



**American University of Beirut**

**Faculty of Engineering and Architecture**

**MECH 499: Undergraduate Research**

**Final Report**

**Spring 2016-2017**

**Project Title: RO wave energy system**

|  |  |  |
| --- | --- | --- |
| *Daniel Farhat:* | *201400923* | *(ME)* |

**Advisor:** *Dr. Matthias Liermann, MECH*

Table of Contents

[1. Spiral Membrane Configuration 3](#_Toc482351701)

[2. Mathematical Model 4](#_Toc482351702)

[2.1. RO Salt Rejection Full model 4](#_Toc482351703)

[2.2. Salt Rejection, Approximate Model 6](#_Toc482351704)

[2.3. Mass balances 6](#_Toc482351705)

[2.4. Solution Method 7](#_Toc482351706)

[**2.4.1.** **MATLAB** 7](#_Toc482351707)

[**2.4.2.** **Open Modelica** 7](#_Toc482351708)

[3. Results and model validation 8](#_Toc482351709)

[4. Appendix 9](#_Toc482351710)

[4.1. RO MATLAB Function 9](#_Toc482351711)

[4.2. Test Code 11](#_Toc482351712)

[Works Cited 12](#_Toc482351713)

**Tables**

Table 1: Comparison between current model and literature results 7

**Figures**

Figure 1: Spiral membrane reverse osmosis unit schematic 3

Figure 2: Schematic of an internal node of the discretized spiral membrane unit 4

Figure 3: last RO module 6

Figure 4: RO unit composed of 5 discretized modules 6

1. **Spiral Membrane Configuration**

There are two types of reverse osmosis (RO) units, the hollow fiber membrane unit and the spiral membrane unit. In this study the spiral membrane unit is modeled and used in experimentation.

The spiral membrane unit consists of one or more sets of alternating layers called leaves. Each leaf consists of a feed channel spacer - RO membrane – permeate collecting membrane - RO membrane. The leaves are pressed onto each other and curled as seen in Figure 1. A solid casing covers the outer layer of the first leaf and serves to seal the unit from any potential leaking outside the system.

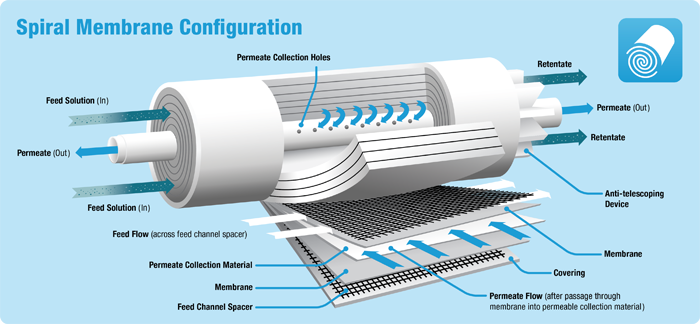


Figure 1: Spiral membrane reverse osmosis unit schematic

The feed is provided across the axial direction and is allowed to pass only into the feed channel spacer. The spacer is used to separate the different membranes and allow for the passage of feed water. As the feed water is pumped into the spacer channel, it is highly pressurized and thus will be filtered across the RO membrane surface whereby the permeate (filtered water) reaches the permeate collection material. The permeate is then forced in the direction perpendicular to the feed flow as shown in Figure 1 and is collected in the central permeate tube through the permeate collection holes.

In the current model the following assumptions are made:

* Solution diffusion model is used to model the salt rejection in the RO membrane.
* Pressure in the permeate membrane side is constant and equal to atmospheric pressure.
* The desalination process is adiabatic.
* The feed fluid viscosity is assumed to remain constant along the axial direction.
* The RO membrane’s salt and water permeability coefficients remain constant.
* The pressure drop on the feed side is governed by the Poisseuille flow, and is corrected by a coefficient that accounts for the spacer.
* Operation is at steady state; the flow inside the feed channels is fully developed.
* The membrane mass transfer coefficient, *k*, depends on the feed flow conditions.
* Concentration polarization is negligible on the permeate side.

The following assumptions are mostly used in the literature []. Some relax the first assumption and the problem becomes a 2-dimensional iterative solution which increases the level of complexity.

1. **Mathematical Model**
   1. **RO Salt Rejection Full model**

In order to model the process of desalination across the RO membrane, it is required to discretize the RO spiral membrane unit along its axial direction. The salt rejection is then calculated at each discretized unit where the desired output variables are assumed constant.

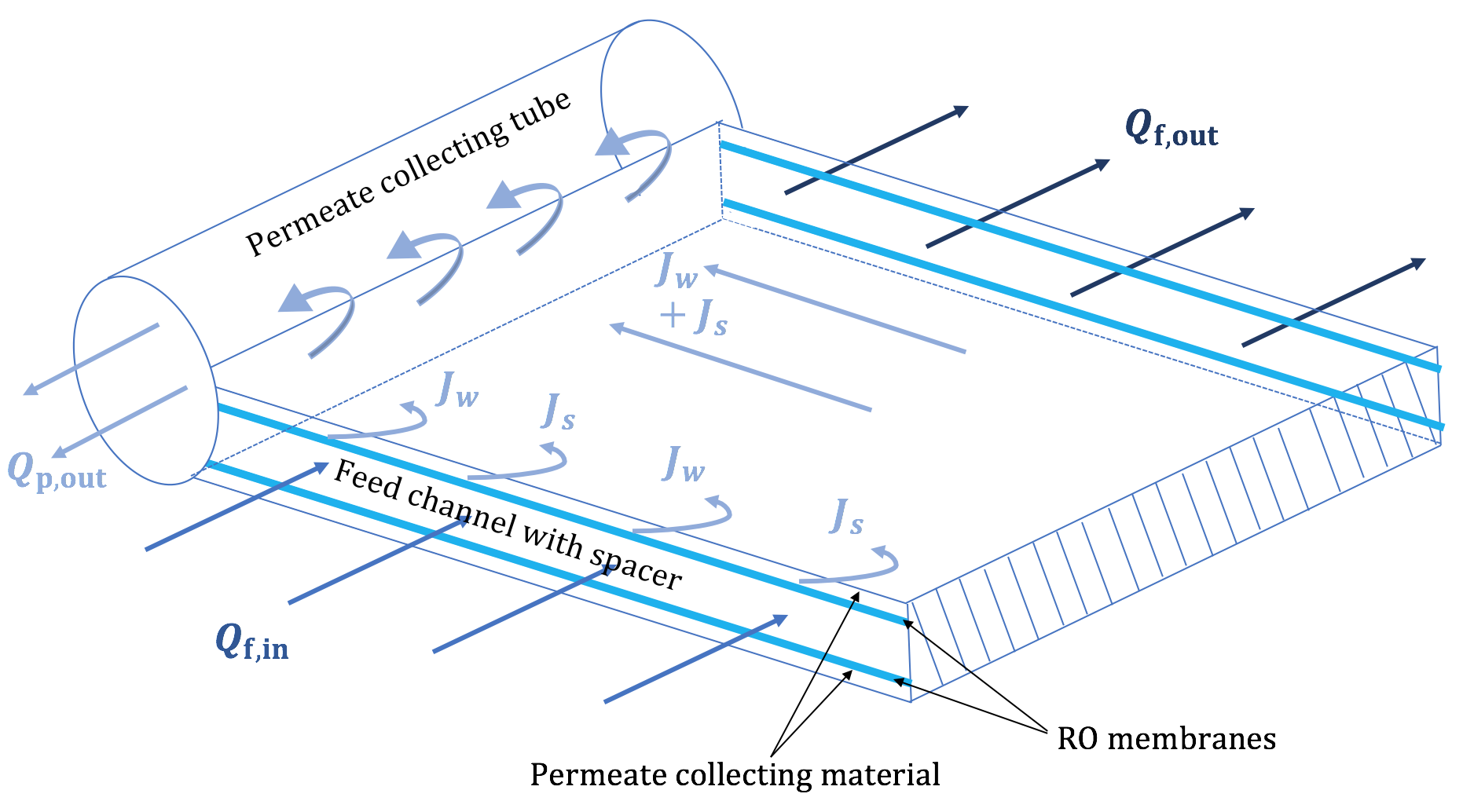


Figure 2: schematic of a discrete unit of the total RO length (one leaf)

The schematic can be reduced to a symbolic diagram that is useful for mathematical modeling.

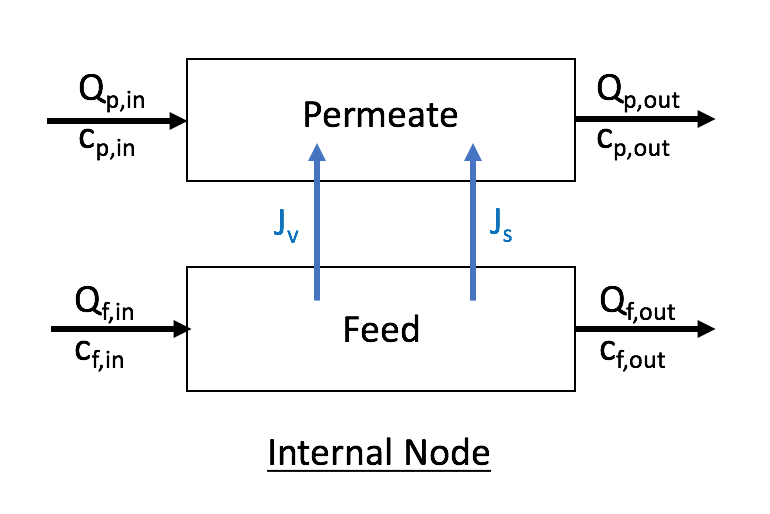


Figure 3: Schematic of an internal node of the discretized spiral membrane unit

Figure 2 illustrates the process of desalination that occurs across one of the internal nodes of the RO unit. The only other type of nodes is the *first node* where the permeate inlet flow is zero. The salt flux, *J­s* is is very small compared to the solvent (water) flux, *J­w*, hence the resulting permeate solution has a low salt concentration.

The fluxes are related to the other variable per the solution diffusion model:

Equation 1

Where, is the difference in pressure between the feed and the permeate side. is the RO membrane solvent permeability coefficient.

is the difference in the osmotic pressure and is given by:

Equation 2

Where, is the concentration at the wall of the RO membrane. This concentration is higher than the feed concentration due to localized salt rejection that accumulates and is then convected with the feed flow. By equating the convective flux to the diffusive flux, one can show the following relationship:

Equation 3

where, *k* is the mass transfer coefficient. This coefficient is dependent on the membrane properties and is also a function of the flow conditions on the feed side. The relationship takes the form [ref]:

Equation 4

where the coefficients *a* and *b* are determined experimentally and are depending on the RO membrane.

Finally, the salt flux across the membrane is expressed as:

Equation 5

* 1. **Salt Rejection, Approximate Model**

The full equations are highly non-linear and require special care in choosing the solution method. However, one assumption can be made which can reduce the level of non-linear coupling, as proposed by Sundaramoorthy et al. [1]

Equation 6

This assumption states that the mass flux across the RO membrane is due to the solvent flux only. In other words, the salt mass flux is negligible compared to the water mass flux across the RO membrane, which is a good assumption given the high rejection ratios of common RO membranes.

Combining equations 1, 2 and 5 the solvent flux becomes:

Equation 7

Substituting from equation 6:

Equation 8

Finally, using equations 1, 2 and 3, the following relation can be derived:

Equation 9

Hence, equations 8 and 9 are 2 equations with two unknowns () that can be solved numerically. Once are found, can be found from equation 6.

* 1. **Mass balances**

In order to find *cap*, the species mass balance equations are used. There will be 2 mass balance equations, one for the salt and another for the solution. The 2 mass balances will be applied on the feed and the permeate sides. Note that in the following, *Jv* is the solvent flux in mol/(m2-s) whereas *Jw* is the solvent flux in m3/ (m2-s).

On the permeate side (for internal node):

Equation 10

Equation 11

where *i* resembles the node position along the axial direction of the RO unit. The factor of 2 accounts for the fact that we have 2 RO membranes squeezing the permeate collecting material per leaf.

The mass balance equations for the first node on the permeate side are as follows:

Equation 12

Equation 13

On the feed side:

Equation 14

Equation 15

* 1. **Solution Method**

The model was implemented on both MATLAB and open Modellica. In this section, I will present the details of the implementation of each model

* + 1. **MATLAB**

On MATLAB, the full model is solved iteratively at every node of the discretized RO unit. The following iterative scheme was applied:

1. Start with an initial guess for ().
2. Dividing equations 6 and 7, *Qp,i+1* is eliminated. Then substituting for *Js* from Equation 5, the following relation is formed:

Equation 16

This equation relates to and can be solved iteratively on MATLAB with an initial guess for

1. Using Equation 3, modify the value of , and repeat unit convergence. The stopping criteria is the error on Equation 1.
2. Find the flow rates from the mass balances.

An alternative scheme is used with the simplifying assumption in equation 6. Hence, equations 8 and 9 are solved simultaneously to get . Then, can be found from equation 6. Those variables are then used in the mass balance equations to find all other variables of the model.

* + 1. **Open Modelica**

On Open Modelica, the discretized modules are each implemented in a modelica model and then are interfaced with appropriate fluid ports that have salinity as a stream variable. The first and last RO discrete modules are interfaced with fluids ports from the Open Hydraulics library. A salinity port is done at the first node to be able to input the sea water salinity.

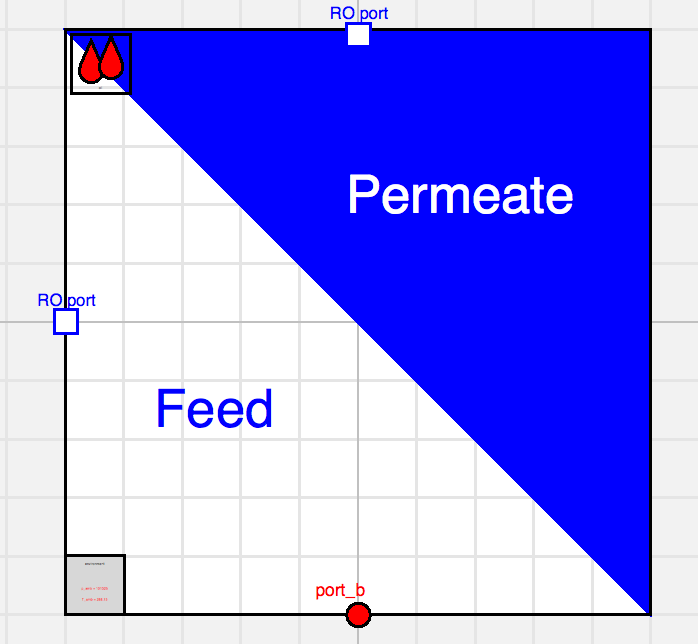


Figure 4: last RO module

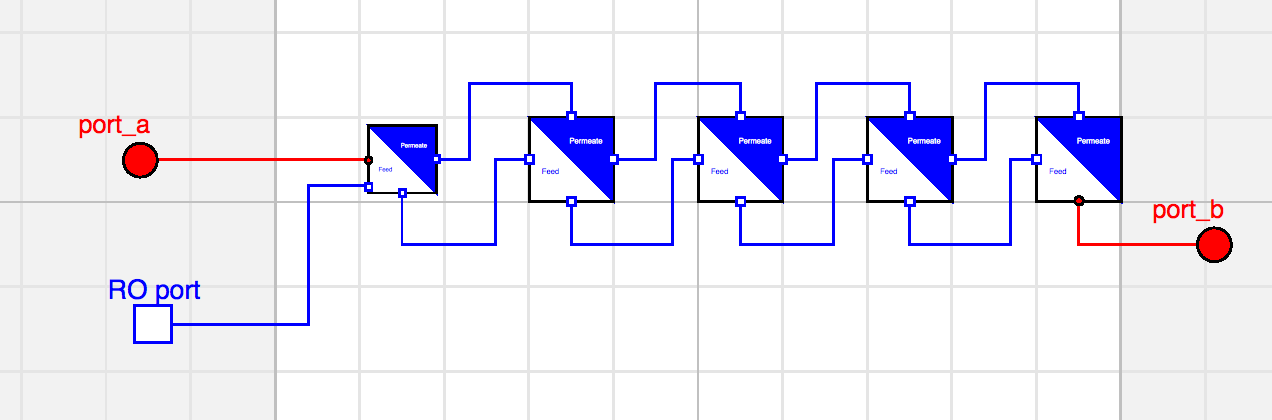


Figure 5: RO unit composed of 5 discretized modules

1. **Results and model validation**

The model is tested on different operating conditions and compared with a model developed in the literature by S. Senthilmurugan et. al [2]. The coefficients of the membrane are assumed the same as the one in the literature just to do the comparison.

The following table illustrates the results and the percent error:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Output**  **Input** |  | | |  | | |
| **Variables**  **()** | Modeled | Literature | % error | Modeled | Literature | % error |
| 25, 1.5, 15 | 18.3 | 18.9 | -3.3 | 4.76 | 4.74 | 0.8 |
| 25, 1.5, 25 | 31.1 | 33.6 | -8.0 | 3.03 | 2.98 | 1.65 |
| 25, 1.5, 35 | 43.9 | 48.2 | -9.8 | 2.30 | 2.30 | 0 |
| 25, 3, 15 | 17.5 | 16.9 | 3.4 | 4.96 | 5.08 | -2.42 |
| 25, 3, 25 | 30.23 | 31.4 | -3.9 | 3.10 | 3.13 | -0.97 |
| 25, 3, 35 | 43.0 | 45.8 | -6.5 | 2.34 | 2.38 | -1.71 |
| 15, 1.5, 15 | 18.4 | 18.8 | -2.2 | 4.90 | 4.98 | -1.63 |
| 15, 1.5, 25 | 31.1 | 33.3 | -7.1 | 3.21 | 3.28 | -2.18 |
| 15, 1.5, 35 | 43.9 | 47.8 | -8.9 | 2.50 | 2.67 | -6.8 |

Table 1: Comparison between current model and literature result

1. **Appendix**
   1. **RO MATLAB Function**

The MATLAB model was implemented through a function that calculates the output flow rates, salinities, and pressures of the permeate and feed, given the inlet flow rates, salinity and pressure of feed. The membrane properties are inputted to the function too.

                %% Spiral RO model: Daniel Farhat March4,2017 %%

function [c\_f,Q\_f,p\_f,c\_p,Q\_p,Power]=RO\_pCQ(h\_b,L,N\_x,W,n\_leaves,L\_mix,K,A\_w,B\_s,k\_f,p\_fin,c\_fin,Q\_fin)

%% constants

M\_Na = 22.9898e-3; % Molar mass of sodium [kg/mol]

M\_Cl = 35.4527e-3; % Molar mass of chlorine [kg/mol]

M\_s = M\_Na + M\_Cl;

M\_w = 18.0153e-3;  % Molar mass of water [kg/mol]

D = 1.6e-9;%salt diffusion coefficient m/s

T = 273+25; %K

gamma = 8.3086e-5;%gas constant R [bar-m3/K-mol]

%% initialization

dx=L/N\_x;

A\_dx=dx\*W;

x\_L(1)=0;

c\_f(1)=c\_fin;

p\_f(1)=p\_fin;

Q\_f(1)=Q\_fin;

rho(1) = NaClDensityc(c\_f(1));%kg/m3

mu(1) = SW\_Viscosity(T,'K',c2S(c\_f(1)),'ppt');%[kg/m-s]

Sc(1) = mu(1)/(D\*rho(1));%Schimdt dimensionless number

U\_b(1) = Q\_f(1)/(2\*h\_b\*W);%velocity m/s

Pe(1) = 2\*h\_b\*U\_b(1)/D;%Peclet dimensionless number L-U/D

%correlation from Senthilmurugan source [1]

k(1) = 0.753\*((K/(2-K))^(0.5))\*(D/h\_b)\*(Sc(1)^(-1/6))\*(Pe(1)\*h\_b/L\_mix)^(0.5);%m/s

no\_permeate=0;

%% solution

for i=1:N\_x

    x\_L(i+1) = x\_L(i)+dx;

    %%check if inlet pressure>osmotic pressure at c\_fin. otherwise, Q\_p=0

    if (p\_f(1)-gamma\*T\*c\_f(1))<=0

        no\_permeate=1;

        %display('inlet pressure is not high enough')

        %break

    end

    %% finding new c\_p, Q\_p and fluxes

    cp=@(Jv)(c\_f(i)/(1+Jv/(B\_s\*exp(Jv/k(i)))));

    fn=@(Jv)((Jv-A\_w\*p\_f(i)/(1+A\_w\*gamma\*T/B\_s\*cp(Jv)))\*1e20);

    J\_v(i)=bisection(fn,0\*p\_f(i)\*A\_w,2\*p\_f(i)\*A\_w);%m3/m2-s

    %J\_v(i)=fsolve(fn,p\_f(i)\*A\_w,options);%m3/m2-s

    J\_v1(i)=J\_v(i)\*NaClDensityc(0)/M\_w;%mol/m2-s

    c\_p(i)=cp(J\_v(i));%mol/m3

    c\_m(i)=c\_p(i)+(c\_f(i)-c\_p(i))\*exp(J\_v(i)/(k(i)));%mol/m3

    if c\_m(i)>S2c(350)

        display('c\_m over solubility limit. Fouling')

    end

    error\_Jv(i)=(A\_w\*(p\_f(i)-gamma\*T\*c\_m(i))-J\_v(i))/J\_v(i)\*100;

    if abs(error\_Jv(i))>10

        display(['error is Jv, e=' num2str(error\_Jv(i))])

    end

    %%check if pressure>osmotic pressure after CP, model FO

    if (p\_f(i)-gamma\*T\*c\_m(i))<=0

        display('inlet pressure is not high enough, CP')

        %break

    end

    J\_s(i)=B\_s\*(c\_m(i)-c\_p(i));%mol/m2-s

    error(i)=J\_s(i)/(J\_v1(i)\*M\_w+M\_s\*J\_s(i))-c\_p(i)/NaClDensityc(c\_p(i))\*100;%on mass balance

    if abs(error(i))>10

        display(['error in mass balance on node' num2str(i) 'e=' num2str(error(i))])

    end

    if i==1

        Q\_p(1)=2\*A\_dx/NaClDensityc(c\_p(1))\*(J\_v1(1)\*M\_w+J\_s(1)\*M\_s);

    elseif i>1

        Q\_p(i)=Q\_p(i-1)+2\*A\_dx/NaClDensityc(c\_p(i))\*(J\_v1(i)\*M\_w+J\_s(i)\*M\_s);

    end

    %% finding new c\_f and Q\_f

    fn2=@(cf)((cf/NaClDensityc(cf)-(Q\_f(i)\*c\_f(i)-2\*J\_s(i)\*A\_dx)/...

        (Q\_f(i)\*NaClDensityc(c\_f(i))-2\*A\_dx\*(J\_v1(i)\*M\_w+J\_s(i)\*M\_s)))\*1e20);

    c\_f(i+1)=bisection(fn2,0,2\*c\_f(i));

    %c\_f(i+1)=fsolve(fn2,0\*c\_f(i),options);

    Q\_f(i+1) = (Q\_f(i)\*c\_f(i)-2\*J\_s(i)\*A\_dx)/c\_f(i+1);

    %% finding new p\_f

    D\_h=2\*h\_b;%hydrulic diameter

    Re(i)=rho(i)\*U\_b(i)\*D\_h/mu(i);

    f(i)=k\_f\*(Re(i))^(-0.5);

    if Re(i)<=100

        k\_e = 7.208\*(Re(i))^(-1) + 4.729\*1e-4\*Re(i) + 0.168;

    elseif Re(i)<=1000 && Re(i)>100

        k\_e = 10.604\*(Re(i))^(-1) + 2.592\*1e-5\*Re(i) + 0.16;

    else

        k\_e = 4.183\*1e-5\*Re(i)+0.152;

    end

    delta\_p(i)=(2\*f(i)\*dx\*(U\_b(i))^2\*rho(i)/D\_h + rho(i)/2\*(U\_b(i))^2\*k\_e)/1e5;%bar

    p\_f(i+1)=p\_f(i)-delta\_p(i);

    %% finding new k

    rho(i+1) = NaClDensityc(c\_f(i+1));

    mu(i+1) = SW\_Viscosity(T,'K',c2S(c\_f(i+1)),'ppt');

    Sc(i+1) = mu(i+1)/(D\*rho(i+1));%Schimdt dimensionless number

    U\_b(i+1) = Q\_f(i+1)/(2\*h\_b\*W);%velocity m/s

    Pe(i+1) = 2\*h\_b\*U\_b(i+1)/D;%Peclet dimensionless number L-U/D

    k(i+1) = 0.753\*((K/(2-K))^(0.5))\*(D/h\_b)\*(Sc(i+1)^(-1/6))\*(Pe(i+1)\*h\_b/L\_mix)^(0.5);

end

Power=Q\_fin\*p\_fin/Q\_p(end) /(36); %kWh/m3 of desalted water

Q\_p=n\_leaves\*Q\_p;

end

* 1. **Test Code**

The function is tested using the following test code that contains the paramters used in the literature.

%%test code for RO model

k\_f=9.6;%friction factor find experimentally

%% from Senthilmurugan [1]

h\_b=0.7e-3;%m

L=0.88;%m

W=1.43;%m

n\_leaves=3;

%mass transfer coefficient k

L\_mix=0.006;%m %characteristic length of mixing net (spacer)

K=0.5; %mixing efficiency of net

A\_w=1.7e-7;%2.08e-7;%m3/m2-s-bar

B\_s=1.11e-7;%m/s

N\_x=10;%number of nodes along RO length

p\_fin=[15 25 35];%bar

c\_fin=[S2c(1.5) S2c(3)];%mol/m3

Q\_fin=1e-5\*[25 15];%m3/s

for k=1:length(Q\_fin)

for j=1:length(c\_fin)

for i=1:length(p\_fin)

[c\_f,Q\_f,p\_f,c\_p,Q\_p,Power(i,j,k)]=RO\_pCQ(h\_b,L,N\_x,W,n\_leaves,...

     L\_mix,K,A\_w,B\_s,k\_f,p\_fin(i),c\_fin(j),Q\_fin(k));

Q\_pout(i,j,k)=Q\_p(end);

c\_pout(i,j,k)=c\_p(end);

r(i,j,k)=100\*c\_pout(i,j,k)/c\_fin(j);

end

end

end

# Works Cited

|  |  |
| --- | --- |
| [1] | G. S. a. D. M. S. Sundaramoorth, "An analytical model for spiral wound reverse osmosis membrane modules: Part I — Model development and parameter estimation," *Desalination,* vol. 280, pp. 403-411, 2011. |
| [2] | A. A. S. K. G. S. Senthilmurugan, "Modeling of a spiral-wound module and estimation of model parameters using numerical techniques," *Desalination,* vol. 173, pp. 269-286, 2005. |