

# prog-tumeur

January 14, 2020

## Labo Maths 2019-2020

Atelier Modélisation : Modélisation de l'impact d'anticorps dans une tumeur

23 Janvier 2020

Nous commençons à charger les librairies Python usuelles

```
In [1]: from math import *
import numpy as np
import scipy as sc
import scipy.linalg as sl
import matplotlib.pyplot as plt
```

Puis nous débutons le programme en spécifiant les données

```
In [2]: #####
# PARAMETRES PHYSIQUES
#####
# Longueur du domaine
L=4.0
# Temps final de simulation
T=10.0
# Vitesse de transport
u=0.4
# Valeur de omega
omega=0.9
# Valeur de nu
nu=0.01
# Valeur de k
k=1.0
# Valeur de p
p=5

#####
# Conditions aux limites et initiales
#####
# Condition aux limites : anticorps en x=0
def cd(t):
    return t*exp(-t/2)
CD=sc.vectorize(cd)
```

```

# Condition initiale des anticorps
def c0(x):
    return 0.0
C0=sc.vectorize(c0)
# Condition initiale des antigènes
def s0(x):
    return 10.0
S0=sc.vectorize(s0)

#####
# PARAMETRES NUMERIQUES
#####
# Nombre de mailles
NX=200
# Nombre de pas de temps
NT=1000

#####
# PARAMETRES DEDUITS (Ne pas modifier ce paragraphe)
#####
# Pas d'espace
DELTAX=L/NX
# Nombre de pas de temps
DELTAT=T/NT
# Condition CFL
CFL=DELTAT/DELTAX

#####
# PARAMETRES DE VISUALISATION
#####
# Delta temps affichage pour voir la solution
DTAFFICHAGE=1.0

```

On initialise maintenant les valeurs des vecteurs C et S représentant les solutions discrètes des variables c et s

```

In [3]: # Vecteur contenant les abscisses des points  $x_i$ 
x=np.linspace(0,L,NX+1)
# Initialisation du temps, Initialisation de C et de S,
temps=0.0
tempsaffichage=0.0
C=C0(x)
S=S0(x)
CNEW=np.zeros(np.size(C))
SNEW=np.zeros(np.size(S))

```

On écrit alors la boucle en temps

```

In [4]: # Initialisation des figures
plt.figure(1)

```

```

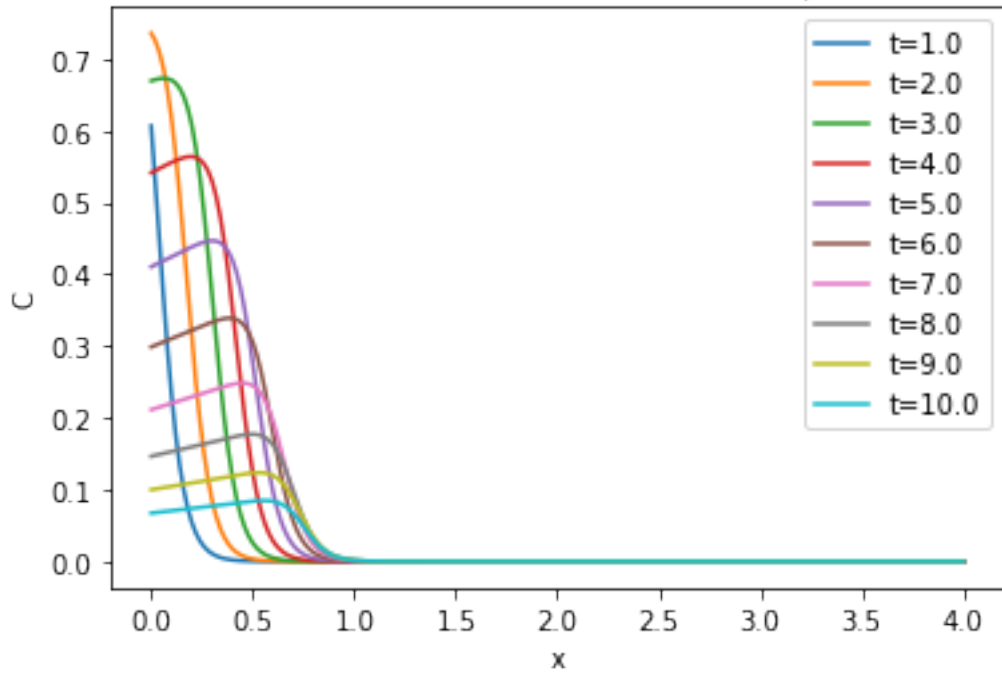
plt.title('Solution, concentration d anticorps')
plt.xlabel('x')
plt.ylabel('C')

plt.figure(2)
plt.title('Solution, concentration d antigenes')
plt.xlabel('x')
plt.ylabel('S')

# Boucle sur le temps
while (temps<T-DELTAT):
    temps=temps+DELTAT
    tempsaffichage=tempsaffichage+DELTAT
    CNEW[0]=cd(temps)
    CNEW[NX]=0.0
    for i in range(1,NX):
        CNEW[i]=C[i]-u*DELTAT/(DELTAX*omega)*(C[i]-C[i-1]) \
                +nu*DELTAT/(DELTAX**2*omega)*(C[i+1]-2*C[i]+C[i-1]) \
                -k*DELTAT/omega*C[i]*S[i]
    for i in range(0,NX+1):
        SNEW[i]=S[i]*(1.0-DELTAT*p*k*C[i])
    C=CNEW
    S=SNEW
    if (tempsaffichage > DTAFFICHAGE):
        tempsaffichage=0.0
        plt.figure(1)
        chaine='t='+str(round(temps,2))
        plt.plot(x,C,label=chaine)
        plt.legend(loc='upper right')
        plt.figure(2)
        chaine='t='+str(round(temps,2))
        plt.plot(x,S,label=chaine)
        plt.legend(loc='upper right')

```

Solution, concentration d anticorps



Solution, concentration d antigenes

