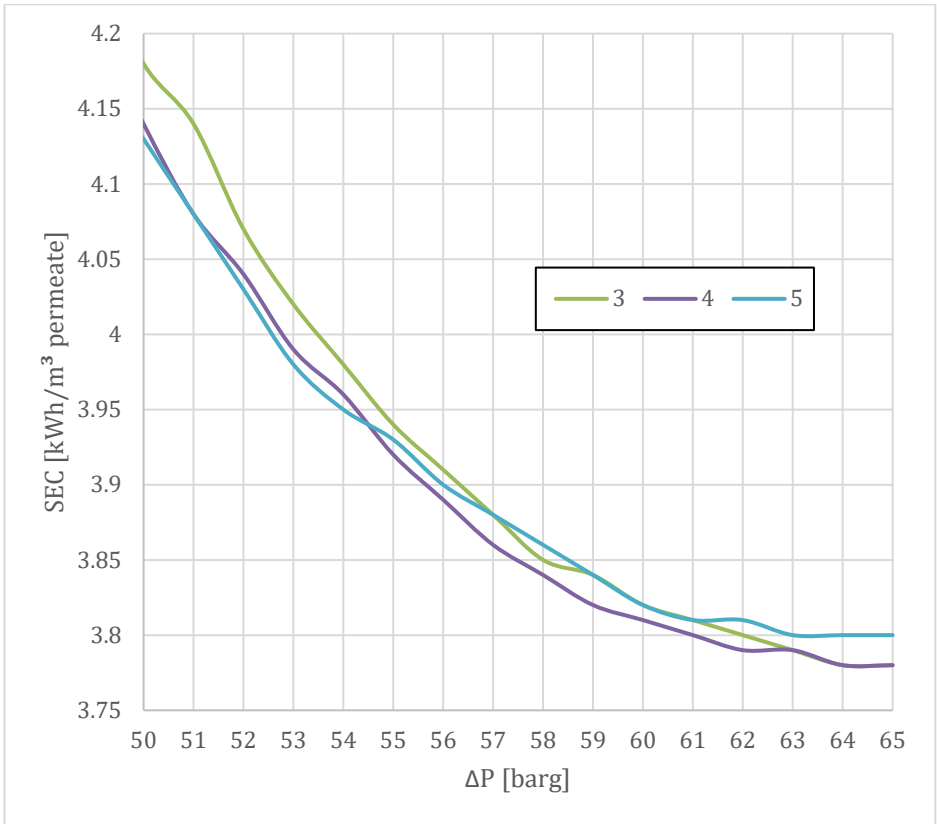


Figure 29. Surface area relation 3:1 second stage permeability optimization for minimum SEC.

The optimum water permeability is 4 again. However, in this case, the optimum is very close to 3 and 5. In this case, the increase in system energy efficiency is 2.6% when compared to a single stage setup.

Note that the precision of the reported values of SEC with the Mathematica model is 0.01, as that accounts for small variations in the results given the same input, improving overall reproducibility. These variations are due to different starting values for the numerical solving methods. In figure 29 and later in figure 31 this drawback is made clear, as the curve has a step-like appearance, with values also landing exactly on top of each other.

For a 5:1 surface area relation, with the same feed conditions as the previous cases, yet again similar results are obtained:

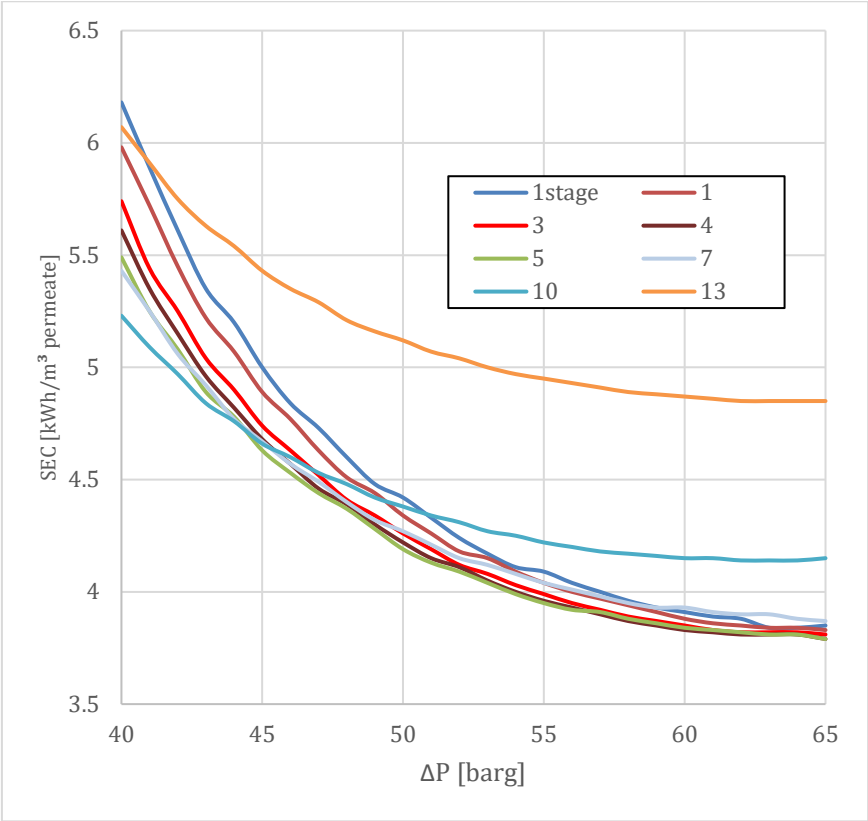


Figure 30. Surface area relation 5:1 SEC simulation results for several second stage permeabilities across the operation range.

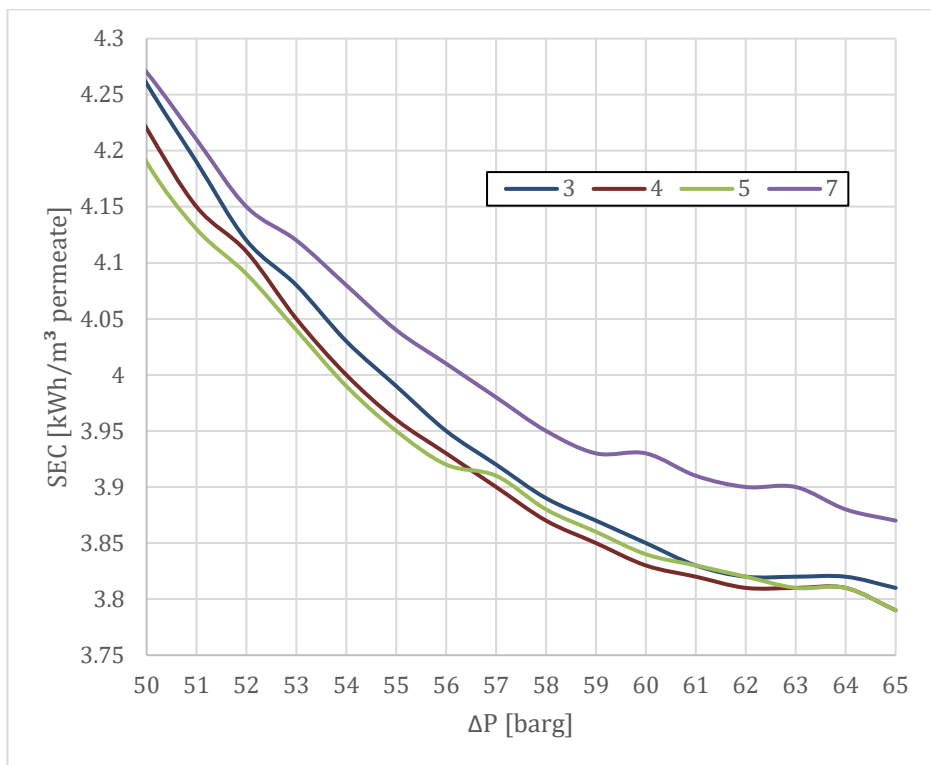


Figure 31. Surface area relation 5:1 second stage permeability optimization for minimum SEC.

The optimum seems to be between 4 times the permeability of the first stage, at least for higher operating pressures. In this case, the optimum is not as distinctive. Here, the total SEC reduction when compared with a single stage system is just 2.0%.

6.3. RESULTS DISCUSSION

For the simulated Seamaxx membranes, which normally have a water permeability of 1.3 LMH/bar, using the found optimum relation, the second stage should have a water permeability of 5.2 LMH/bar.

Recent literature, in which very similar conditions are simulated (also 35 g/L TDS), for a first stage with 1.51 LMH/bar water permeability, it was found that the second stage would optimally

have to be of 10.62 LMH/bar water permeability. In this case, the tradeoff constraint in equation 33 was also used. [19]

Comparing the three simulated cases of different surface area relation, it appears that the optimum permeability of the second stages does in fact barely depend on this variable. For smaller second stages, as can be seen in figure 31, using the optimum second stage permeability doesn't have as much effect on the SEC.

In addition, it seems that the smaller the second stage, the lesser energy efficiency increase with respect to the single stage. This brings up a compromise with capital investment costs and profitability over time, as larger second stages are the ones improving SEC the most.

However, it still is very clear that in any case, no major cost savings are to be found in implementing LSRRO for SWRO needs. Even in the best case scenario simulated, 2.8% SEC decrease is not satisfactory enough to validate an investment in the LSRRO stage, as it would require a very long period to pay off.

In a similar manner, 2-stage non-LSRRO configurations for SWRO have also been known to operate at higher SEC, in order to minimize concentrate water discharge, which is sometimes preferable. [24]

6.4. FUTURE RESEARCH PROPOSALS

It remains to be tested with this Mathematica model if feed TDS really has any effect on these optimums and potential SEC reduction. For example, lower feed TDS requires overall less applied pressure, as the osmotic pressures that have to be overcome are lower.

As observed in figures 26, 28 and 30, it's in the lower pressure range where these LSRRO systems really improve SEC, so further research with simulation should target these cases.

Some literature indicates that LSRRO is in fact particularly more energy efficient and viable in treating lower concentration streams such as those found in brackish water, which range from 5 to 30 g/L TDS. [20][25]

Nevertheless, the data gathered in this work can still be used to more easily find the optimum values, as the general behavior of the system has been determined from further analyzing the flowrates of all the currents.

It seems that if the second stage's water permeability is too low, then not much energy is saved, because the remaining pressure of the first stage's concentrate stream won't be high enough to overcome its own osmotic pressure, which is higher than the first stage's as the concentrate is always more concentrated than the feed. Overall, this results in only a very small permeate recirculation.

On the other hand, if the second stage's water permeability is too high (too loose), with the pressure of the concentrated stream most of the water will permeate, giving very large permeate recirculation. Because this permeate needs to be pumped again, a lot of energy is wasted. Therefore, when refurbishing a membrane, care must be taken not to over-do the oxidation process, as that will give unusable membranes.

7. CONCLUSIONS

Development of more energy efficient reverse osmosis systems is crucial to the development of a sustainable future, and low-salt-rejection reverse osmosis is an option to be considered for this purpose. However, currently available commercial specialized software is unable to simulate these complex systems. Hence a new program using Wolfram Mathematica's functionality was created from the ground up, to address these needs.

This program has been tested with great success in agreeing with DuPont's WAVE software, the most prevalent RO simulation software, on average producing results within 97% of one another. In addition, the program has been made to have a very intuitive and user-friendly UI.

Putting in use the produced program, simulations have been run to find optimum parameters for a LSRRO system which minimize SEC. It has been found that for SWRO systems, the second stage's water permeability must be 4 times the one in the first stage in order to reduce energy consumption by 2% in 5:1 stage size system increasing to 2.8% with 2:1 stage size systems.

It is proposed that the research in this field following this study focuses on optimizing for cases with lower feed salt concentrations, such as with brackish water, where it seems that the addition of the LSRRO stage can be much more financially appealing.

REFERENCES AND NOTES

1. United Nations. Sustainable Development Goals. <https://www.un.org> (accessed May 25, 2024)
2. Catalan Water Agency. Desalination. <https://aca.gencat.cat> (accessed May 25, 2024)
3. Glater, J. (1998). "The early history of reverse osmosis membrane development". *Desalination*. 117 (1–3): 297–309.
4. Weintraub, Bob (December 2001). "Sidney Loeb, Co-Inventor of Practical Reverse Osmosis". *Bulletin of the Israel Chemical Society* (8): 8–9.
5. Cadotte, John E. (1981) "Interfacially synthesized reverse osmosis membrane" U.S. patent 4,277,344
6. Jones, Edward; et al. (20 March 2019). "The state of desalination and brine production: A global outlook". *Science of the Total Environment*. 657: 1343–1356.
7. Ritt, C.L., Stassin, T., Davenport, D.M., DuChanois, R.M., Nulens, I., Yang, Z., Ben-Zvi, A., Segev-Mark, N., Elimelech, M., Tang, C.Y., Ramon, G.Z., Vankelecom, I.F.J., Verbeke, R. (2022). The open membrane database: Synthesis–structure–performance relationships of reverse osmosis membranes. *Journal of Membrane Science*, 641, 119927
8. Lennotech. Reverse Osmosis Demineralization. <https://www.lennotech.com> (accessed Apr 12, 2024)
9. Yuhao Du, Zhangxin Wang, Nathaniel J. Cooper, Jack Gilron, Menachem Elimelech. Module-scale analysis of low-salt-rejection reverse osmosis: Design guidelines and system performance, *Water Research*, Volume 209, 2022, 117936
10. Chen, C.; Qin, H. A Mathematical Modeling of the Reverse Osmosis Concentration Process of a Glucose Solution. *Processes* 2019, 7, 271.
11. Hoek, E.M.V., Weigand, T.M. & Edalat, A. Reverse osmosis membrane biofouling: causes, consequences and countermeasures. *npj Clean Water* 5, 45 (2022).
12. Puretec Industrial Water. The basis of reverse osmosis. <https://puretecwater.com> (accessed Mar 12, 2024)
13. Veolia Water Technologies and Solutions. Winflows. <https://www.watertechnologies.com> (accessed Apr 8, 2024)
14. National Alliance for Water Innovation. WaterTap. <https://www.nawihub.org> (accessed Apr 8, 2024)
15. EcosimPro. ROSIM. <https://www.ecosimpro.com> (accessed Apr 8, 2024)
16. Water Desalination. Texas Water Development Board. 1004831107.
17. US Department of Energy. Reverse Osmosis Optimization. <https://www.energy.gov> (accessed Mar 20, 2024)
18. DuPont. Resource center. <https://www.dupont.com> (accessed May 2, 2024)
19. Adam A. Atia, Jeff Allen, Ethan Young, Ben Knueven, Timothy V. Bartholomew. Cost optimization of low-salt-rejection reverse osmosis. *Desalination*, Volume 551, 2023, 116407
20. Haoqi Zhao, Zhangxin Wang, Yuanmiaoliang Chen. A theoretical analysis on upgrading desalination plants with low-salt-rejection reverse osmosis. *Desalination*, Volume 565, 2023, 116827
21. Zhangxin Wang, Akshay Deshmukh, Yuhao Du, Menachem Elimelech. Minimal and zero liquid discharge with reverse osmosis using low-salt-rejection membranes, *Water Research*, Volume 170, 2020, 115317
22. Yip, N.Y., Elimelech, M., 2011. Performance Limiting Effects in Power Generation from Salinity Gradients by Pressure Retarded Osmosis. *Environmental Science & Technology* 45, 10273–10282.

-
23. Delft University of Technology. Water treatment, nanofiltration and reverse osmosis. <https://ocw.tudelft.nl> (accessed Mar 12, 2024)
 24. Kim, J., Park, K., & Hong, S. (2020). Optimization of two-stage seawater reverse osmosis membrane processes with practical design aspects for improving energy efficiency. *Journal of Membrane Science*, 601, 1-11. Article 117889.
 25. Song, L., Schuetze, B., Rainwater, K. (2012). Demonstration of a High Recovery and Energy Efficient RO System for Small-Scale Brackish

ACRONYMS

RO: reverse osmosis

LSRRO: low salt rejection reverse osmosis

SWRO: sea water reverse osmosis

PV: pressure vessel

CP: concentration polarization

CPF: concentration polarization modulus factor

CFV: cross-flow velocity

TFC: thin film composite

ERD: energy recovery device.

SEC: specific energy consumption

WAVE: water application value engine

UI: user interface

OS: operating system

TDS: total dissolved solids

CIP: clean in place

KPI: key performance indicator

LMH: $\text{L} \cdot \text{m}^{-2} \cdot \text{h}^{-1}$

LCOW: Levelized cost of water

SYMBOLS

T: temperature [K]

R: ideal gas constant [$\text{L}\cdot\text{bar}\cdot\text{mol}^{-1}\cdot\text{K}^{-1}$]

A: water permeability [LMH/bar]

B: salt permeability [LMH]

J_w : water flux [LMH]

J_s : salt flux [$\text{kg}\cdot\text{m}^{-2}\cdot\text{h}^{-1}$]

c: salt concentration [g/L] or [kg/m^3]

w: stream flowrate [m^3/h]

P: pressure [barg]

k_F : water mass transfer coefficient [LMH]

Sh: Sherwood number [-]

Re: Reynolds number [-]

Sc: Schmidt number [-]

ρ : density [kg/m^3]

μ : viscosity [$\text{Pa}\cdot\text{s}$]

u : cross-flow velocity [m/s]

d_h : hydraulic diameter [m]

D: diffusivity [m^2/s]

APPENDICES

APPENDIX 1: USER MANUAL

A few warnings and instructions need to be given for what is otherwise a rather simple and intuitive program.

Program installation:

The software needed to run this program is free to download and use without account creation or time restrictions. The official download is available through <https://www.wolfram.com/player/>

Please select the option matching your OS (Windows, macOS, or Linux).



Figure 32. Wolfram Player download page.

Please follow the onscreen instructions for the installation of the program on your computer. Be aware that about 5.1 GB of free storage in your installation drive is needed.

Once the Wolfram Player is installed, you should be able to double-click .cdf files and open them automatically. If that does not work, please open Wolfram Player application manually, select Open..., and choose the .cdf file.

Important simulation instruction:

It is needed to run a simulation using 1 element of discretization before increasing this value further, or else the simulation might crash or give inaccurate results.

This is especially true when switching from a 1-stage system setup to a 2-stage or 3-stage. The simulation must be run using the “1 element of discretization” option first, and it is recommended to set it before switching the number of stages.

On the other hand, this requirement is not necessary if the next simulation is very similar, for example, when increasing the pressure from 50 barg to 55 barg.

Simulation stuck or looping between two values:

The program is struggling to converge to a single result. It is recommended to slightly change a variable to force it to land on a single value.

If the program gets completely stuck and stops responding to user inputs, please wait about a minute. If the issue persists, the only solution would be to exit the program and restart it.

Simulation errors:

Three different error modes can be encountered. These occur when simulating extreme cases, due to failure to converge or internal overflowing.

Streams as {flow rate[m ³ /h], TDS[g/L]}			
stage	1-feed	2-permeate	3-concentrate
1	{Round[w[1, 1, 1] /. {}][1], 0.0001], Round[c[1, 1, 1] /. {}][1], 0.0001]}	{Round[w[1, 1, 2] /. {}][1], 0.0001], Round[c[1, 1, 2] /. {}][1], 0.0001]}	{Round[w[1, 1, 3] /. {}][1], 0.0001], Round[c[1, 1, 3] /. {}][1], 0.0001]}
2	{Round[w[2, 1, 1] /. {}][1], 0.0001], Round[c[2, 1, 1] /. {}][1], 0.0001]}	{Round[w[2, 1, 2] /. {}][1], 0.0001], Round[c[2, 1, 2] /. {}][1], 0.0001]}	{Round[w[2, 1, 3] /. {}][1], 0.0001], Round[c[2, 1, 3] /. {}][1], 0.0001]}
Power consumption [kW]: Round[3.48092 (w[1, 1, 1] /. {})[1]), 0.01]			
SEC [kWh/m ³]: Round[Round[3.48092 (w[1, 1, 1] /. {})[1]), 0.01] / (w[1, 1, 2] /. {})[1]), 0.01]			
Net Power consumption (w/ ERD) [kW]: Round[3.48092 (w[1, 1, 1] /. {})[1]), 0.01] - Round[2.77778 (w[2, 1, 3] /. {})[1]), 0.01]			
SEC [kWh/m ³]: Round[(Round[3.48092 (w[1, 1, 1] /. {})[1]), 0.01] - Round[2.77778 (w[2, 1, 3] /. {})[1]), 0.01] / (w[1, 1, 2] /. {})[1]), 0.01]			
Observed salt rejection: Round[1 - 0.0285714 (c[1, 1, 2] /. {})[1]), 0.0001]			
water recovery: Round[0.01 (w[1, 1, 2] /. {})[1]), 0.0001]			

Figure 33. Results UI when a failure to converge occurs.

This error is encountered when suddenly high applied pressure is selected within the 1 element of discretization mode. The user can easily recover from this error by changing the inputs so that the simulation deals with more realistic cases.

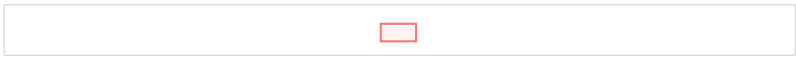


Figure 34. Results UI when the program crashes.

This error appears mainly when, while the error of the previous figure is on screen, the user switches to a larger than 1 element discretization mode. The program has crashed and the user cannot recover from this situation, and the only remediation is to close and restart the program.

Streams as {flow rate[m³/h], TDS[g/L]}

stage	1-feed	2-permeate	3-concentrate
1	{102.196, 43.903}	{52.5784, 0.9777}	{65.8672, 65.8769}
2	{55.1614, 80.4892}	{44.1233, 12.8878}	{31.6298, 151.906}

Figure 35. Results UI displaying nonsensical data.

While not as apparent as the previous errors, the user must be careful when conducting extreme case simulations in an over 1 element of discretization mode involving high pressure and high secondary stages surface area. Depending on the starting values, the program will fail to converge to an accurate solution, but will give its closest estimate. For the example figure above, note that the system mass balance is not satisfied, among other equations.

$$100 \neq 52.5784 + 31.6298$$

The user might be able to tinker with the inputs to help the program find a good starting value and converge, but it's very likely that an overflow error will be encountered along the way, leading to the same error screen as in the previous section.

Other recommendations:

Simulation results do not save between sessions, when the program is exited, all inputted information is lost. It is recommended to use the Ctrl+P keyboard shortcut to bring up the Print menu and save a screenshot from there.

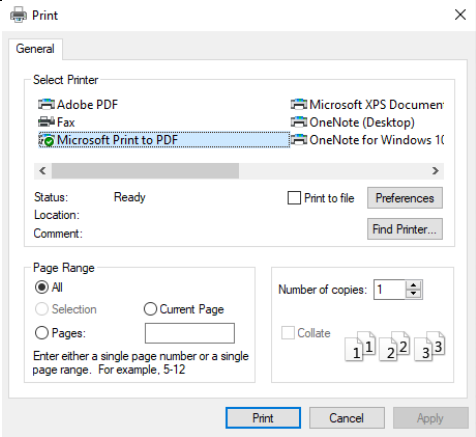


Figure 36. Wolfram Player print screen

Accessibility option:

The user might find the text too small and difficult to read. An option to increase the zoom is available, but beware that then, the UI might not fully fit in the screen. In this case, use the sidebar scrollers to navigate the UI.

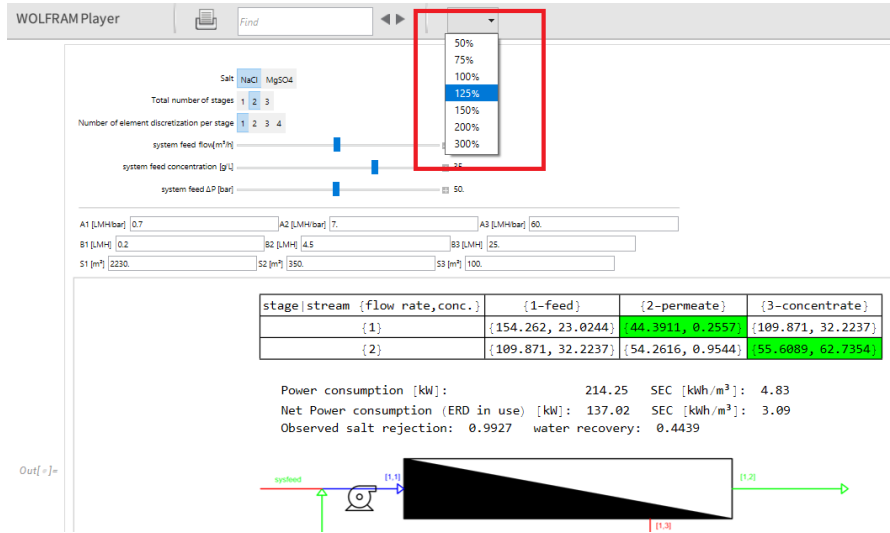


Figure 37. Wolfram Player zoom option.

APPENDIX 2: MATHEMATICA CODE FOR THE SIMULATION PROGRAM

The code is available for viewing to any user of the Wolfram Player, by double clicking the cell brackets to the right of the program. However, to modify said code, a Wolfram Mathematica license is needed.

The whole UI for the program is built around Mathematica's Manipulate functionality. This allows for a simple yet very customizable UI.

Note that the first instructions to run are at the end of the code, in an Initialization option, which in this case contains the values for the constants used and the images for the diagram display.

```
In[ ]:= Manipulate[
  Posm[c_] = c R (T + T0) v h / m mass;
  w[0, Nelements, 3] = wsysfeed; c[0, Nelements, 3] = csysfeed;
  A[1] = A1; A[2] = A2; A[3] = A3;
  B[1] = B1; B[2] = B2; B[3] = B3;
  S[1] = S1; S[2] = S2; S[3] = S3;
  Table[P[i, j] = ΔP, {i, 1, Nstages}, {j, 1, Nelements}];
  CPFTs;

  (*=====Mathematical model=====*)

  eqA = Table[w[i, j] == A[i] (P[i, j] - (Posm[CPF[i, j] × c[i, j, 1]] - Posm[c[i, j, 2]])),
    {i, 1, Nstages}, {j, 1, Nelements}];
  eqB = Table[S[i, j] == B[i] (CPF[i, j] × c[i, j, 1] - c[i, j, 2]), {i, 1, Nstages}, {j, 1, Nelements}];

  eqSA = Table[w[i, j, 2] == S[i] × w[i, j] / 1000 / Nelements, {i, 1, Nstages}, {j, 1, Nelements}];
  eqSB = Table[w[i, j, 2] × c[i, j, 2] == S[i] × S[i, j] / 1000 / Nelements, {i, 1, Nstages}, {j, 1, Nelements}];

  BMstage =
  Table[{w[i, j, 1] == w[i, j, 2] + w[i, j, 3],
    w[i, j, 1] × c[i, j, 1] == w[i, j, 2] × c[i, j, 2] + w[i, j, 3] × c[i, j, 3]}, {i, 1, Nstages}, {j, 1, Nelements}];
  BMnode = Table[{
    If[j == 1, If[i == Nstages, w[i, j, 1] == w[i - 1, Nelements, 3],
      w[i, j, 1] == w[i - 1, Nelements, 3] + Sum[w[i + 1, j, 2], {j, 1, Nelements}]], w[i, j, 1] == w[i, j - 1, 3]],
    If[j == 1, If[i == Nstages, c[i, j, 1] == c[i - 1, Nelements, 3],
      w[i, j, 1] × c[i, j, 1] == w[i - 1, Nelements, 3] × c[i - 1, Nelements, 3] +
      Sum[w[i + 1, j, 2] × c[i + 1, j, 2], {j, 1, Nelements}], c[i, j, 1] == c[i, j - 1, 3]]], {i, 1, Nstages},
    {j, 1, Nelements}];
```

The mathematical model contains all the equations specified in section 4.2. They have been coded in a very generalized manner, so that an arbitrary number of stages and elements can be used. The system of equations produced has eight equations per stage and eight equations per

element of discretization, so for example, to simulate a 2-stage system with 2 elements of discretization, 32 equations need to be simultaneously solved in each iteration. While the number of equations increases linearly with the number of stages and elements, it must be noted that the computation time does not.

```
(*=====Solving=====*)
listeqs = Flatten[Join[eqA, eqB, eqSA, eqSB, BMstage, BMnode]];
listvars = Flatten[Join[Table[{w[i, j, k], c[i, j, k]}, {i, 1, Nstages}, {j, 1, Nelements}, {k, 1, 3}],
  Table[{Jw[i, j], Js[i, j]}, {i, 1, Nstages}, {j, 1, Nelements}]]];

listvarmem =
Join[Flatten[Table[{w[i, j, k], ws[i, 1, k], 0, 10000}, {c[i, j, k], cs[i, 1, k], 0, 10000}],
  {i, 1, Nstages}, {j, 1, Nelements}, {k, 1, 3}], 3],
  Flatten[Table[{Jw[i, j], Jws[i, 1], 0, 10000}, {Js[i, j], Jss[i, 1], 0, 10000}], {i, 1, Nstages},
    {j, 1, Nelements}], 2]];

SetAttributes[{w, c, Jw, Js}, NHoldAll];

xx1 = If[Nelements == 1, Quiet[NSolve[listeqs, listvars, PositiveReals, WorkingPrecision -> 8]],
  Quiet[FindRoot[SetPrecision[listeqs, acc], listvarmem, WorkingPrecision -> acc, AccuracyGoal -> acc,
    PrecisionGoal -> acc, MaxIterations -> 100, Method -> "AffineCovariantNewton"]]]];
```

The system of non-linear equations is solved, following the pseudocode in fig 11. The solving functions are wrapped in Quiet functions, which suppress warning and error messages that would otherwise clutter the final UI.

```
(*=====Solution extraction=====*)
Quiet[Table[{Jws[i, j] = Jw[i, j] /. xx1[[1]], Jss[i, j] = Js[i, j] /. xx1[[1]], {i, 1, Nstages},
  {j, 1, Nelements}]]];
Quiet[Table[{ws[i, j, k] = w[i, j, k] /. xx1[[1]], cs[i, j, k] = c[i, j, k] /. xx1[[1]], {i, 1, Nstages},
  {j, 1, Nelements}, {k, 1, 3}]]];
Table[ws[i, Nelements + 1, 2] = Sum[ws[i, j, 2], {j, 1, Nelements}], {i, 1, Nstages}];
Table[cs[i, Nelements + 1, 2] = Sum[ws[i, j, 2] × cs[i, j, 2], {j, 1, Nelements}] / ws[i, Nelements + 1, 2],
  {i, 1, Nstages}];

(*=====CPF check=====*)
Table[If[(1 < Exp[Jws[i, j] / kf] < 1.9) && (Abs[Exp[Jws[i, j] / kf] - CPF[i, j]] > 0.005),
  CPF[i, j] = Exp[Jws[i, j] / kf]], {i, 1, Nstages}, {j, 1, Nelements}];
CPFts = CPF[Nstages, Nelements];
```

The CPF iteration loop is performed according to the pseudocode in fig 12. Note that the variable CPFts is auxiliary and used to keep the loop running so long as it gets updated. The option tracked symbols for the Manipulate function checks if this variable has been updated, and if that's the case, runs the code again.

```
(*=====KPI=====*)
powercons = Round[(ws[1, 1, 1] × P[1, 1] + If[Nstages == 3, ws[3, 1, 1] × P[3, 1], 0]) 100 / 3600 / pumpeff / motoreff,
0.01];
ERDpowergen = Round[ws[Nstages, Nelements, 3] × P[Nstages, Nelements] 100 / 3600 ERDeff, 0.01];
netpowercons = powercons - ERDpowergen;

SEC = Round[powercons / ws[1, Nelements + 1, 2], 0.01];
SECDERD = Round[netpowercons / ws[1, Nelements + 1, 2], 0.01];
Rsobs = Round[1 - cs[1, Nelements + 1, 2] / csysfeed, 0.0001];
rec = Round[ws[1, Nelements + 1, 2] / wsysfeed, 0.0001];
```

KPI are defined as per section 4.4, taking into consideration equipment efficiencies.

```
(*=====Graphics=====*)
headings = {Table[i, {i, 1, Nstages}], {"1-feed", "2-permeate", "3-concentrate"}};

data =
Round[Table[{ws[i, 1, 1], cs[i, 1, 1]}, {ws[i, Nelements + 1, 2], cs[i, Nelements + 1, 2]},
{ws[i, Nelements, 3], cs[i, Nelements, 3]}], {i, 1, Nstages}], 0.0001];

textoverlay = List[Text["feed", {20, 600}], Text["permeate", {900, 600}], Text["concentrate", {900, 300}]];
wvalueoverlay = List[Text[w[0, Nelements, 3], {20, 580}], Text[ws[1, Nelements + 1, 2], {900, 580}],
Text[ws[Nstages, Nelements, 3], {900, 280}]];
cvalueoverlay = List[Text[c[0, Nelements, 3], {20, 560}], Text[cs[1, Nelements + 1, 2], {900, 560}],
Text[cs[Nstages, Nelements, 3], {900, 260}]];

Column[{
Row[{"Streams as {flow rate[m³/h], TDS[g/L]}"},
Grid[Join[Transpose[{Join[{"stage"}, headings[[1]]}], Join[{headings[[2]], data}, 2],
Frame → All, Background → {None, None, {{2, 3} → Green, {Nstages + 1, 4} → Green}},
Row[{}],
Grid[{"Power consumption [kW]:", powercons, "SEC [kWh/m³]: ", SEC,
{"Net Power consumption (w/ ERD) [kW]: ", netpowercons, "SEC [kWh/m³]:", SECDERD},
{"Observed salt rejection:", Rsobs, "water recovery:", rec}], Alignment → ":",
Row[{}],
Show[bdiag[Nstages], Graphics[textoverlay], Graphics[wvalueoverlay], Graphics[cvalueoverlay],
ImageSize → 600]
}],
```

The UI graphics for the result screen are built from nested column, row and grid elements, so that they can be properly spaced and aligned.

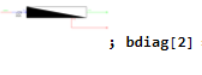
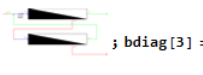

For the system diagram, the text is also overlayed here, using the coordinates of an average of the positions in the three possible diagrams. Unfortunately, the text can't be fixed to the image, so when the UI is resized to different screen sizes and ratios, it misbehaves and moves on top of the diagram graphics.

```

(*=====Control options=====*)
Control[{{T, 25, "Temperature [°C]"}, ControlType → InputField}},
Control[{{mmass, 58.44, "Salt"}, {58.44 → "NaCl", 120.37 → "MgSO4"}}, ControlType → SetterBar}},
Row[Control[{{mmass, 58.5, "Or input average molar mass"}}, Control[{{vh, 2, "Van't Hoff index"}]}],
Delimiter,
{{Nstages, 2, "Total number of stages"}, 1, 3, 1, ControlType → SetterBar},
{{Nelements, 1, "Number of element discretization"}, 1, 4, 1, ControlType → SetterBar},
Delimiter,
{{wsysfeed, 100., "system feed flow rate [m³/h]"}, 1, 200, Appearance → "Labeled"},
{{csysfeed, 35., "system feed TDS [g/L]"}, 1, 50, Appearance → "Labeled"},
{{ΔP, 50., "system feed ΔP [barg]"}, 1, 100, Appearance → "Labeled"},
Delimiter,
Row[Control[{{pumpeff, 0.84, "Pump Efficiency"}}, Control[{{motoreff, 0.95, "Motor Efficiency"}},
Control[{{ERDeff, 1.00, "ERD Efficiency"}]}],
Delimiter,
Grid[{{Control[{{A1, 0.7, "A1 [LMH/bar]"}}, Control[{{A2, 7., "A2 [LMH/bar]"}},
Control[{{A3, 60., "A3 [LMH/bar]"}]}],
{Control[{{B1, 0.2, "B1 [LMH]"}}, Control[{{B2, 4.5, "B2 [LMH]"}}, Control[{{B3, 25., "B3 [LMH]"}]}],
{Control[{{S1, 2230., "S1 [m²]"}}, Control[{{S2, 350., "S2 [m²]"}}, Control[{{S3, 100., "S3 [m²]"}]}],
Dividers → {All, False}, Alignment → Left},
{acc, 32, "numerical accuracy (log scale)"}, 8, 128, 8, Appearance → "Labeled"},

TrackedSymbols → {Nelements, Nstages, wsysfeed, csysfeed, ΔP, A1, A2, A3, B1, B2, B3, S1, S2, S3,
mmass, vh, CPFts, T, acc, pumpeff, motoreff, ERDeff},
ContinuousAction → False,
ContentSize → Full,
FrameLabel → "Made by Andreu Navarro, ChemE bachelor thesis - Universitat de Barcelona",
Alignment → Center,

Initialization → {
  R = 0.0831446261815324; T0 = 273.15; kf = 5 × 10-5 × 1000 × 3600;
  Table[CPF[i, j] = 1.05, {i, 1, 3}, {j, 1, 4}]; CPFts = 1.05;

  bdiag[1] = ; bdiag[2] = ; bdiag[3] = ;
}
]

```

The control options are set as per section 4.5. A variety of Manipulate options are also given to improve the user experience, such as Continuous Action False, which makes it so that the simulation doesn't constantly update when the user is moving a slider, and only does so when the slider is finally let go.

