Le traitement de l'eau de mer : une solution pour des villes côtières

Thème : La ville

MOLOHIO Alexandre
Numéro d'inscription: 31372

Projection du rapport entre les prélèvements d'eau douce et les ressources en eau disponibles en 2040

https://www.lindependant.fr/2023/05/04/secheresse-des-elus-des-pyrenees-orientales-a-barcelone-pour-etudier-le-dessalement-deau-de-mer-11174720.php

Usine d'osmose inverse d'El Prat à Barcelone

https://www.casamerchants.com/nosproduits/product/cid-163.html

Tubes d'osmose inverse

Objectifs :

- Réaliser un osmoseur inverse par nos propres moyens à l'aide d'une membrane brackish water reliée à un tube en PVC par une jonction étanche.
- Mettre en évidence l’influence de paramètres comme la concentration initiale en sel lors de l’utilisation de notre osmoseur
- Comparer la méthode de congélation et d’osmose inverse

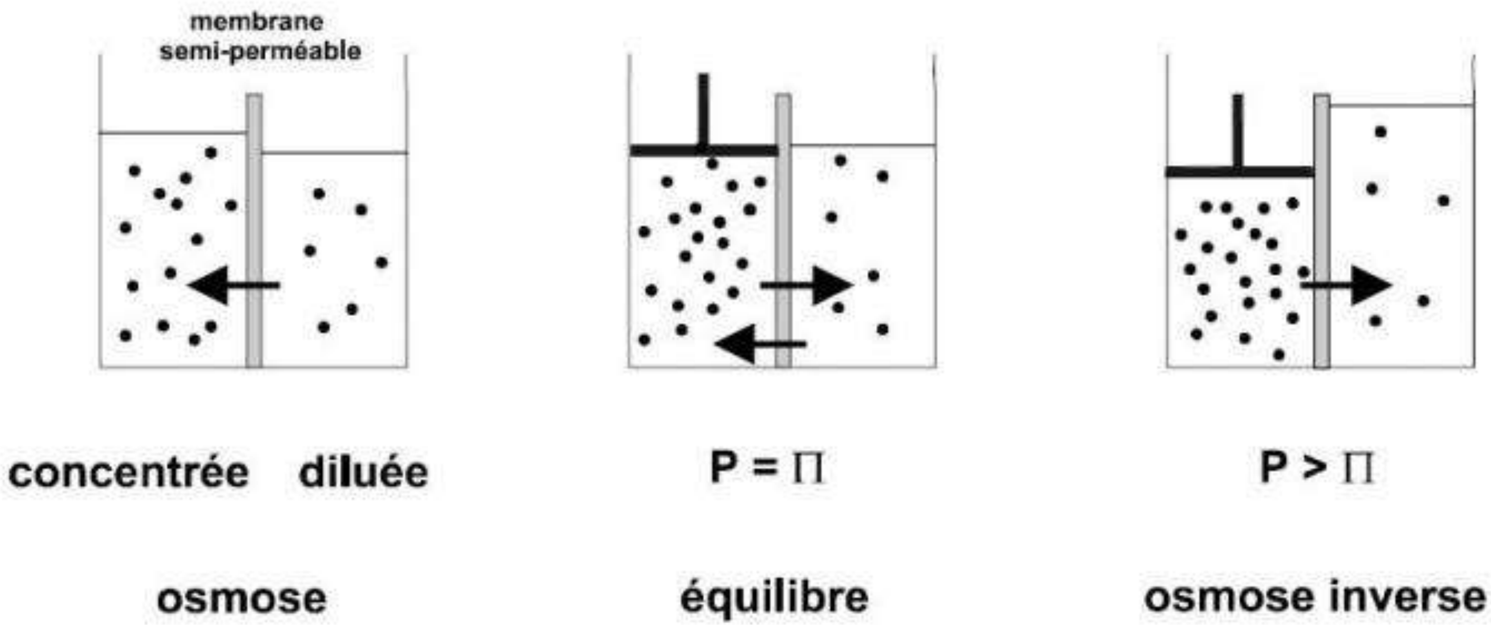
Le dessalement de l'eau de mer par osmose inverse est-il un procédé efficace en comparaison à la congélation ?

1) L'osmose inverse

2) La congélation

3) Comparaison des systèmes

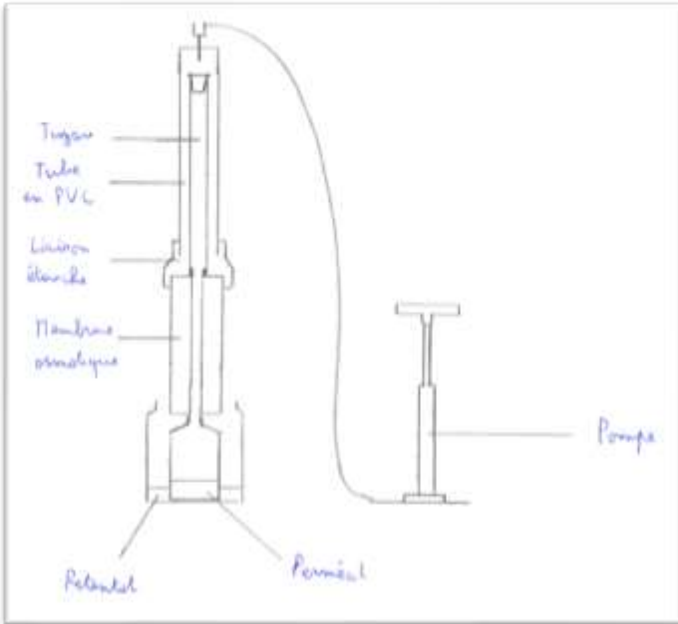
L'osmose inverse : théorie

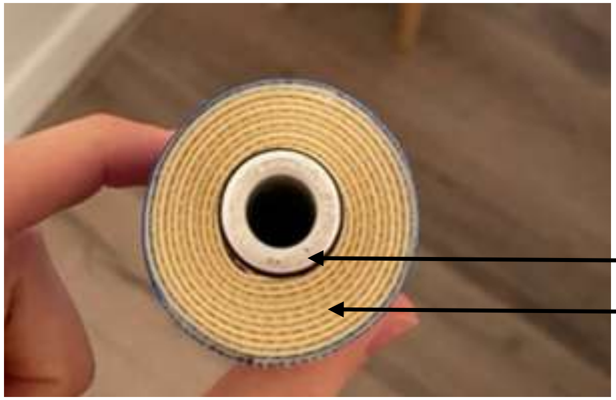


$$\textit{Loi de Van't Hoff} : \boxed{\Delta\Pi = C \times i \times R \times T}$$

Eau de mer : C=35 g/l $\Delta\Pi = 55.2\ bar$

Notre osmoseur inverse





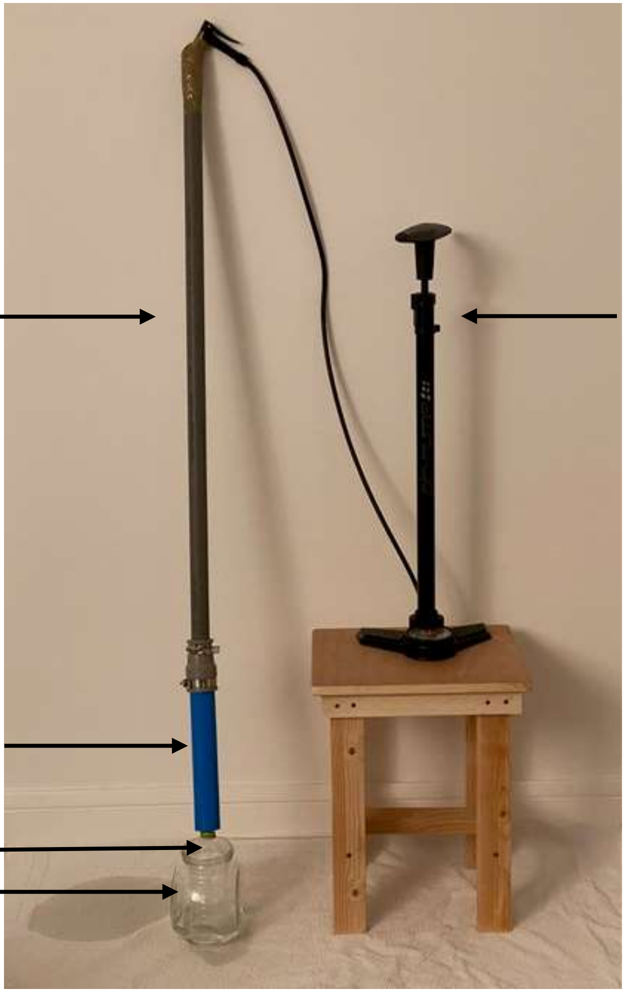
Tube en PVC

Pompe à vélo équipée d'un baromètre

Membrane d'osmose inverse brackish water

Perméat

Rétentat

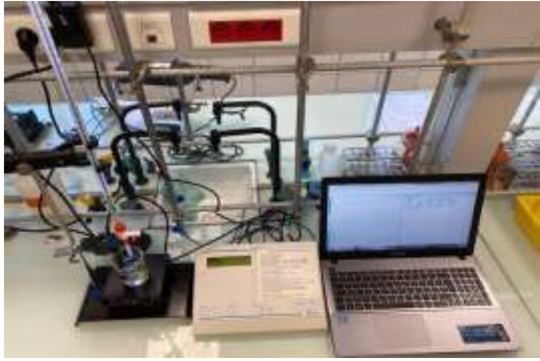


Les problèmes d'étanchéité

Une première expérience

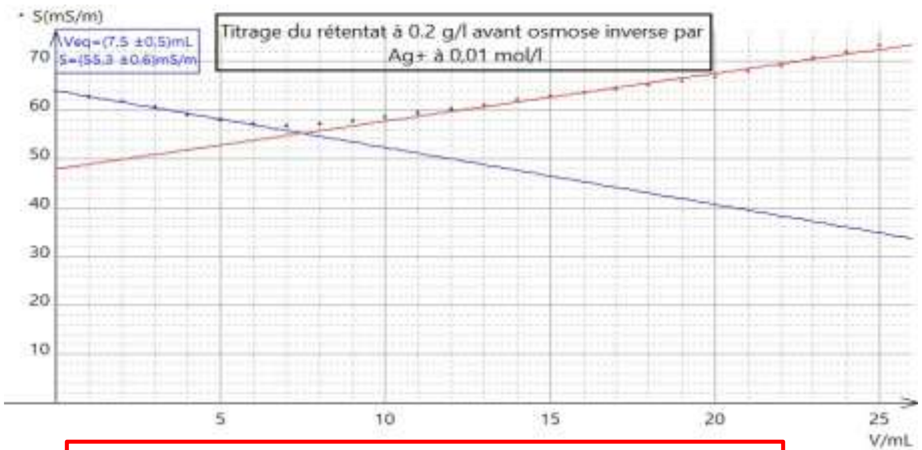
- 4.4 bars
- Ions NaCl à 0.2 g/l dans 1L d'eau
- Pompage pendant environ 5 min

Titrages par une solution de nitrate d'argent AgNO3



- Suivi par conductimétrie

$$\boxed{Ag^{+}_{(aq)} + Cl^{-}_{(aq)} = AgCl_{(s)}}$$



Perméat : Cp= (0.041 ± 0.004) g/l

Rétentat : Cr= (0.265 ± 0.003)g/l

$$Taux\ de\ rétention : R = (1 - \frac{C_p}{C_a}) \times 100$$

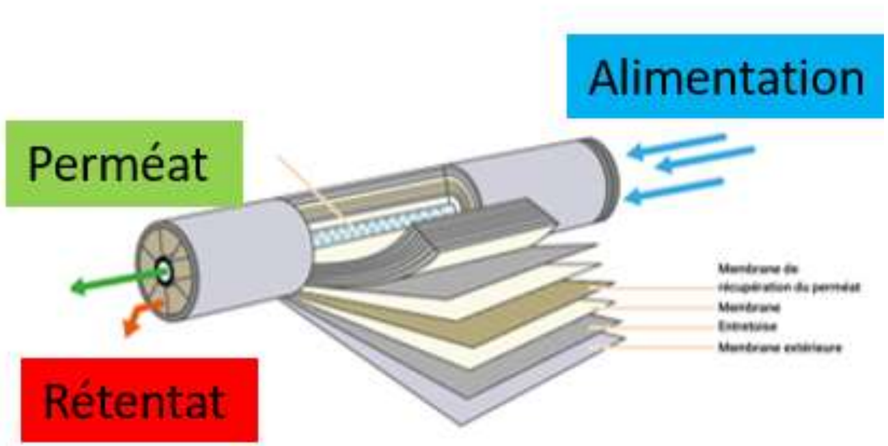
$$\boxed{R = 79,5\%}$$

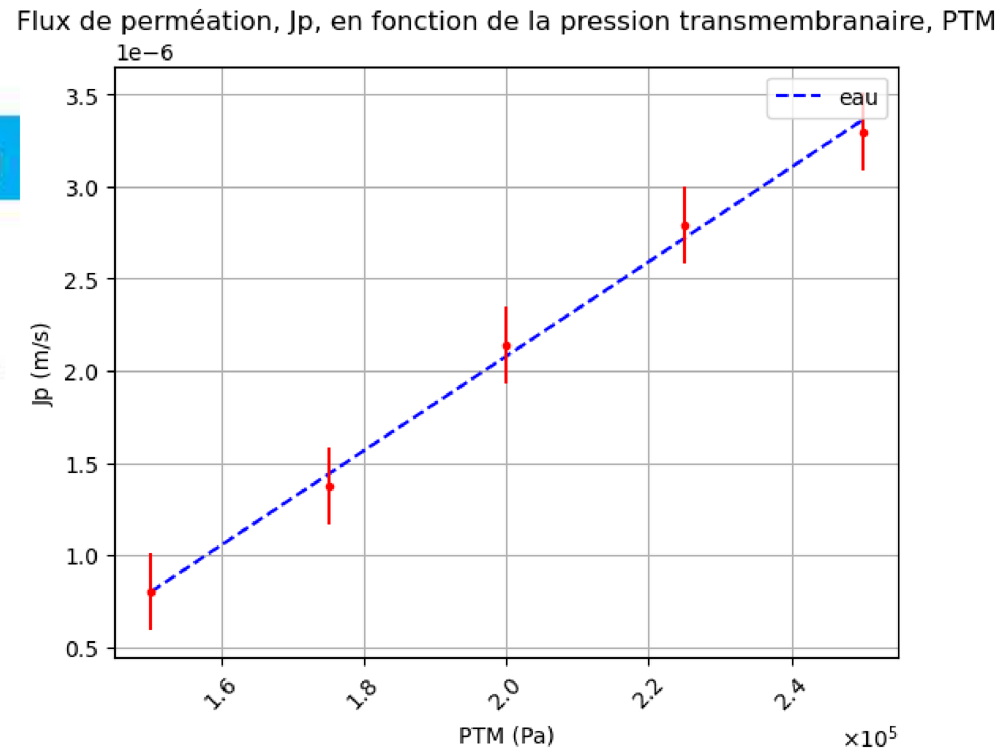
Procédé membranaire

Loi de filtration pour l'eau

$$Flux\ de\ perméation : J_p = \frac{PTM}{\eta R_m}$$

- **Avec de l'eau**



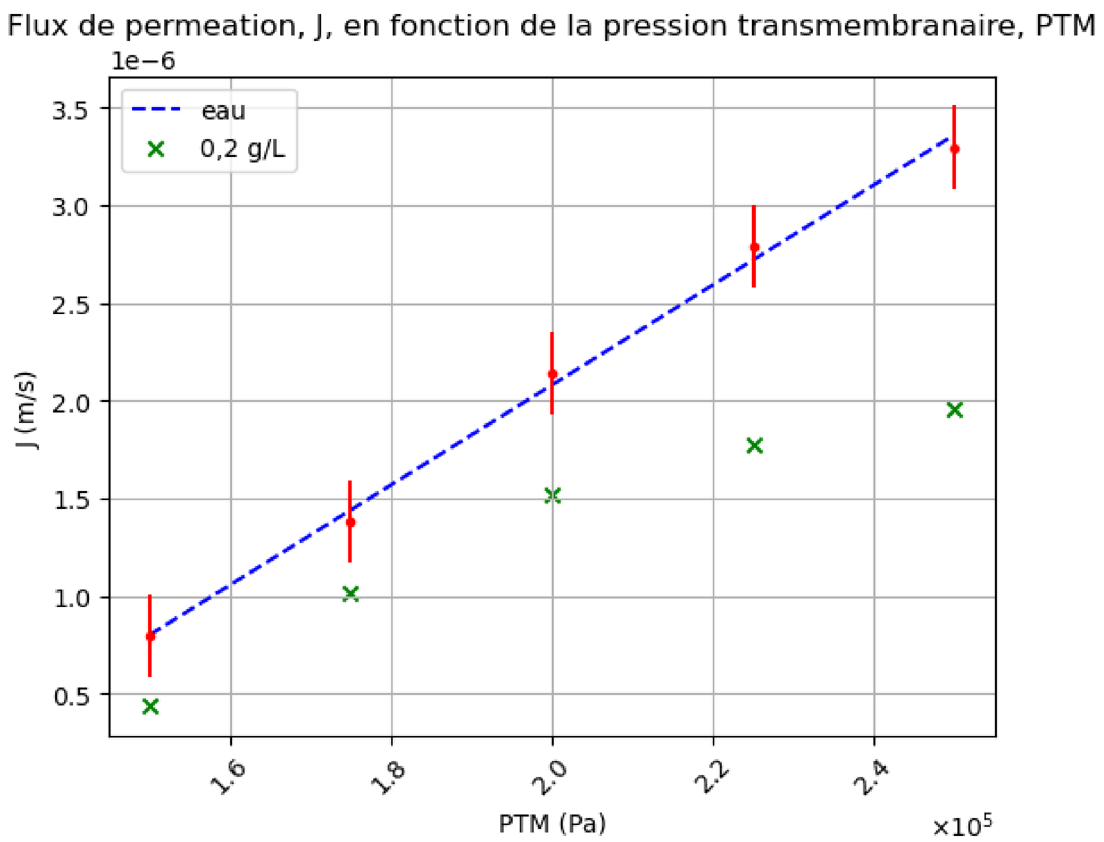


$$J_p = \frac{V}{t \times S} \qquad PTM = \frac{Pe + Ps}{2} - \cancel{P}_p$$

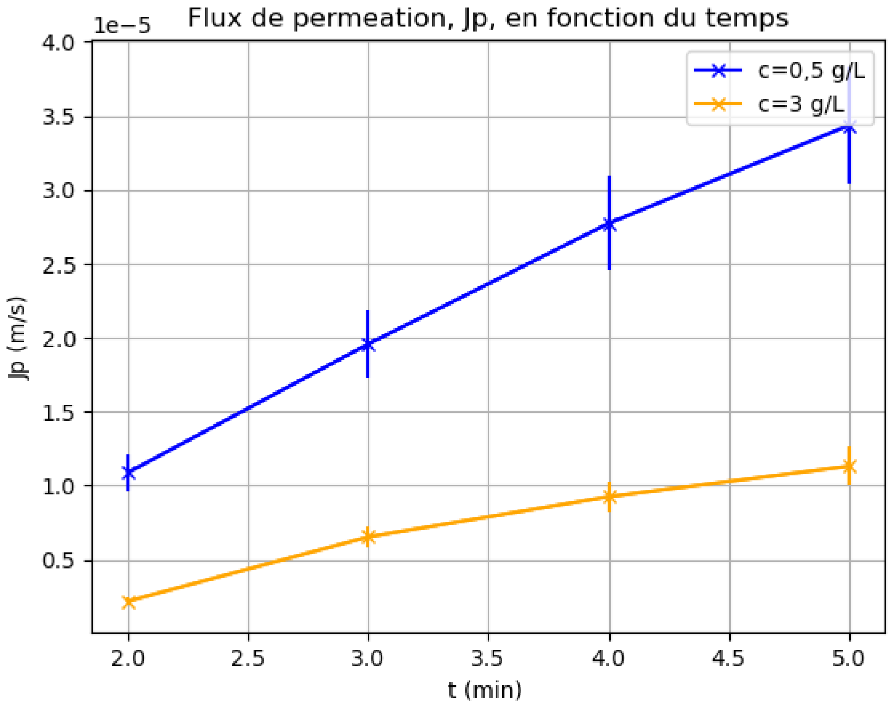
Résistance hydraulique de la membrane, $\boxed{Rm= 3.90e+13\ m-1}$

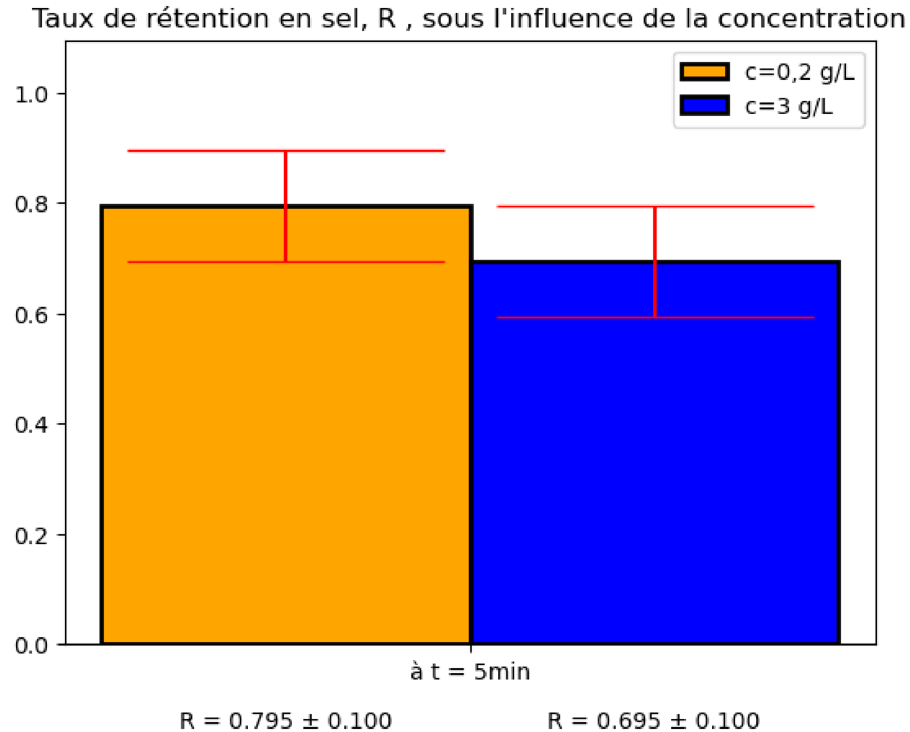
Mise en évidence de l'existence d'une Contre-pression osmotique

Loi de la filtration pour un soluté $J_p = \dfrac{PTM - \Delta\Pi}{\eta R}$ Pression osmotique



Effet de la concentration d'alimentation sur le flux de perméat Jp et sa qualité





$$J_p = \frac{PTM - \Delta\Pi}{\eta R}$$

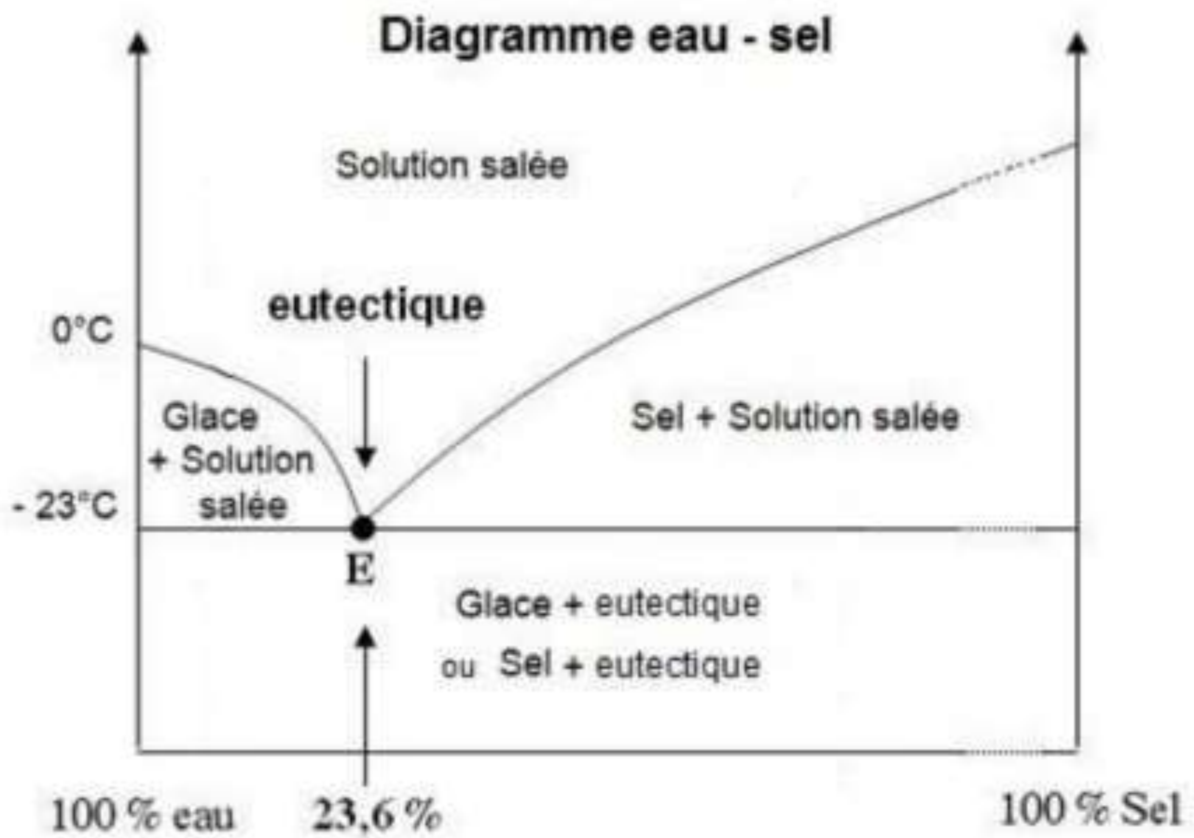
$$Loi\ de\ Van't\ Hoff : \Delta\Pi = C \times i \times R \times T$$

Tableau récapitulatif des solutions traitées sous l’influence de différents paramètres

	Pression	Concentration	Flux d’alimentation	Température	Plusieurs ions
Valeur élevé	+++	– – –	+++	+++	– – –
Valeur faible	– – –	+++	– – – –	– – – –	+++

https://www.memoireonline.com/11/13/7958/Influence-des-parametres-operatoires-sur-les-performances-d-un-systeme-de-dessalement-par-osmose.html

Congélation : théorie

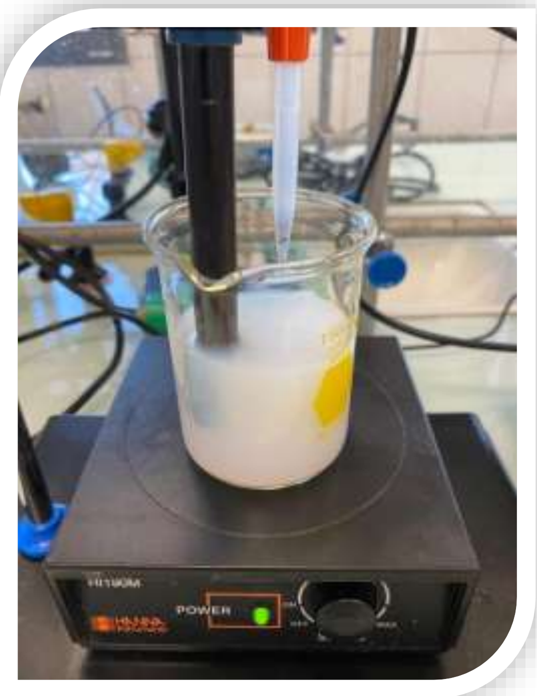


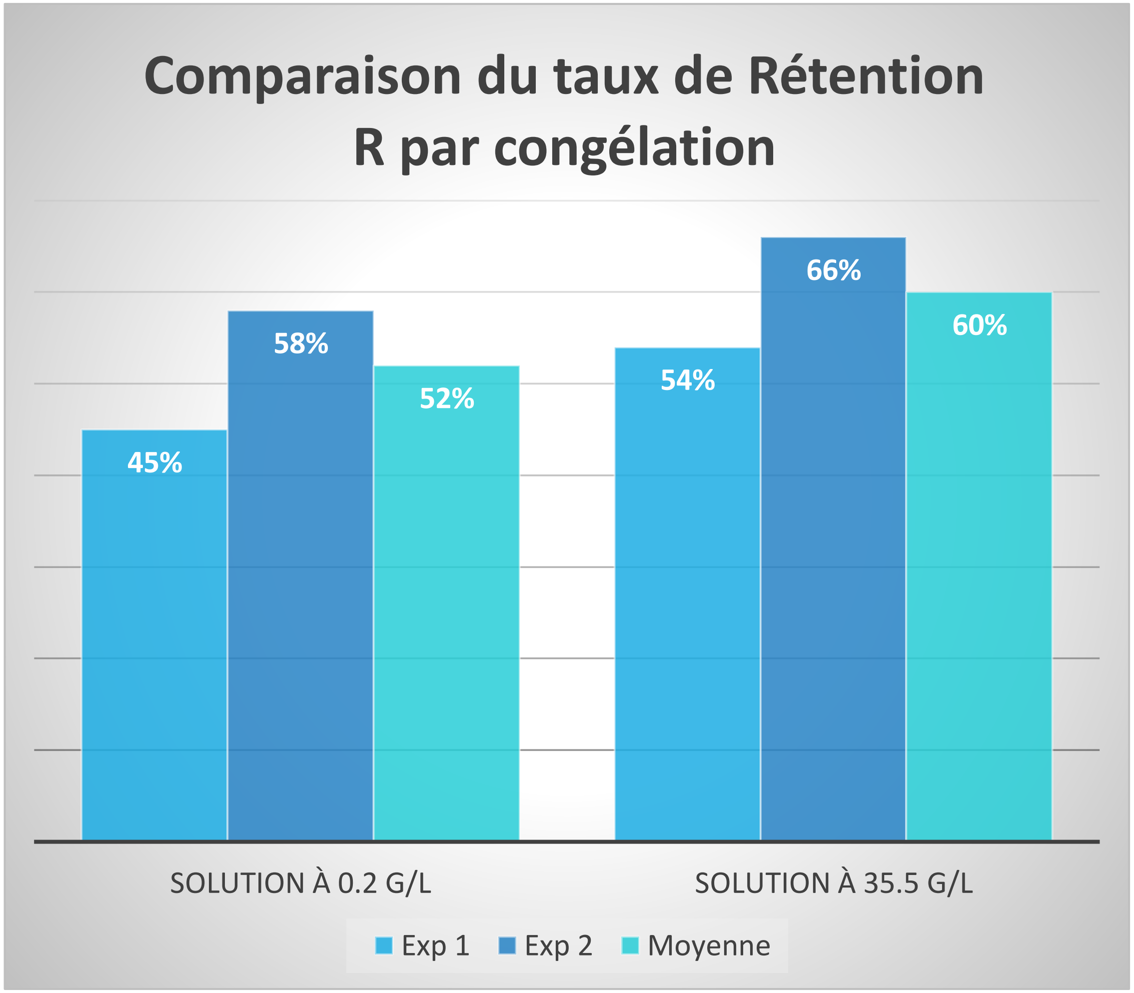
https://public.iutenligne.net/chimie/valls/chimie-11cg1%20-%20copie/liqusol_savoirs_definir5.htm

Deux expériences

Résultats

Titrage par argentimétrie

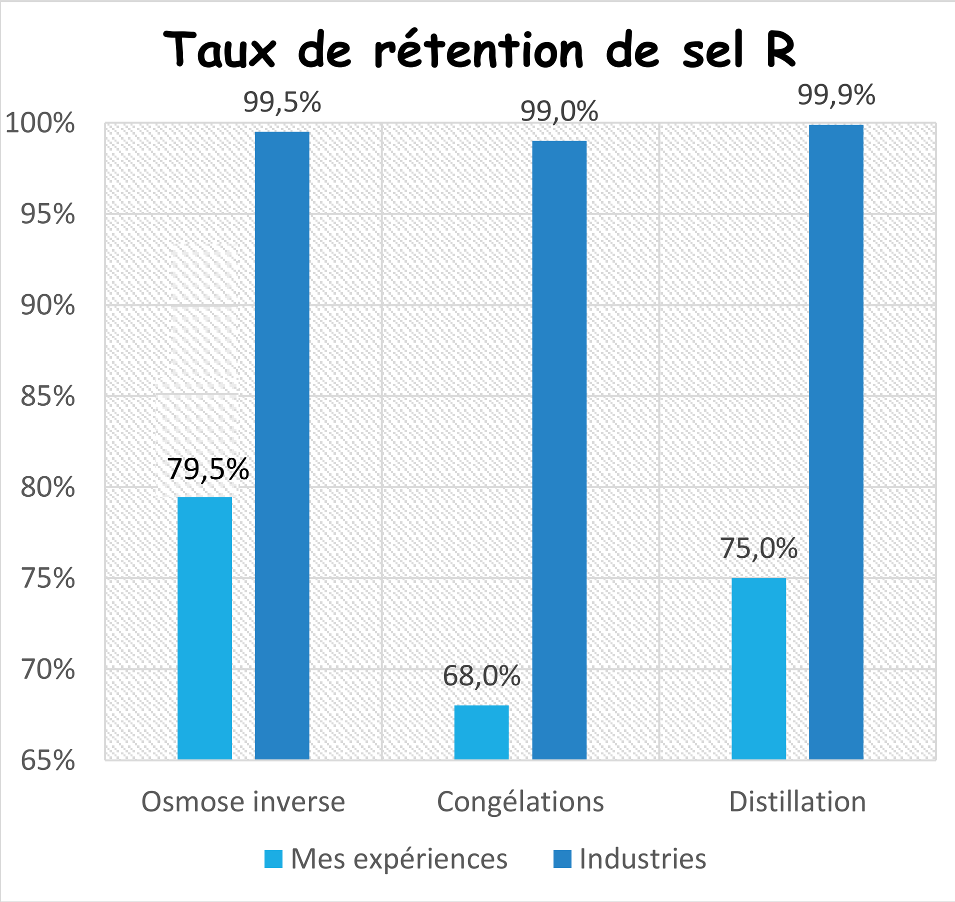


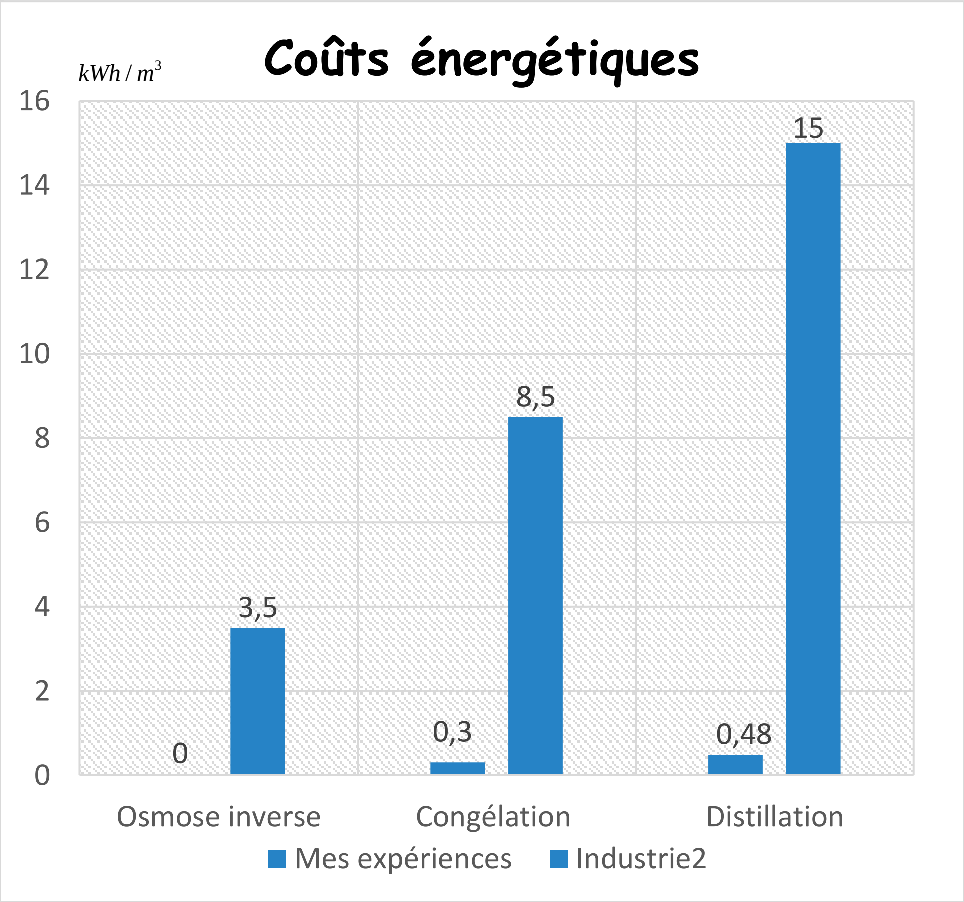


Comparaison des systèmes

Rendement

énergie





https://www.sft.asso.fr/Local/sft/dir/user-3775/documents/actes/congres_2013/articles/6253.pdf

https://www.encyclopedie-energie.org/le-dessalement-deau-de-mer-et-des-eaux-saumatres/#_ftn18

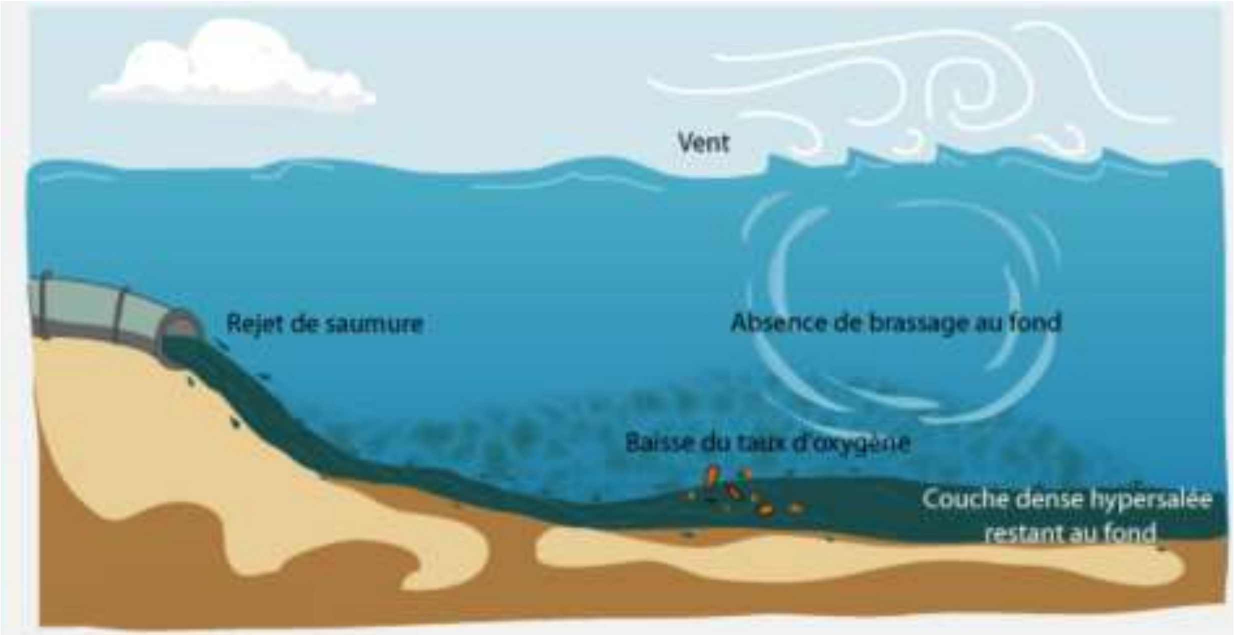
Conclusion :

Osmose inverse >> Congélation

cogénération d'énergie

+ Meilleur rétention de sel
+ Qualité
+ Coût

ATTENTION Environnement

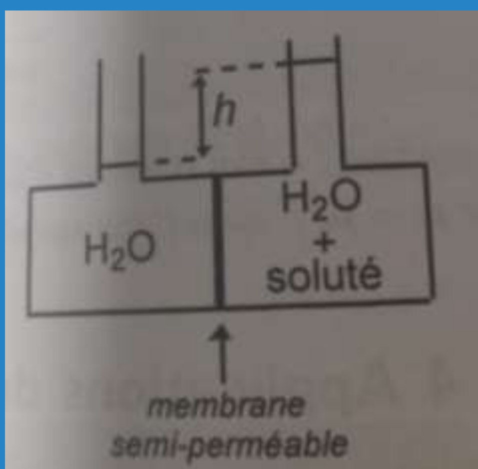


Etat du milieu après implantation de l'usine de dessalement (rejet de saumure)

https://tpestationdedessalement.wordpress.com/2017/02/18/causes-des-impacts-environnementaux%E2%80%AF-et-leur-consequences/

Annexe 1:

Loi de Van't Hoff



On se place dans le cadre des hypothèses suivantes :

- l'effet de la pression sur une phase condensée est prise en compte *via* la considération du volume molaire de l'eau $V^*_{m,H_2O_{(\ell)}}$;
- ce volume molaire dépend peu de la pression : $V^*_{m,H_2O_{(\ell)}}(T,P) \approx V^*_{m,H_2O_{(\ell)}}(T)$;
- l'activité de l'eau est prise égale à sa fraction molaire $x_{H_2O_{(\ell)}}$ (avec $x^g_{H_2O_{(\ell)}} = 1$ et $x^d_{H_2O_{(\ell)}} \to 1$).

On en déduit l'expression du potentiel chimique de l'eau dans chaque compartiment :

Compartiment de gauche :

$$\mu^g_{H_2O_{(\ell)}}(T,P^g) = \mu^\circ_{H_2O_{(\ell)}}(T) + V^*_{m,H_2O_{(\ell)}}(P^g - P^\circ)$$

Compartiment de droite :

$$\mu^d_{H_2O_{(\ell)}}(T,P^d) = \mu^\circ_{H_2O_{(\ell)}}(T) + V^*_{m,H_2O_{(\ell)}}(P^d - P^\circ) + RT\ln\left(x^d_{H_2O_{(\ell)}}\right)$$

L'égalité des potentiels chimiques conduit alors à :

$$-RT\ln\left(x^d_{H_2O_{(\ell)}}\right) = V^*_{m,H_2O_{(\ell)}}(P^d - P^g) = V^*_{m,H_2O_{(\ell)}}\Pi$$

La somme des fractions molaires de l'eau et du soluté vaut 1 dans le compartiment de droite : $x^d_{H_2O_{(\ell)}} + x_s = 1$, on peut ainsi procéder à un développement limité du logarithme compte tenu du fait que $x_s \ll 1$:

$$-RT\ln\left(x^d_{H_2O_{(\ell)}}\right) = -RT\ln(1 - x_s) \approx RTx_s$$

De même, dans l'expression de x_s : $x_s = \dfrac{n_s}{n_s + n_{H_2O_{(\ell)}}} \approx \dfrac{n_s}{n_{H_2O_{(\ell)}}}$

Enfin, puisque $V^*_{m,H_2O_{(\ell)}} = \frac{V}{n_{H_2O_{(\ell)}}}$ avec V le volume du compartiment de droite, on a l'expression finale de la pression osmotique, connue sous le nom de loi de van't Hoff.

Loi de van't Hoff

La **loi de van't Hoff** relative à l'osmose relie la pression osmotique au nombre de moles de soluté du système.

$$\Pi V = n_s RT$$

Chimie Pc/Pc* Ediscience, Dunod p69

Annexe 2:

Estimation de l'incertitude des titrages par la méthode de Monte Carlo

```
#Données du problème
V = 10          #Volume titré en mL
Ct = 1e-2       # Concentration de la solution titrante de soude en mol L^-1
Veq = 12.8      #Volume équivalent en mL

# Données pour la simulation Monte Carlo
a_V = 0.02       #Tolérance de la pipette jaugée de 10 mL en mL
u_Ct = 1e-4      #Incertitude type sur la concentration
a_Veq= 0.05      #demi largeur de la distribution uniforme(ou rectangulaire) sur le volume équivalent

#simulation Monte Carlo
N = 10000                                                  #nombre de simulations souhaité

Ct_sim = Ct + np.random.normal(0, u_Ct, N)                 #utilisation d'une distribution gaussienne(ou normale) car on connaît l'incertitude-type.
Veq_sim = Veq + np.random.uniform(-a_Veq, +a_Veq, N)       #utilisation d'une distribution uniforme pour Veq car on connaît la 1/2 étendue.
V_sim = V + np.random.uniform(-a_V, +a_V, N)               #utilisation d'une distribution uniforme pour V car on connaît la 1/2 étendue.

C=Ct_sim*Veq_sim/V_sim

Cmoy=np.average(C)                        #Renvoie la moyenne des N valeurs  de C obtenues lors de la simulation
u_C=np.std(C,ddof=1)                      #Renvoie l'écart-type expérimental de la série des N simulations : c'est l'incertitude-type

print('Cmoy=', Cmoy, 'mol L^-1')
print('u_C=', u_C*35,5, 'g L^-1')

plt.hist(C*35.5,bins='rice',label='Histogramme des mesures',color='magenta')
plt.title('Histrogramme pour N = 100000 simulations')
plt.xlabel('C (g/L)')
plt.ylabel('Effectifs')
plt.legend()
plt.show()
```

Annexe 3 :

Incertitude sur Jp

```python
#Incertitude sur le flux de perméat

V = np.array([22, 38, 59, 77, 91])
t = np.array([60, 60, 60, 60, 60])

# Equations permettant de calculer le débit de permeation, Q, puis le flux de permeation, J
Q = V * 1e-6 / t  # m3/s
S = 0.46  # m2
J = Q / (S)  # m/s

# Données pour la simulation Monte Carlo
u_V = 10  # incertitude sur le volume du perméat
u_t = 1 # Incertitude type sur le temps

# Simulation Monte Carlo
N = 10000  # nombre de simulations souhaité
u_J = np.zeros(len(V))  # tableau pour stocker les incertitudes de chaque couple de données
for i in range(len(V)):
    J_sim = np.zeros(N)  # tableau pour stocker les valeurs simulées de J
    for j in range(N):
        V_sim = V[i] + rd.uniform(-u_V, u_V)  # variation aléatoire de V[i]
        t_sim = t[i] + rd.uniform(-u_t, u_t)  # variation aléatoire de t[i]
        Q_sim = V_sim * 1e-6 / t_sim
        J_sim[j] = Q_sim / (S)
    u_J[i] = np.std(J_sim, ddof=1)  # Incertitude-type de J pour le couple de données t[i] et V[i]

for i in range(len(V)):
    print(f"Incertitude-type pour le couple de données t[{i}]={t[i]} et V[{i}]={V[i]} : u_J[{i}] = {u_J[i]}")
```

Annexe 4 :

Détermination de la résistance hydraulique

```python
#Valeurs de pression transmembranaire
PTM_bar_alim=np.array([2,2.5,3,3.5,4])
PTM_bar_ret=np.array([1,1,1,1,1])
PTM = 1e5*(PTM_bar_alim +  PTM_bar_ret)/2     #calcul de la PTM en Pa

#Valeurs de volume filtré en mL
V=np.array([22,38,59,77,91])
#Valeurs de temps en s correspondant aux volumes
t=np.array([60,60,60,60,60])

#Equations permettant de calculer le débit de permeation, Q, puis le flux de permeation, J
Q=V*1e-6/t #m3/s
S=0.46 #m²
J=Q/S #m/s

reglin_perm= np.polyfit(PTM,J,1)
a=reglin_perm[0]
b=reglin_perm[1]
plt.plot(PTM, a*PTM+b, 'b--', label='eau')
plt.errorbar(PTM, J, yerr = u_J, fmt='r.')

plt.title('Flux de perméation, Jp, en fonction de la pression transmembranaire, PTM')
plt.xlabel('PTM (Pa)')
plt.ylabel('Jp (m/s)')
plt.legend()

# écriture scientifique en abscisse
ax = plt.gca()
ax.xaxis.set_major_formatter(ScalarFormatter(useMathText=True))
ax.ticklabel_format(axis='x', style='sci', scilimits=(0, 0))
plt.xticks(rotation=45)
plt.grid(True)        # Ajoute un cadrillage

#calcul de la résistance hydraulique
mu=0.001 #Pa.s ou kg m-1 s-1
Rm=1/(mu*a)
###print('Perméabilité de la membrane, Lp=', a*1000*3600*1e5,' L/(h.m2.bar)')
print('Résistance hydraulique de la membrane, Rm=', format(Rm, ".2e"),'m-1')
```

Annexe 4 :

Mise en évidence de la contre-pression osmotique

```
#Mise en évidence de la pression osmotique

#Valeurs de temps en s correspondant aux volumes
t=np.array([60,60,60,60,60])

#Tableaux de volumes en mL
V02=np.array([12,28,42,49,54])

#Calculs des flux de perméation
J_02gl=(V02*1e-6/t)/(S) #m/s

#Tracé des flux de permeation en fonction de la PTM

reglin_perm= np.polyfit(PTM,J,1)
a=reglin_perm[0]
b=reglin_perm[1]
plt.plot(PTM, a*PTM+b, 'b--', label='eau')
plt.errorbar(PTM, J, yerr = u_J, fmt='r.')

plt.scatter(PTM, J_02gl, marker='x', color='g', label='0,2 g/L')

# écriture scientifique en abscisse
ax = plt.gca()
ax.xaxis.set_major_formatter(ScalarFormatter(useMathText=True))
ax.ticklabel_format(axis='x', style='sci', scilimits=(0, 0))
plt.xticks(rotation=45)
plt.grid(True)

plt.legend(loc='best')
plt.title('Flux de permeation, J, en fonction de la pression transmembranaire, PTM')
plt.xlabel('PTM (Pa)')
plt.ylabel('J (m/s)')
plt.grid(True)
plt.show()

#Calcul approximatif de la pression osmotique
A=a
CoPO=(J-J_02gl)*(1/A)
print('La contre pression osmotique est', CoPO*1e-5 , 'bar')
l=list(CoPO)
L=[valeur*1e-5 for valeur in l]
for i in range(len(L)):
    print (format(L[i],".2f") , 'bar')
```

Annexe 5 :

Influence de la concentration

```python
#INFLUENCE DE LA CONCENTRATION

#Concentrations utilisées
'''Jp1 (C= 0.5 g/L ; P=4,4 bars)
Jp2 (C= 3 g/L ; P=4,4 bars)'''

#Tableaux de volumes en mL, on lit les volumes dans le perméat obtenu au temps correspondant
V_05gl=np.array([10,27,51,79])
V_3gl=np.array([2,9,17,26])

#Temps correspondant au prise de volume
t1=np.array([2,3,4,5])

#Calculs des flux de perméation
J_05gl=(V_05gl*1e-6/(t1))/(S) #m/s
J_3gl=(V_3gl*1e-6/(t1))/(S) #m/s

#incertitude sur J
deltaJ_05= 20/100* J_05gl  #choix sur incertitude de J
u_J05=  deltaJ_05 / np.sqrt(3)

deltaJ_3= 20/100 * J_3gl
u_J3=  deltaJ_3 / np.sqrt(3)

#Tracé des flux de permeation
plt.plot(t1, J_05gl, marker='x', color='b', label='c=0,5 g/L')
plt.plot(t1, J_3gl, marker='x', color='orange', label='c=3 g/L')

plt.legend(loc='best')
plt.title('Flux de permeation, Jp, en fonction du temps')
plt.xlabel('t (min) ')
plt.ylabel('Jp (m/s)')
plt.errorbar(t1, J_05gl, yerr = u_J05, fmt='b-', label='c=0,5 g/L')  #incertitude sur le flux de perméation de J_0,5g/L de l'ordre de 2*u_J
plt.errorbar(t1, J_3gl, yerr = u_J3, fmt='orange', label='c=3 g/L')  #incertitude sur le flux de perméation de J_3g/L de l'ordre de 2*u_J
plt.grid(True)
plt.show()
```

Annexe 6 :

Histogramme du taux de rétention

```python
### influence de la concentration sur la qualité du permeat

largeur_barre = 0.3# Largeur de chaque barre

u_R = np.array([0.1, 0.1]) #incertitude sur R

#mesure pour c_initial 0.2 g/L :on a 0.041 g/L
#mesure pour c_initial 2 g/L : on a 0.23 g/L
cp=np.array([0.041,0.61])  # 0.2 g/L ; 2 g/L
ca=np.array([0.2,2])
R=np.array(1-(cp/ca)) #taux de rétention (pourcentage de sel retenu par la membrane)

y1 = [R[0]]
y2 = [R[1]]
x1 = range(len(y1)) # Position de c=0.2g/L
x2 = [i + largeur_barre for i in x1] # Position de c= 2 g/L

#tracé de l'histogramme
b1=plt.bar(x1, y1, width = largeur_barre, color = 'orange', edgecolor = 'black', linewidth = 2,)
b2=plt.bar(x2, y2, width = largeur_barre, color = 'blue', edgecolor = ['black' for i in y1], linewidth = 2)

#barre d'incertitude
plt.errorbar(x1, y1, yerr=u_R[0], fmt='none', ecolor='red', capsize=70)
plt.errorbar(x2, y2, yerr=u_R[1], fmt='none', ecolor='red', capsize=70)

#légende
plt.xticks([r + largeur_barre / 2 for r in range(len(y1))],['à t = 5min'])
plt.title("Taux de rétention en sel, R , sous l'influence de la concentration")
plt.text(x1[0], -0.15, f"R = {R[0]:.3f} ± {u_R[0]:.3f}", ha='center')
plt.text(x2[0], -0.15, f"R = {R[1]:.3f} ± {u_R[1]:.3f}", ha='center')
plt.ylim(0, max(max(y1), max(y2)) + 0.3)
plt.legend([b1, b2], ['c=0,2 g/L', 'c=3 g/L'])
plt.show()
```

Annexe 7 :

Influence de la pression

```
#INFLUENCE DE LA PRESSION
'''Jp1 (C= 0.2 g/L ; P=2 bars)
Jp2 (C= 0.2 g/L ; P=4,4 bars)'''

#Tableaux de volume en mL
V_2bar=np.array([6,19,34,51])
V_44bar=np.array([10,42,85,128])

#temps
t1=np.array([2,3,4,5])

#Calculs des flux de perméation
J_2bar=(V_2bar*1e-6/(t1))/(S) #m/s
J_44bar=(V_44bar*1e-6/(t1))/(S) #m/s

#incertitude sur J
deltaJ_2bar=(20/100) * J_2bar
u_J_2bar=  deltaJ_2bar / np.sqrt(3)

deltaJ_44bar=(20/100) * J_44bar
u_J_44bar=  deltaJ_44bar / np.sqrt(3)

#Tracé des flux de permeation

plt.plot(t1, J_2bar, marker='x', color='r', label='2 bar')
plt.plot(t1, J_44bar, marker='x', color='g', label='4,4 bar')

plt.legend(loc='best')
plt.title('Flux de permeation, J, en fonction du temps')
plt.xlabel('t (min) ')
plt.ylabel('J (m/s)')
plt.errorbar(t1, J_2bar, yerr = 2*u_J_2bar, fmt='r.')
plt.errorbar(t1, J_44bar, yerr = 2*u_J_44bar, fmt='g.')
plt.grid(True)
plt.show()
```

Annexe 8 :

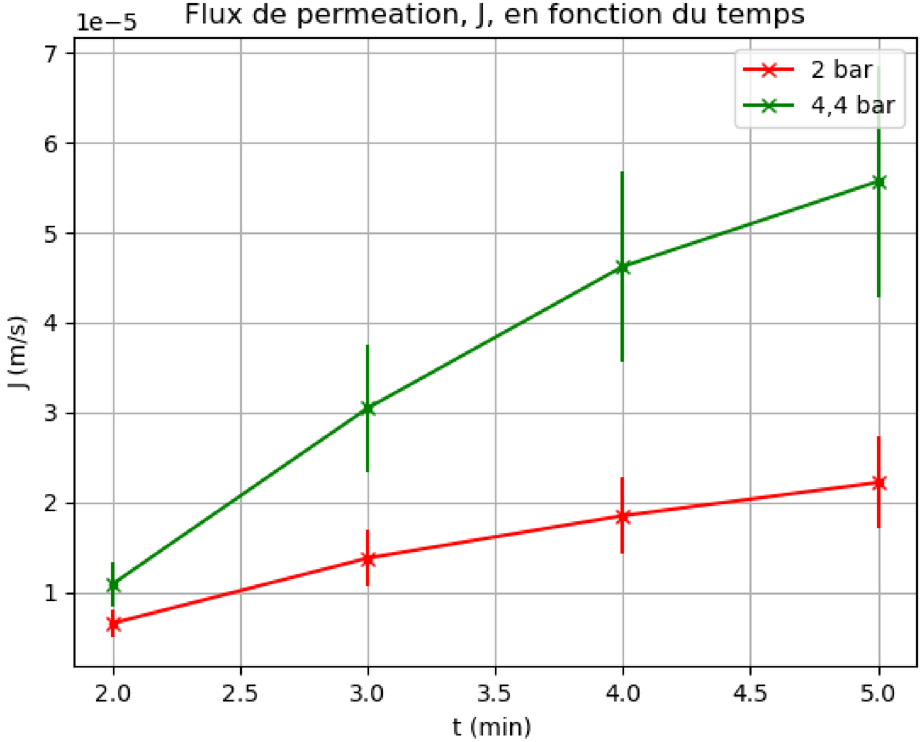
Histogramme du taux de rétention avec comme paramètre la pression

```
#influence de la pression
#mesure à 0.2g/l pression : 2 bars ; 4.4bars

largeur_barre = 0.3# Largeur de chaque barre

u_R = np.array([0.2, 0.2]) #incertitude sur R
cp=np.array([0.021,0.051]) # 2 bars ; 4.4bars
ca=np.array([0.2,0.2])
R=np.array(1-(cp/ca))

#tracé de l'histogramme
y1 = [R[0]]
y2 = [R[1]]
x1 = range(len(y1)) # Position de P=4,4bar
x2 = [i + largeur_barre for i in x1] # Position de P=2bar
b1=plt.bar(x1, y1, width = largeur_barre, color = 'green', edgecolor = 'black', linewidth = 2,)
b2=plt.bar(x2, y2, width = largeur_barre, color = 'red', edgecolor = ['black' for i in y1], linewidth = 2)
plt.errorbar(x1, y1, yerr=u_R[0], fmt='none', ecolor='black', capsize=70)
plt.errorbar(x2, y2, yerr=u_R[1], fmt='none', ecolor='black', capsize=70)

#Légende
plt.xticks([r + largeur_barre / 2 for r in range(len(y1))],['à t = 5min'])
plt.title("Taux de rétention en sel, R , sous l'influence de la pression")
plt.text(x1[0], -0.15, f"R = {R[0]:.3f} ± {u_R[0]:.3f}", ha='center')
plt.text(x2[0], -0.15, f"R = {R[1]:.3f} ± {u_R[1]:.3f}", ha='center')
plt.ylim(0, max(max(y1), max(y2)) + 0.3)
plt.ylabel('R')
plt.legend([b1, b2], ['P=4,4 bars', 'P=2 bars'])
plt.show()
```

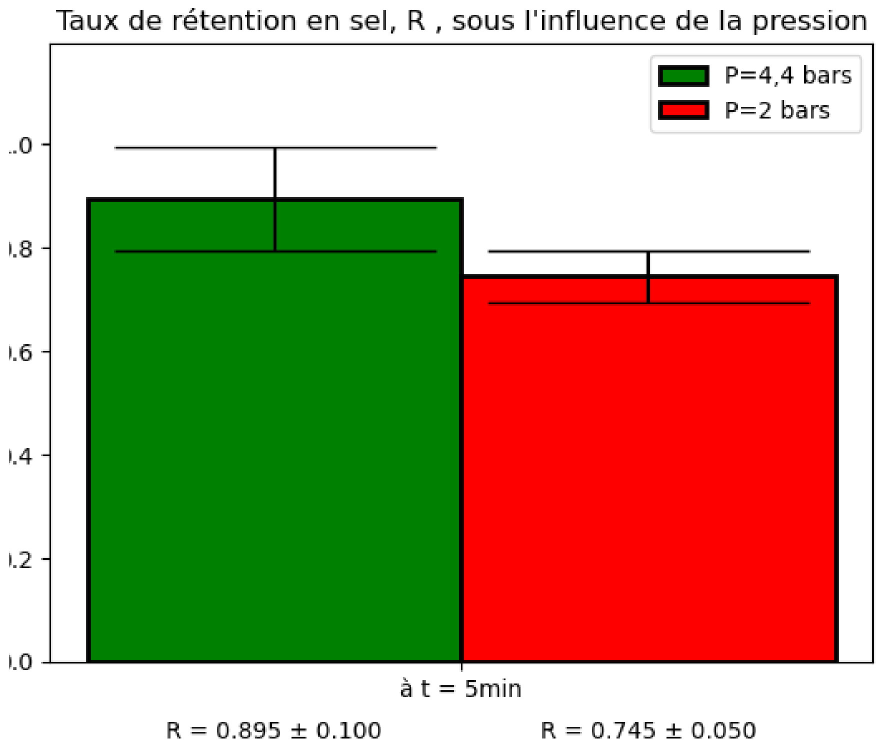
Annexe 9 :

Effet de la pression d’attaque sur le flux de perméat Jp et sa qualité

Concentration en sel : $0.2\ g/l$

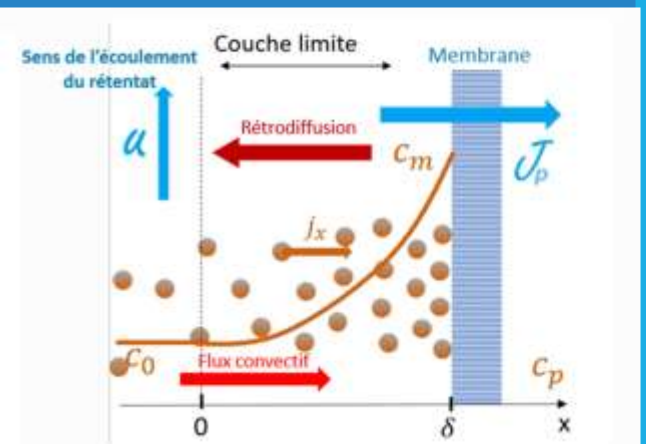


$$J_p = \frac{PTM - \Delta\Pi}{\eta R}$$



Annexe 10 :

Accumulation de la matière à la membrane



Polarisation de concentrations

Équation de conservation de la masse : $\cancel{\frac{\partial c}{\partial t}} = -div(\vec{j}) \cancel{\pm r}$ $\quad \frac{dj_x}{dx} = 0 \quad j_x = cste$

Equa. diff.

Flux de perméation : $j_x = \boxed{Jc - D\frac{dc}{dx} = Jc_p}$

Intégration :

$$\frac{dc}{c - c_p} = \frac{J}{D}dx$$

$$\int \frac{dc}{c - c_p} = \frac{J}{D}\int dx$$

$$\int \frac{d(c - c_p)}{c - c_p} = \frac{J}{D}\int dx$$

$$\boxed{\ln(c - c_p) = \frac{J}{D}x + K}$$

Application des Conditions limites:

$x = 0 \quad c = c_0 \quad \frac{c - c_p}{c_0 - c_p} = e^{\frac{Jx}{D}}$

$x = \delta \quad c = c_m \quad \frac{c_m - c_p}{c_0 - c_p} = e^{\frac{J\delta}{D}}$

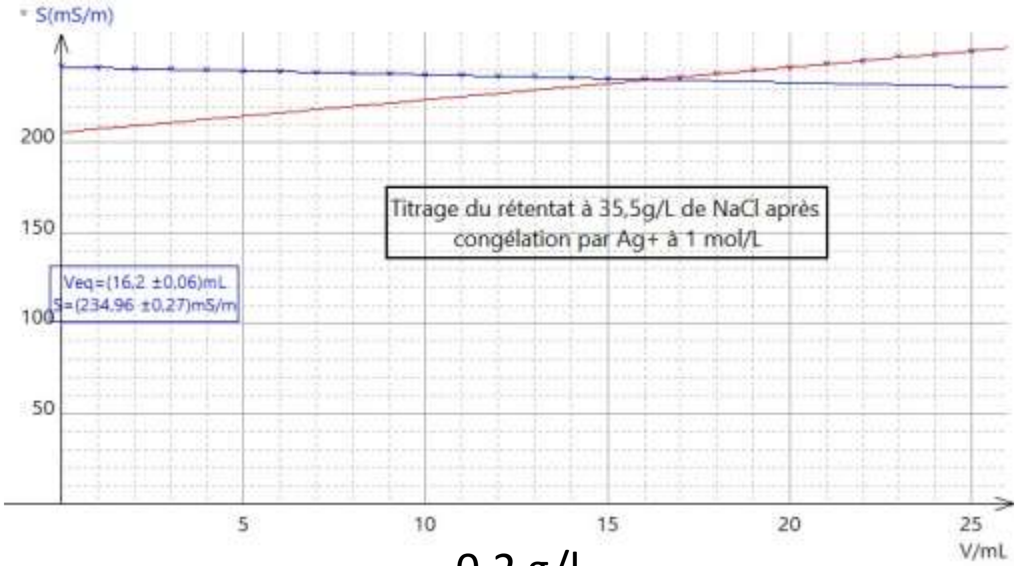
nombre de Péclet : $Pe = \frac{J\delta}{D} = \frac{transfert\ convectif}{transfert\ diffusif} \searrow \begin{cases} J \searrow \\ u \nearrow \delta \searrow \end{cases}$

Module du film $J_{stat} = \frac{D}{\delta}\ln\left(\frac{c_m - c_p}{c_0 - c_p}\right)$

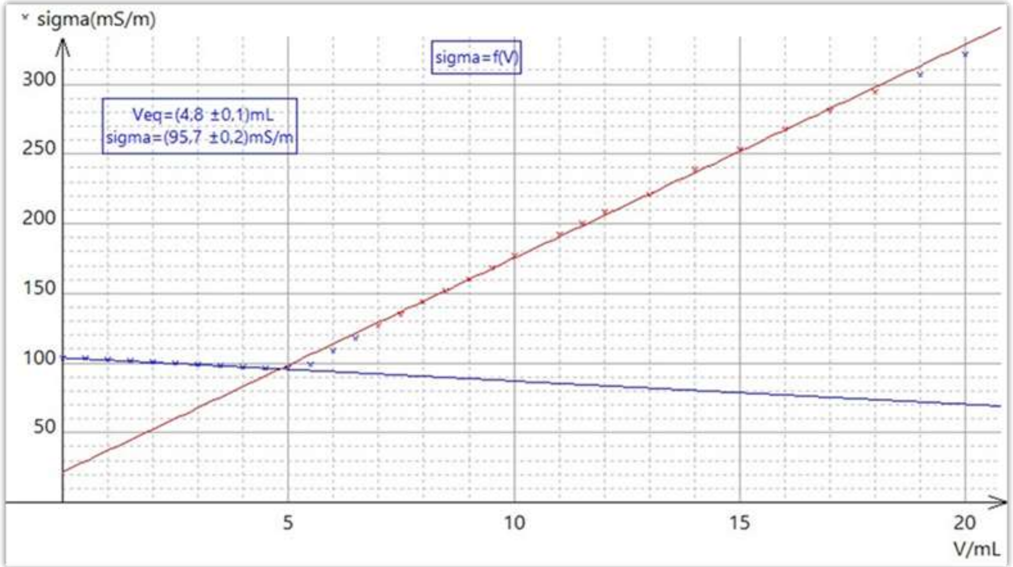
Module du gel $J_{lim} = \frac{D}{\delta}\ln\left(\frac{c_g - c_p}{c_0 - c_p}\right)$ où c_g est la concentration de gel

Annexe 11 :

Titrages des rétentats issues des congélations pour t=5min



0.2 g/l



35g/l