



La terre se réchauffe ,  
la glace fond

TIPE 2020/2021:  
Enjeux sociétaux.

# Introduction

Dans certaines régions, les phénomènes météorologiques extrêmes et les précipitations sont de plus en plus fréquents, tandis que d'autres sont confrontées à des vagues de chaleur et des sécheresses de plus en plus extrêmes. En effet, Le changement climatique est tellement rapide que beaucoup de plantes et d'espèces animales ont dû mal à s'y adapter. Ces effets ne devraient pas s'intensifier au cours des prochaines décennies.

# Les conséquences des changements climatiques qui justifient l'intérêt de ce sujet



La Fonte des glaciers: Le nombre d'ours polaires aura chuté de 30 % vers 2050.



Montée des eaux : l'effrayant impact du réchauffement climatique si rien ne change.



Le changement climatique est le principal responsable du [doublement des catastrophes naturelles](#) dans le monde en vingt ans



- ❖ Ces dernières années, les données notaient déjà une forte diminution de la quantité moyenne de glace en arctique.



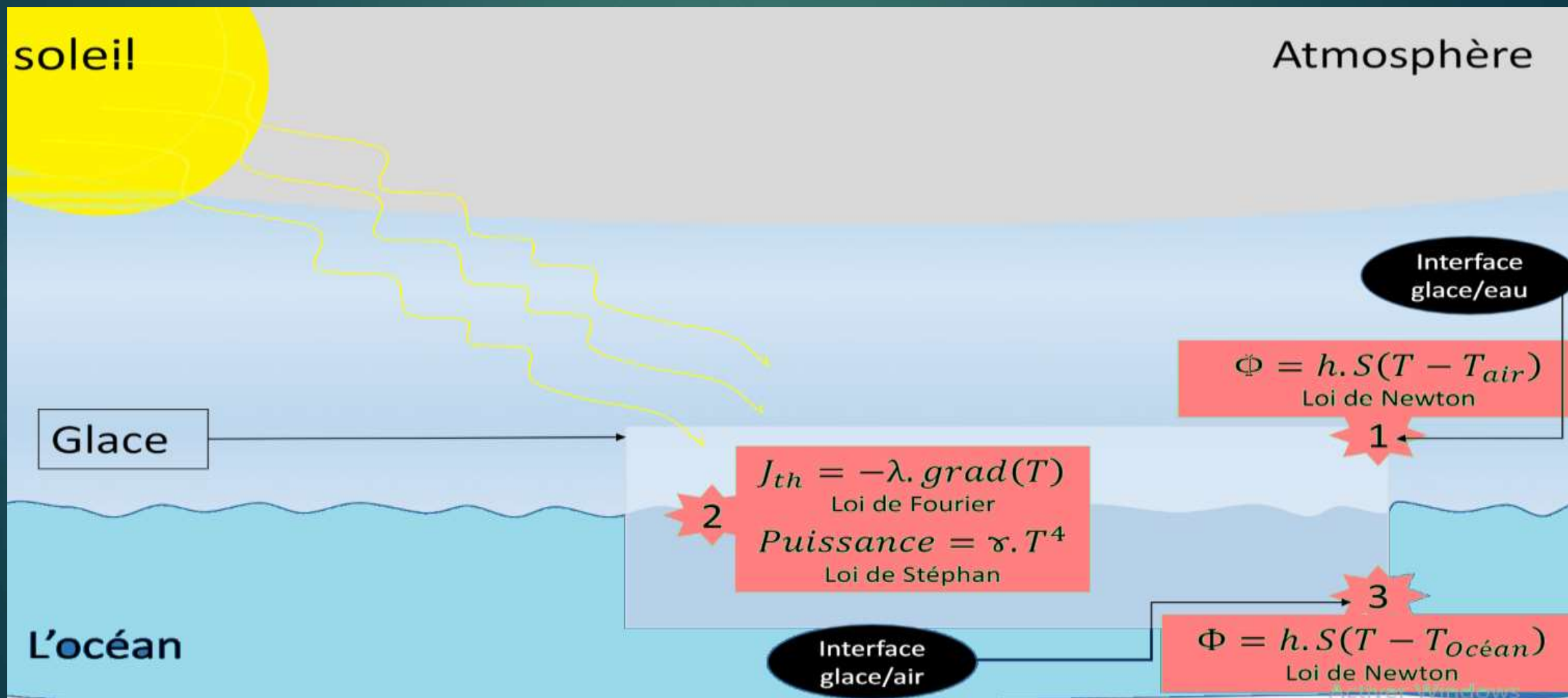
# Le plan

- ❖ modélisation des différents échanges thermiques entre un bloc de glace et son environnement avec l'équation de chaleur .
- ❖ Résolution numérique de l'équation de la chaleur .
- ❖ Modélisation avec les lois de newton et Stefan .
- ❖ exploitation d'une base de données et implémentation avec python.

# objectif

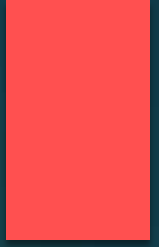
Le constat étant fait, on essaie donc d'inhiber les dangers. Afin d'appréhender ce fléau on va modéliser tout en utilisant l'équation de chaleur et une base de données.

# La modélisation des différents échanges thermiques entre un bloc de glace et son environnement avec l'équation de chaleur





# La Résolution analytique de l'équation de chaleur



On considère une calotte avec une température qui présente une allure gaussienne avec aucun effet extérieur du à une quelconque convection.

L'équation de chaleur s'écrit :

Avec :  $D = \frac{\lambda}{\rho C_p}$

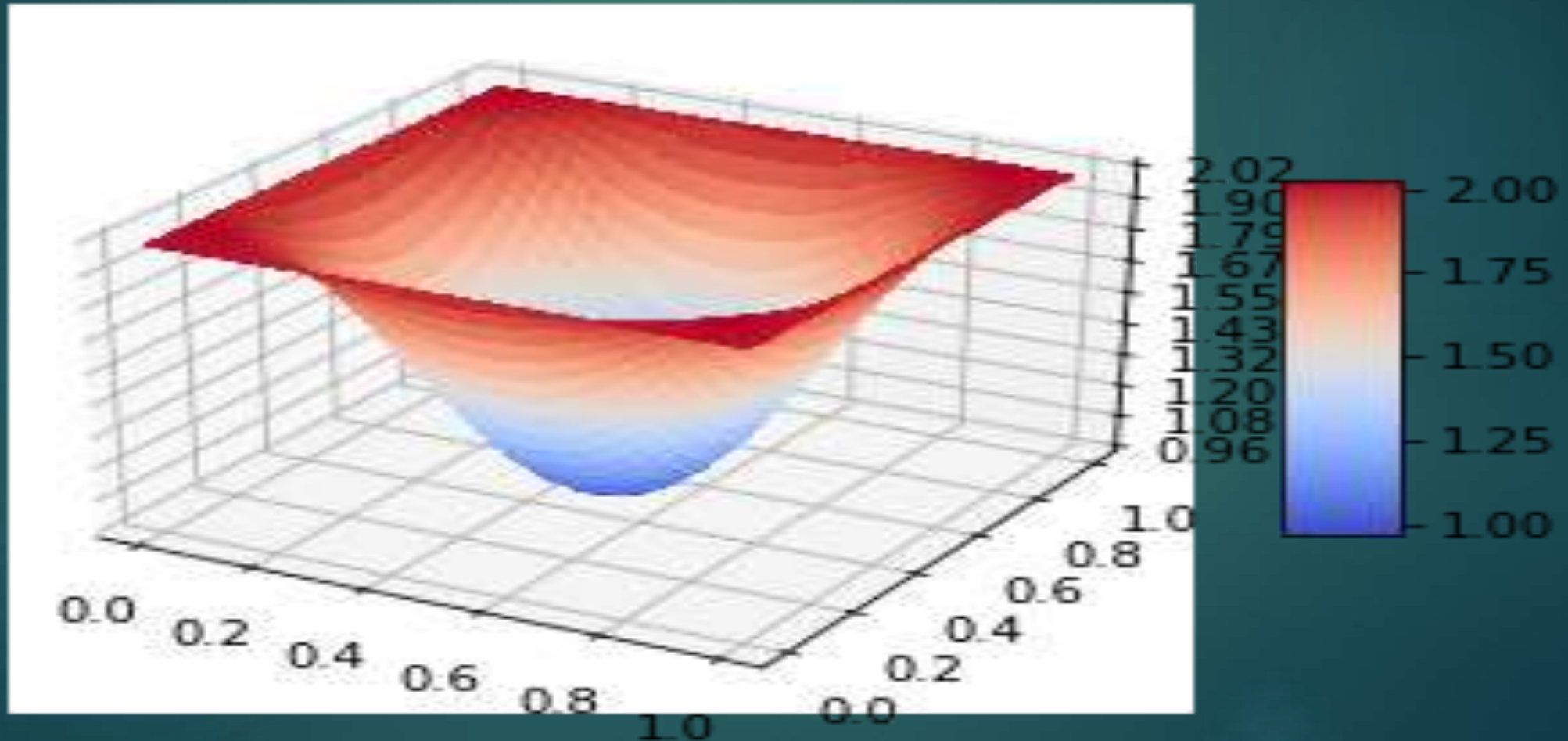
$$\frac{\partial \theta(x, y, t)}{\partial t} = D \left( \frac{\partial^2 \theta}{\partial x^2} + \frac{\partial^2 \theta}{\partial y^2} \right)$$

On résout cette équation et on obtient :

$$\theta(x, y, t) = \frac{1}{\sqrt{1 + \frac{2Dt}{\sigma}}} e^{-\frac{x^2 + y^2}{4Dt + 2\sigma}}$$



# La Résolution analytique de l'équation de chaleur




# Modélisation avec les lois de Newton et Stefan

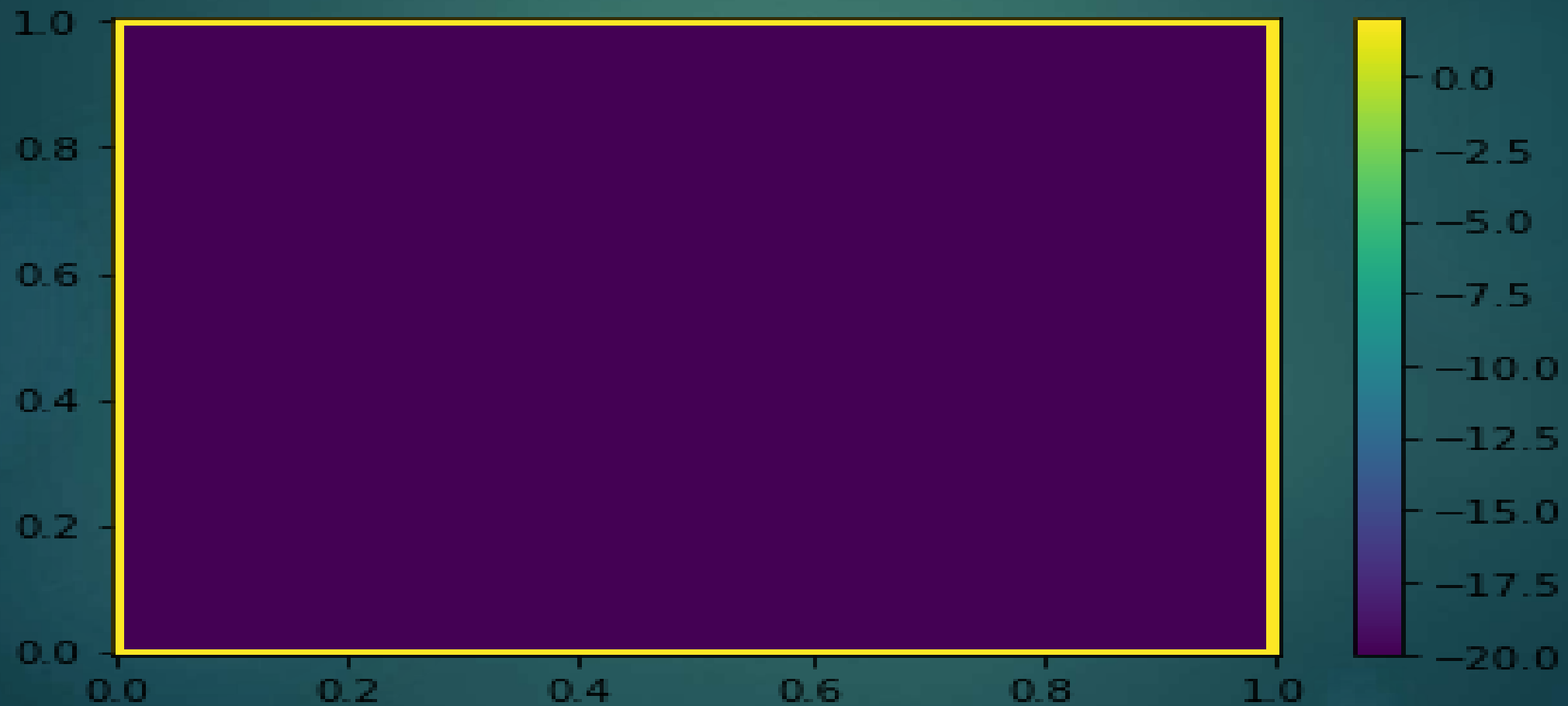
Le model précèdent ignore l'impact de la température extérieure , en fait, c'est un cas idéaliste et irréal .

L'équation à résoudre à présent se présente comme suit:

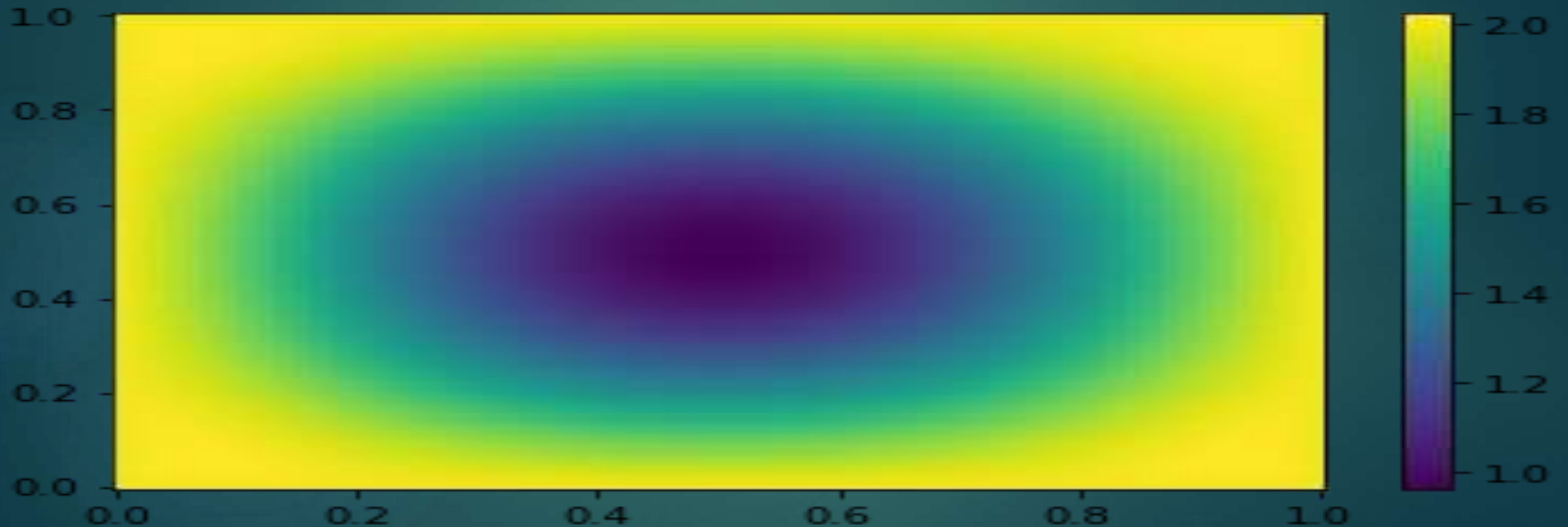
$$\frac{\partial \theta(x, y, t)}{\partial t} = D \left( \frac{\partial^2 \theta}{\partial x^2} + \frac{\partial^2 \theta}{\partial y^2} \right) + h(\theta - \theta_{air}) + \gamma \theta^4$$

- 
- ❖ Considérons un bloc de glace soumis aux phénomènes considérés dans l'équation.  
le glaçon commence par fondre par les bords .

# Modélisation avec les lois de newton et Stefan



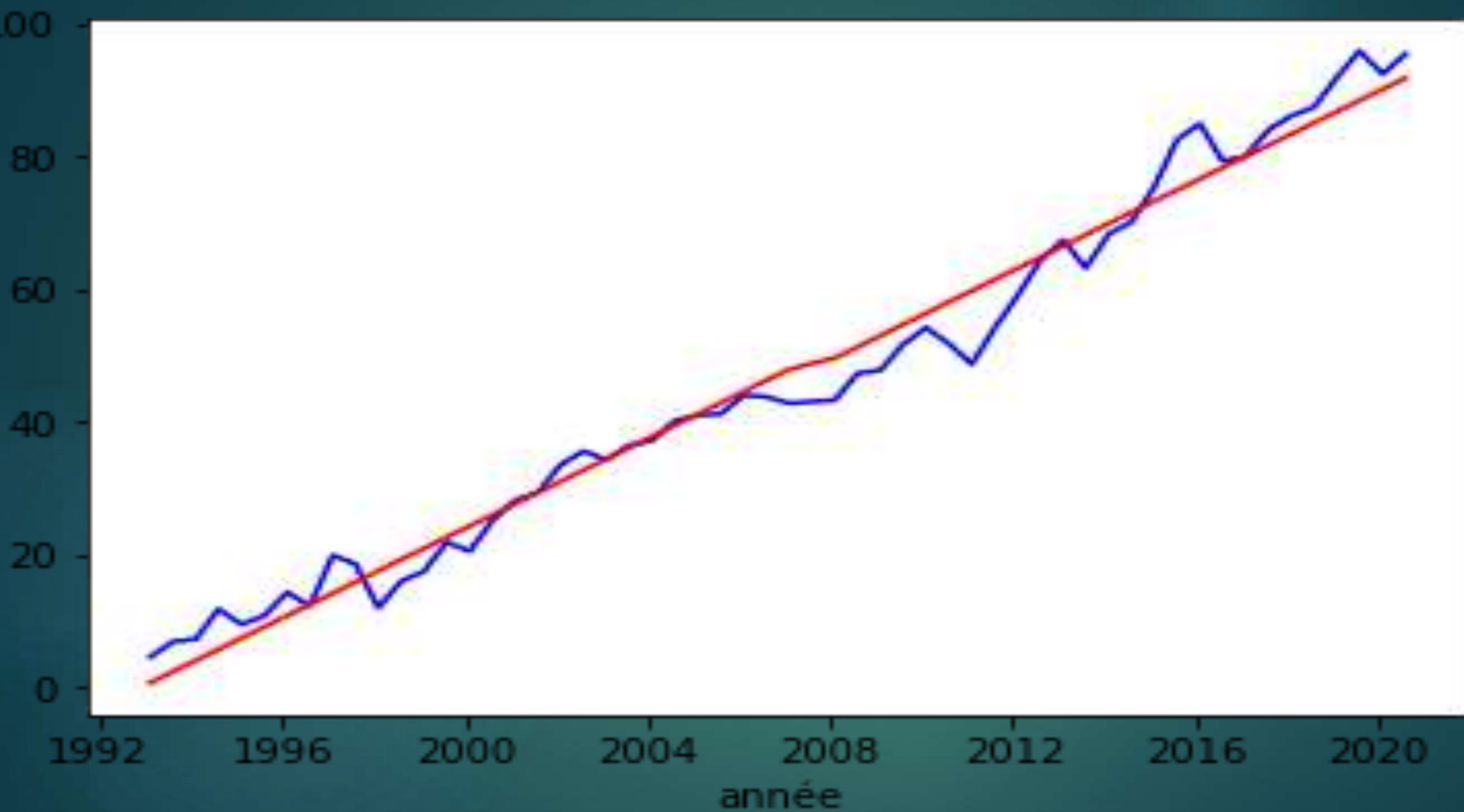
# Modélisation avec les lois de Newton et Stefan





# Exploitation d'une base de données et implémentation avec python

en se basant sur la base de données retrouvées qui contient l'ensemble des valeurs indiquant la variation de la hauteur de la mer en fonction du temps ( dans ce cas, la période étant 6 mois) la courbe représentée par ces points est :





# Annexe

# B. La résolution numérique de l'équation de la chaleur:

Entrée [1]: `pylab inline`

Populating the interactive namespace from numpy and matplotlib

Entrée [2]: `T = 0.1`  
`l = 1`  
`M = 101`  
`N = 4 * M ** 2`  
`xx = linspace(0, l, M)`  
`yy = linspace(0, 1, M)`  
`h = l / (M - 1)`  
`k = T / (N - 1)`  
`ll = k / h ** 2`

Entrée [3]: `gamma = 1`  
`H = 1`

Entrée [4]: `X, Y = meshgrid(xx, yy)`  
`U0 = zeros((M, M)) - 20`  
`U0[0, :] = 2 + zeros(M)`  
`U0[-1, :] = 2 + zeros(M)`  
`U0[:, 0] = 2 + zeros(M)`  
`U0[:, -1] = 2 + zeros(M)`

# B. La résolution numérique de l'équation de la chaleur:

```
Entrée [5]: e = ones(M)
            ee = ones(M - 1)
            LM = -2 * diag(e) + diag(ee, 1) + diag(ee, -1)
```

```
Entrée [6]: V = U0
            for i in range(N):
                V = V + 11 * (dot(LM, V) + dot(V, LM)) + k * (H * V + gamma * V ** 4)
                V[0, :] = 2 + zeros(M)
                V[-1, :] = 2 + zeros(M)
                V[:, 0] = 2 + zeros(M)
                V[:, -1] = 2 + zeros(M)
```

```
Entrée [7]: fig, ax = subplots(subplot_kw={"projection": "3d"})
            surf      = ax.plot_surface(X, Y, V, cmap=cm.coolwarm,
                                       linewidth=0, antialiased=False)

            # Customize the z axis.
            ax.zaxis.set_major_locator(LinearLocator(10))

            # A StrMethodFormatter is used automatically
            ax.zaxis.set_major_formatter('{x:.02f}')

            # Add a color bar which maps values to colors.
            fig.colorbar(surf, shrink=0.5, aspect=5)
```



# C. La résolution numérique de l'équation de la chaleur avec les lois de Newton et Stefan

Entrée [1]: `pylab inline`

Populating the interactive namespace from numpy and matplotlib

Entrée [2]: `T = 0.1`  
`l = 1`  
`M = 101`  
`N = 4 * M ** 2`  
`xx = linspace(0, l, M)`  
`yy = linspace(0, 1, M)`  
`h = l / (M - 1)`  
`k = T / (N - 1)`  
`ll = k / h ** 2`

Entrée [3]: `gamma = 1`  
`H = 1`

Entrée [4]: `X, Y = meshgrid(xx, yy)`  
`U0 = zeros((M, M)) - 20`  
`U0[0, :] = 2 + zeros(M)`  
`U0[-1, :] = 2 + zeros(M)`  
`U0[:, 0] = 2 + zeros(M)`  
`U0[:, -1] = 2 + zeros(M)`

# C. La résolution numérique de l'équation de la chaleur avec les lois de Newton et Stefan

```
Entrée [5]: e = ones(M)
            ee = ones(M - 1)
            LM = -2 * diag(e) + diag(ee, 1) + diag(ee, -1)
```

```
Entrée [6]: V = U0
            for i in range(N):
                V = V + 11 * (dot(LM, V) + dot(V, LM)) + k * (H * V + gamma * V ** 4)
                V[0, :] = 2 + zeros(M)
                V[-1, :] = 2 + zeros(M)
                V[:, 0] = 2 + zeros(M)
                V[:, -1] = 2 + zeros(M)
```

```
Entrée [7]: fig, ax = subplots(subplot_kw={"projection": "3d"})
            surf = ax.plot_surface(X, Y, V, cmap=cm.coolwarm,
                                   linewidth=0, antialiased=False)

            # Customize the z axis.
            ax.zaxis.set_major_locator(LinearLocator(10))

            # A StrMethodFormatter is used automatically
            ax.zaxis.set_major_formatter('{x:.02f}')

            # Add a color bar which maps values to colors.
            fig.colorbar(surf, shrink=0.5, aspect=5)
```

## C. La résolution numérique de l'équation de la chaleur avec les lois de Newton et Stefan

```
Entrée [5]: e = ones(M)
            ee = ones(M - 1)
            LM = -2 * diag(e) + diag(ee, 1) + diag(ee, -1)
```

```
Entrée [6]: V = U0
            for i in range(N):
                V = V + ll * (dot(LM, V) + dot(V, LM)) + k * (H * V + gamma * V ** 4)
                V[0, :] = 2 + zeros(M)
                V[-1, :] = 2 + zeros(M)
                V[:, 0] = 2 + zeros(M)
                V[:, -1] = 2 + zeros(M)
```

```
Entrée [7]: #1ère courbe
            im1 = pcolormesh(X, Y, U0, shading='auto')
            colorbar(im1)
            show()
            #2ème courbe
            im2 = pcolormesh(X, Y, V, shading='auto')
            colorbar(im2)
            show()
```

# D. La base de données

la variation de la hauteur de la mer	date				
1,4	07/02/1993	33,5	07/08/2002	58,9	07/02/2012
4,5	07/08/1993	35,5	07/02/2003	64,2	07/08/2012
6,8	07/02/1994	34,2	07/08/2003	67,3	07/02/2013
7,2	07/08/1994	36,4	07/02/2004	63,1	07/08/2013
11,7	07/02/1995	37,1	07/08/2004	68,2	07/02/2014
9,4	07/08/1995	40,1	07/02/2005	70	07/08/2014
10,7	07/02/1996	40,9	07/08/2005	75,3	07/02/2015
14,2	07/08/1996	41,1	07/02/2006	82,4	07/08/2015
12,2	07/02/1997	43,9	07/08/2006	84,8	07/02/2016
19,8	07/08/1997	43,7	07/02/2007	79,2	07/08/2016
18,5	07/02/1998	42,4	07/08/2007	79,9	07/02/2017
11,9	07/08/1998	43,2	07/02/2008	83,9	07/08/2017
15,9	07/02/1999	47,2	07/08/2008	86	07/02/2018
17,4	07/08/1999	47,7	07/02/2009	87,3	07/08/2018
21,8	07/02/2000	51,6	07/08/2009	91,8	07/02/2019
20,4	07/08/2000	54,1	07/02/2010	95,8	07/08/2019
25,1	07/02/2001	51,7	07/08/2010	92,3	07/02/2020
28,1	07/08/2001	48,6	07/02/2011	95,3	07/08/2020
29,3	07/03/2002	54	07/08/2011		
		58,9	07/02/2012		

# Conclusion

Pour des blocs de glace de dimension de même ordre de grandeur, la glace fond lorsqu'on rajoute les termes de Newton et de Stefan dans une condition similaire à un changement climatique. L'influence est claire.

Etant conscient des dangers qui menacent notre planète liés aux glaciers, on doit tous faire pour prédire leurs évolutions futures et d'éviter les catastrophes.

Finalelement cette étude a été fort instructive pour mon statut d'étudiant car cela m'a permis de manipuler des équations assez complexes et découvrir certaines méthodes de résolutions numériques d'équations.