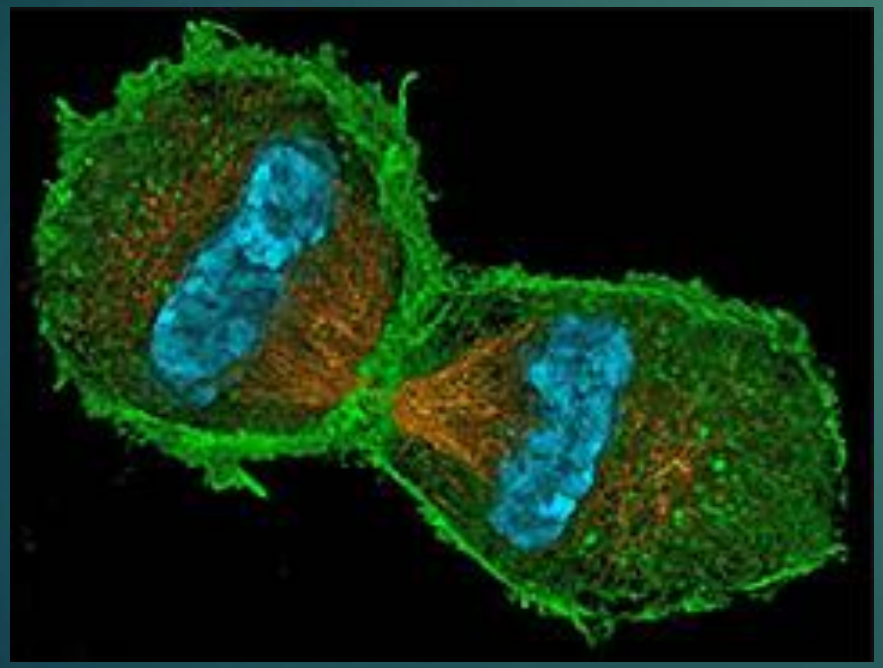
TIPE: LA MODELISATION MATHEMATIQUE DANS L'ONCOLOGIE

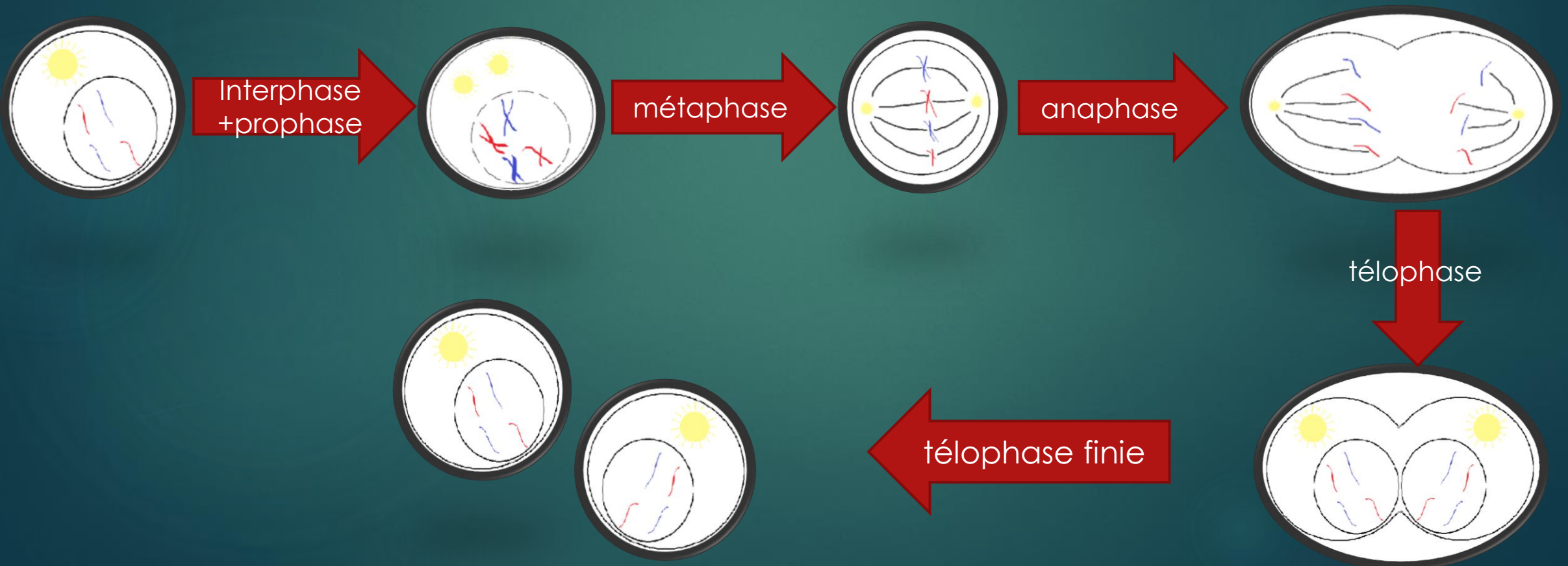


TRAVAIL FAIT PAR MOHAMED FAROUK.

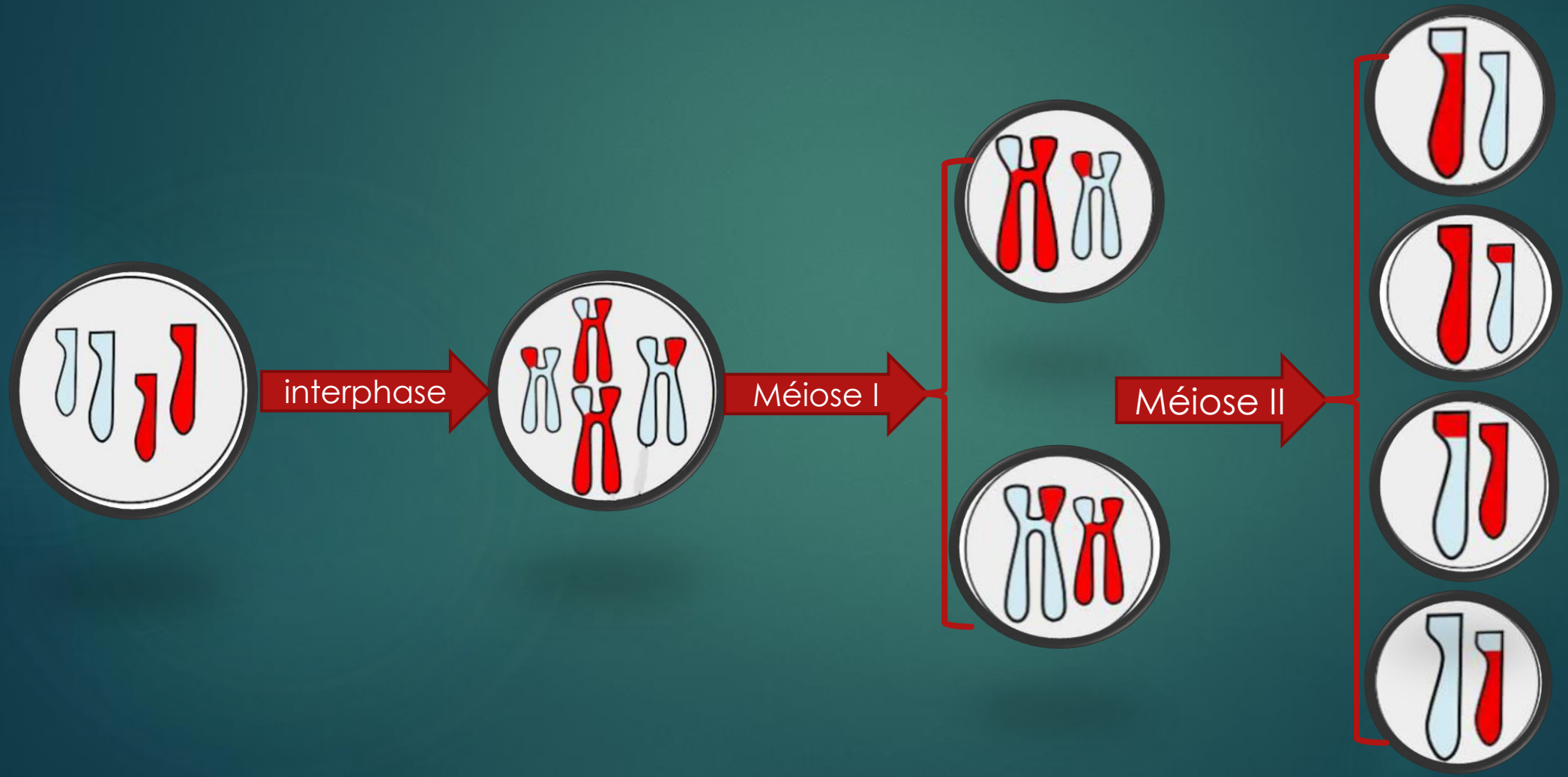
SPÉCIALITÉ: SCIENCES DE L'INGÉNIEUR.

Le cancer, une maladie complexe

- La mitose: c'est un mécanisme de reproduction cellulaire qui consiste à la production de deux cellules filles identiques à une cellule mère.



- La méiose: Il s'agit d'une division cellulaire au cours de laquelle il y a formation de quatre cellules filles haploïdes d'une cellule diploïde.



Sources de stochasticité dans la croissance tumorale:

i) le phénomène d'apoptose

ii) le phénomène de division cellulaire

iii) le phénomène de mutation génétique

iv) la mort ou la persistance d'une cellule cancéreuse dans un organisme sous traitement.

PLAN

I. Construction d'un modèle stochastique décrivant la croissance tumorale.
II. Construction du modèle déterministe du traitement (exemple: traitement par transfert adoptif de cellules)
III. Simulation spatiotemporelle par des automates cellulaires de la croissance tumorale sous traitement.

I.Construction d'un modèle stochastique décrivant la croissance tumorale

- Soit $(t_n)_{n>0}$ une suite de réels strictement croissante représentant des instants successifs. On note $M(t_k)$ la variable aléatoire donnant le nombre de cellules à l'instant t_k. Alors, $(M(t_n))_{n>0}$ est un processus markovien, c'est-à-dire:

$$\forall n \in \mathbb{N}\ ,\ \forall x_1, ...x_n, y \in \mathbb{N}^*\ ,$$
$$\mathbb{P}(M(t_{n+1}) = y | M(t_n) = x_n, ..., M(t_1) = x_1) = \mathbb{P}(M(t_{n+1}) = y | M(t_n) = x_n).$$

- Processus markovien de saut à temps continu: c'est un processus au cours duquel l'état est constant entre deux sauts consécutifs. Les instants des sauts sont aléatoires.
  Choix du modèle des instants des sauts: les évènements décrivant la croissance tumorale seront régis par un loi de probabilité exponentielle dont les taux seront des paramètres du modèle.

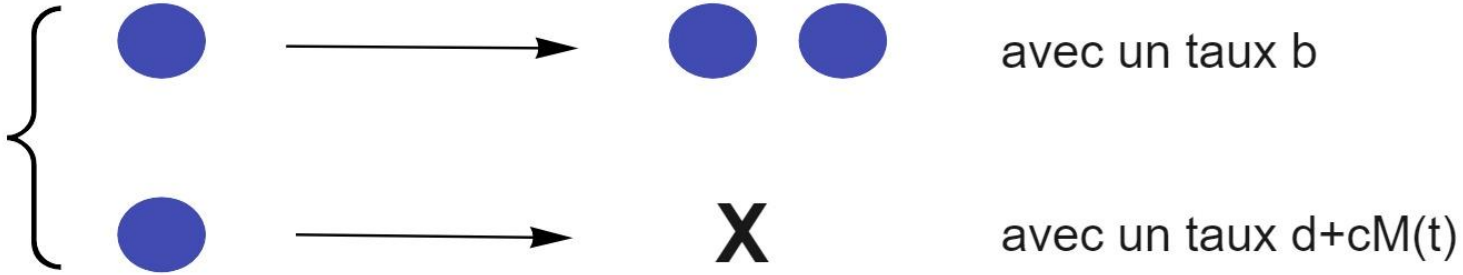
Processus de naissance et de mort logistique

- Définition : *PROCESSUS DE NAISSANCE ET DE MORT:* Le processus $\{M(t)\}_{t \geq 0}$ est un processus de naissance et de mort s'il correspond à un processus de Markov à temps continu à valeurs dans IN telles que les seules transitions possibles à partir d'un état M soient les états M + 1 ou M − 1.

A l'échelle de la cellule, Modèle individu centré:

On symbolise une cellule cancéreuse par  et une cellule morte par **X**.

On note M(t) le nombre de cellules cancéreuses à l'instant t. Les transitions possibles sont:



avec un taux b

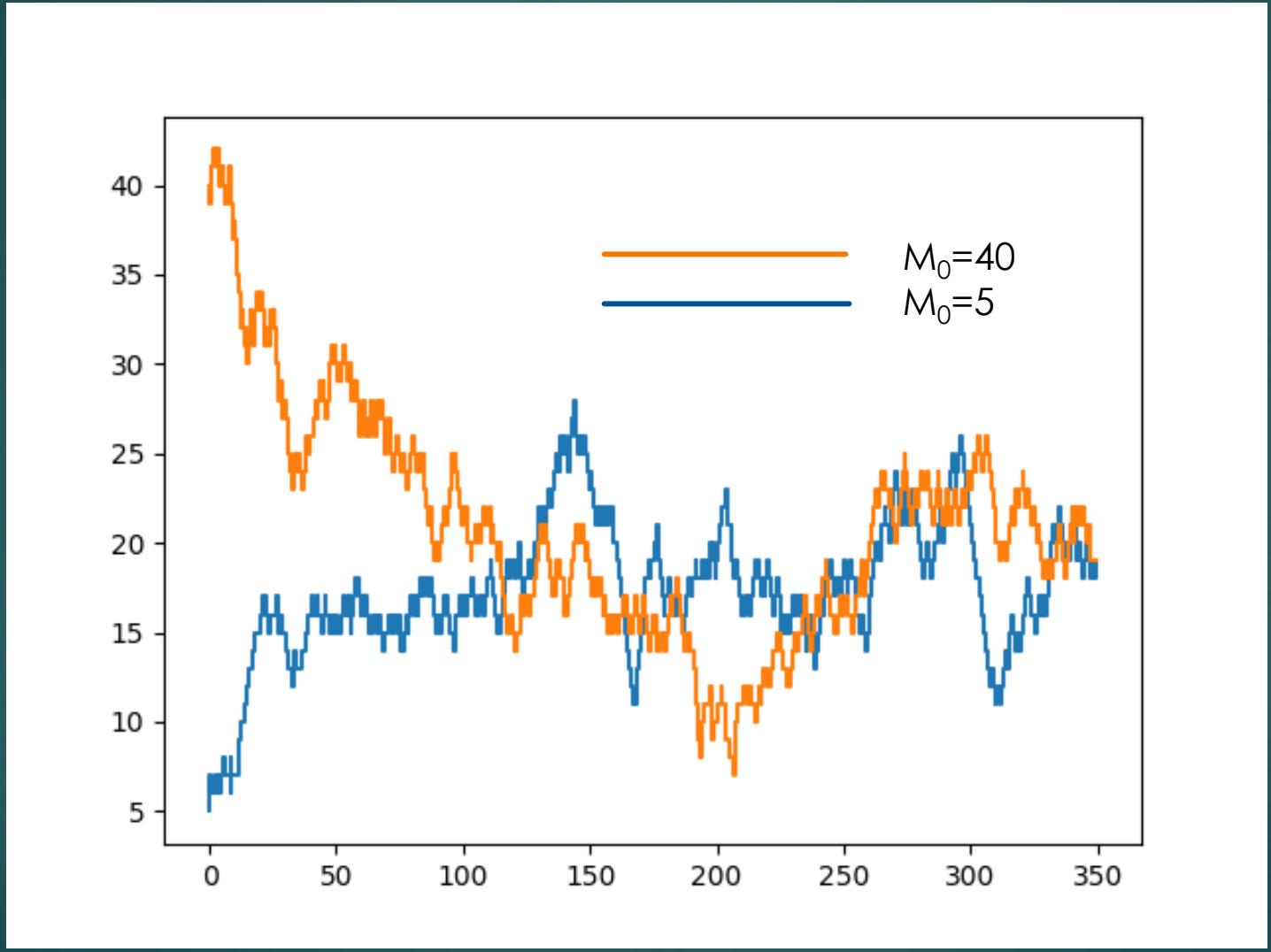
avec un taux d+cM(t)

A l'échelle de la tumeur vue comme une population d'individus(cellules):

On note M(t) le nombre de cellules cancéreuses à l'instant t. Les transitions possibles sont:

$$\begin{cases} M(t) \longrightarrow M(t)+1 & \text{avec un taux } bM(t) \\ M(t) \longrightarrow M(t)-1 & \text{avec un taux } (d+cM(t))\,M(t) \end{cases}$$

Résultat de la simulation



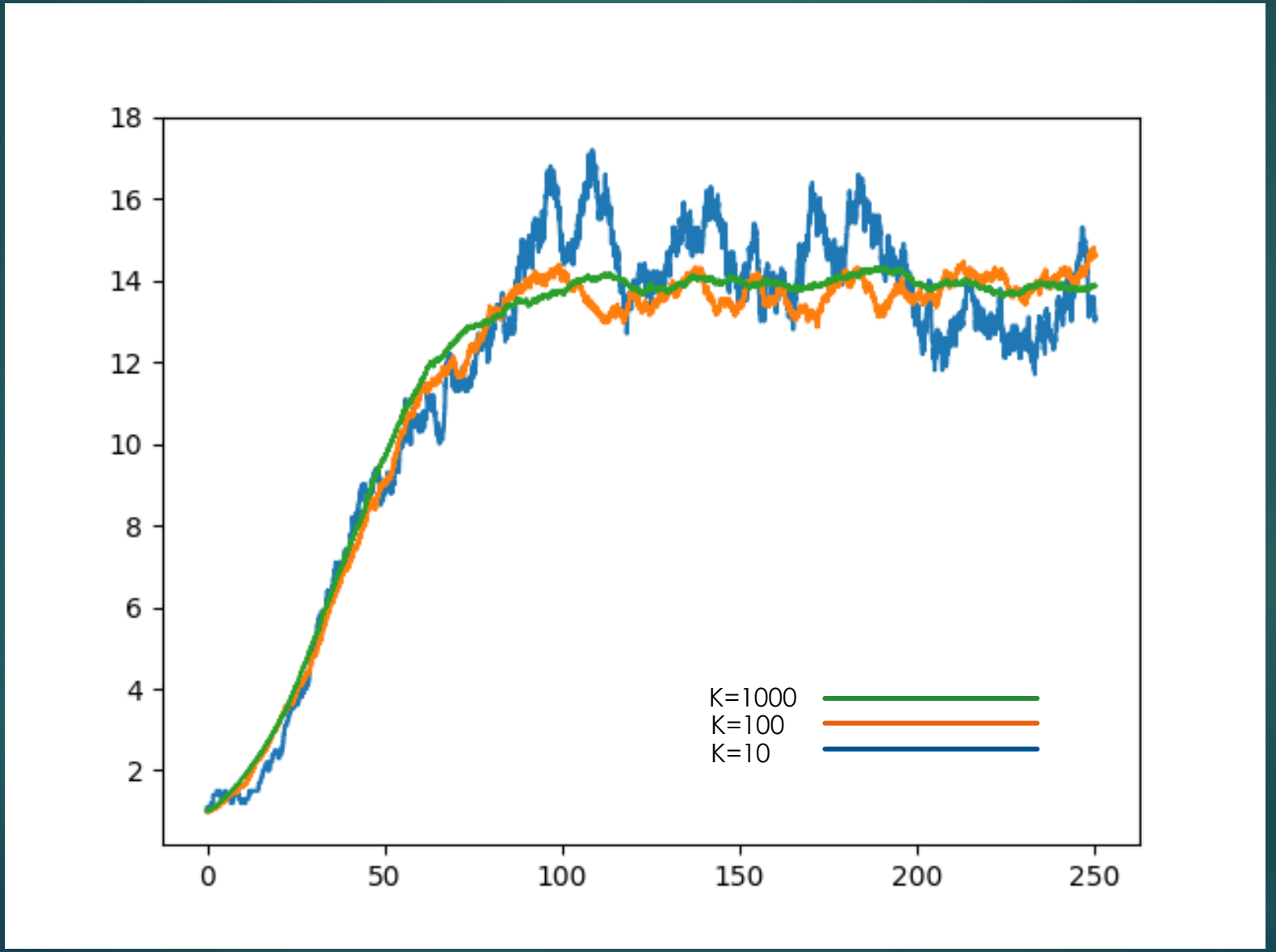
- Afin de diminuer les fluctuation autour du point fixe attractif, on utilise l'approximation en grande population.

Approximation en grande population du processus de naissance et de mort logistique:

Soit $K \in \mathbb{N}$ un entier non nul fixé. A l'instant t, considérons la quantité $M_k(t) = M(t)/k$ (redimensionnement par k de la taille de la population). Les transitions possibles de $M_k(t)$ sont données par :

$$\begin{cases} M_k(t) \longrightarrow (M(t) + 1)/K & \text{avec un taux } bKM_k(t) \\ M_k(t) \longrightarrow (M(t) - 1)/K & \text{avec un taux } (d + c\ M_k(t))\ K\ M_k(t) \end{cases}$$

Résultat de la simulation pour différentes valeurs de K



Nécessité d'un modèle déterministe

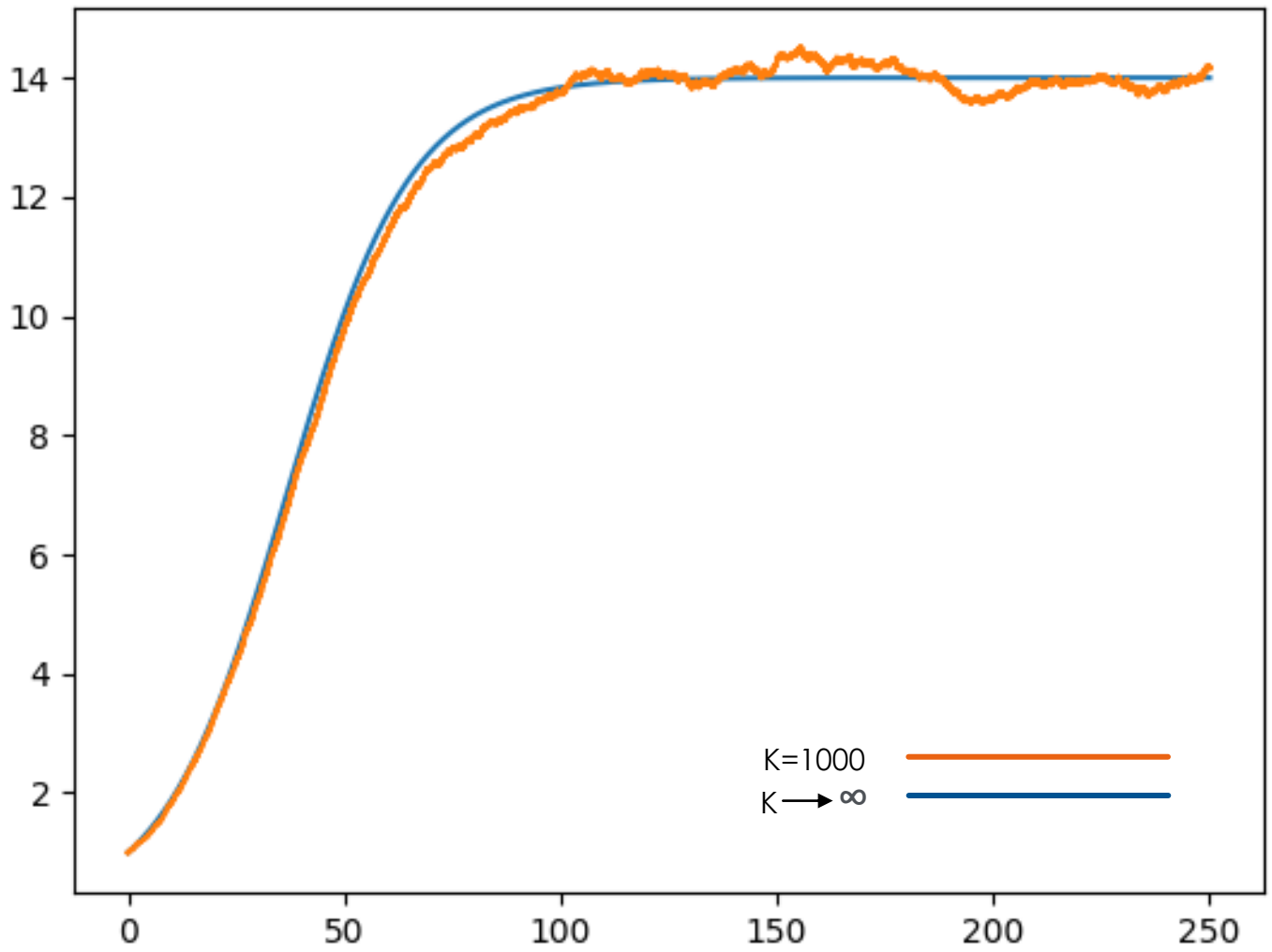
- Modèle déterministe: c'est un modèle dans lequel, partant d'un état initial, on aboutit à un seul état final, parfaitement connu sans aucun phénomène aléatoire.
- Théorème:

On suppose que : $lim_{k\mapsto\infty} M_k(0) \overset{d}{=} n_0$, où n_0 constitue une limite en loi supposée déterministe. On suppose aussi que b, c et d sont des réels non nuls. Alors, sur tout segment $[0, t_F]$, la suite des processus $(M_K(t))_{K\in\mathbb{N}^*}$ converge en loi :

$lim_{k\mapsto\infty}(M_k(t))_{t\in[0,t_F]} \overset{d}{=} (n(t))_{t\in[0,t_F]}$

tel que n soit solution de l'équation différentielle : $\dot{n}(t) = (b - d - cn(t))n(t)$ avec la condition initiale n_0.

Résultat de la simulation



II. Construction du modèle déterministe du traitement (exemple: traitement par transfert adoptif de cellules)

On note :

1) M(t): le nombre de cellules différenciées à l'instant t

2) D(t): le nombre de cellules dédifférenciées à l'instant t

3) T(t): le nombre de cellules T cytotoxiques à l'instant t

Les transitions possibles

Croissance tumorale:

Reproduction de cellules différenciées : $(M(t),D(t),T(t)) \xrightarrow{b_M M(t)} (M(t)+1,D(t),T(t))$

Mort de cellules différenciées : $(M(t), D(t), T(t)) \xrightarrow{(d_M + c_{MM} M(t) + c_{MD} D(t))\, M(t)} (M(t) - 1, D(t), T(t))$

Reproduction de cellules dédifférenciées : $(M(t), D(t), T(t)) \xrightarrow{b_D D(t)} (M(t), D(t) + 1, T(t))$

Mort de cellules dédifférenciées : $(M(t), D(t), T(t)) \xrightarrow{(d_D + c_{DD} D(t) + c_{DM} M(t))D(t)} (M(t), D(t) - 1, T(t))$

Changement de phénotype : $(M(t), D(t), T(t)) \xrightarrow{s_{MD} M(t)} (M(t) - 1, D(t) + 1, T(t))$

Changement de phénotype: $(M(t), D(t), T(t)) \xrightarrow{s_{DM} D(t)} (M(t) + 1, D(t) - 1, T(t))$

Thérapie ACT:

Reproduction de cellules T : $(M(t),D(t),T(t)) \xrightarrow{b_T\, M(t)T(t)} (M(t),D(t),T(t)+1)$

Mort de cellules T : $(M(t), D(t), T(t)) \xrightarrow{d_T\, T(t)} (M(t) , D(t), T(t)-1)$

Changement de phénotype induit par la thérapie : $(M(t), D(t), T(t)) \xrightarrow{s_A\, M(t)} (M(t)-1, D(t) + 1, T(t))$

Mort de cellules différenciées induite par thérapie : $(M(t), D(t), T(t)) \xrightarrow{d_M^{[T]}\, M(t)T(t)} (M(t)-1, D(t) , T(t))$

Modèle déterministe final

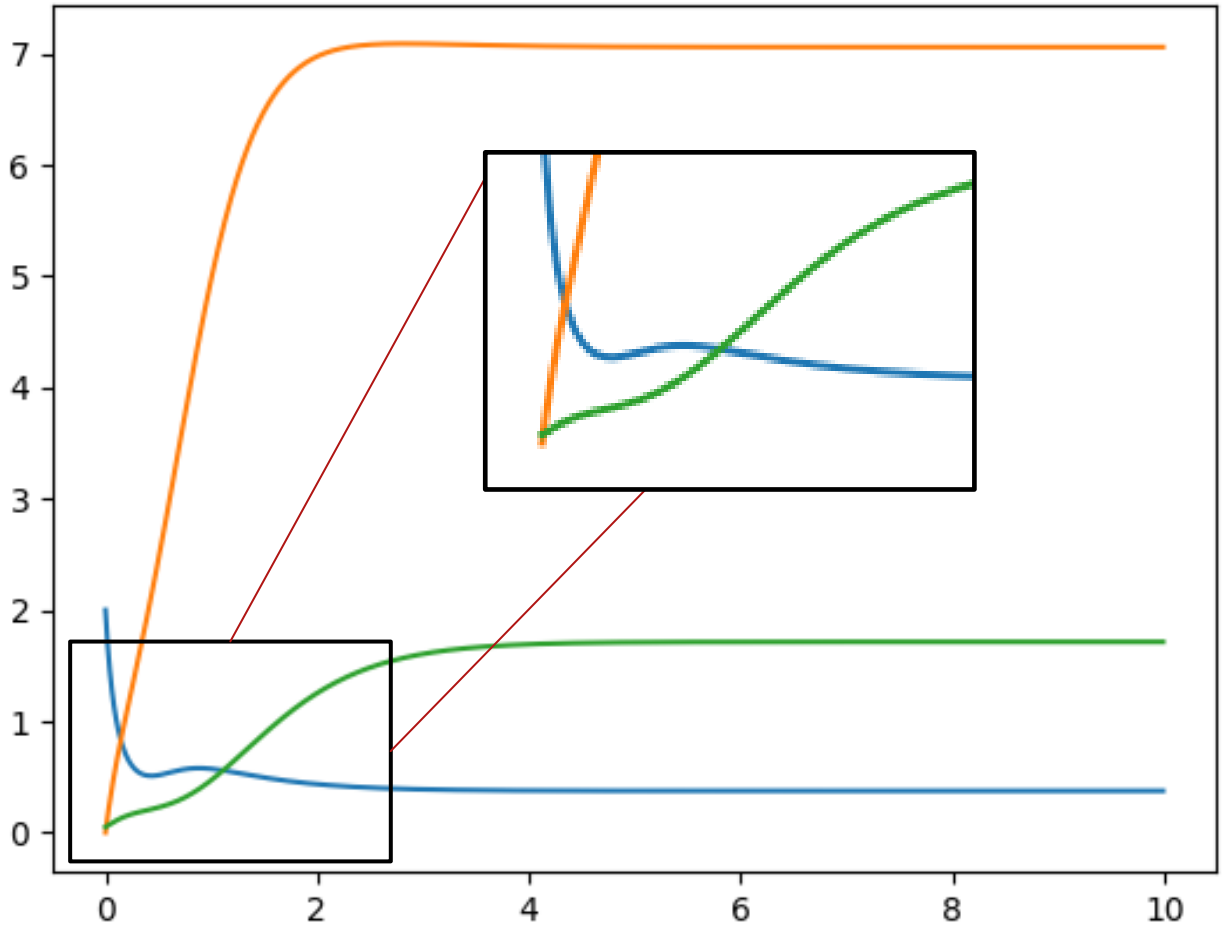
- En utilisant le théorème en annexe 4, on obtient le système d'équations différentielles:

$$\begin{cases} \dot{n_M}(t) = (b_M - d_M - c_{MM}n_M(t) - c_{MD}n_D(t) - s_{MD} - (s_A + d_M^{[T]})n_T)n_M(t) + s_{DM}n_D(t) \\ \dot{n_D}(t) = (b_D - d_D - c_{DD}n_D(t) - c_{DM}n_M(t) - s_{DM})n_D(t) + s_{MD}n_M(t) + s_An_M(t) \\ \dot{n_T}(t) = (b_Tn_M(t) - d_T)n_T(t) \end{cases}$$

Avec :

1. n_M est la limite déterministe donnant le nombre de cellules différenciées.
2. n_D est la limite déterministe donnant le nombre de cellules dédifférenciées.
3. n_T est la limite déterministe donnant le nombre de cellules T.

Résultat de la simulation

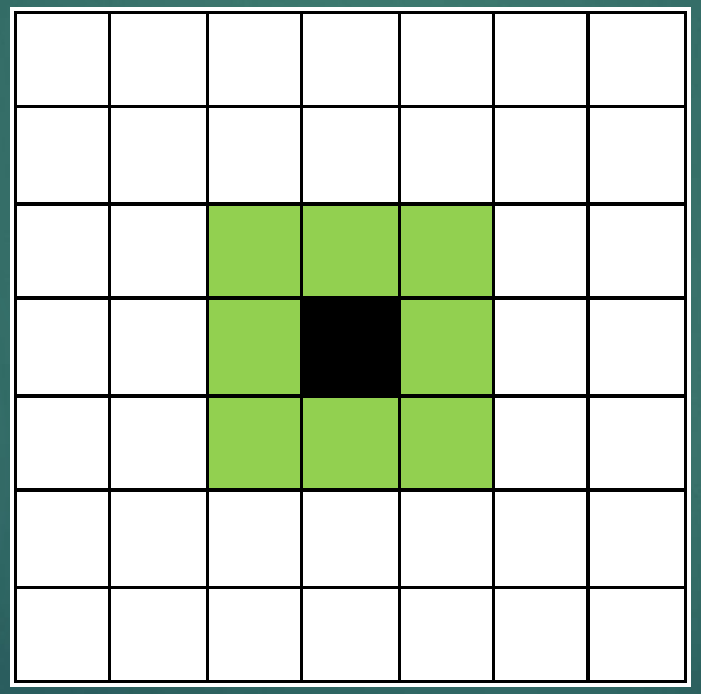


n_M

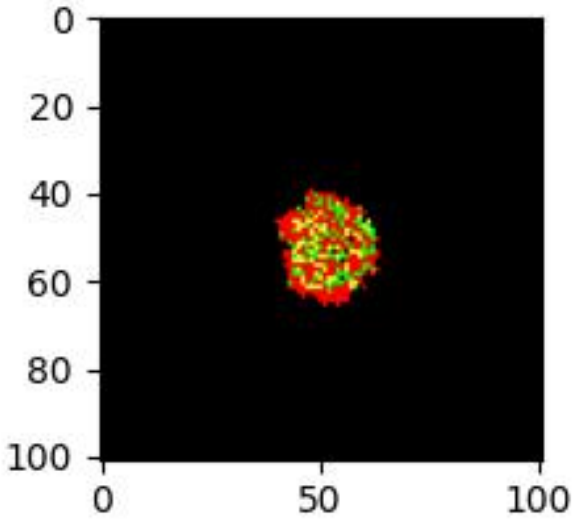
n_D

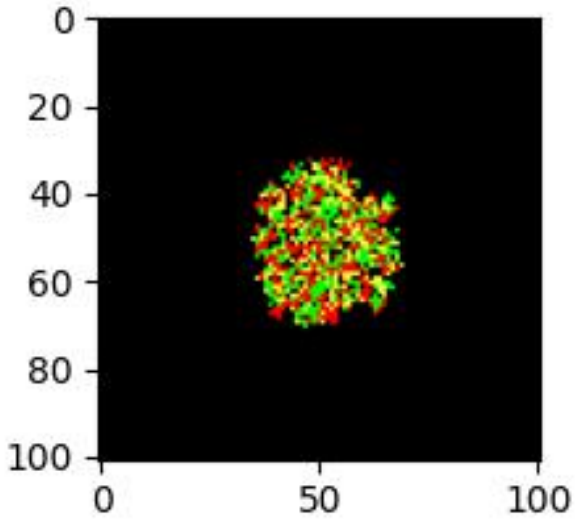
n_T

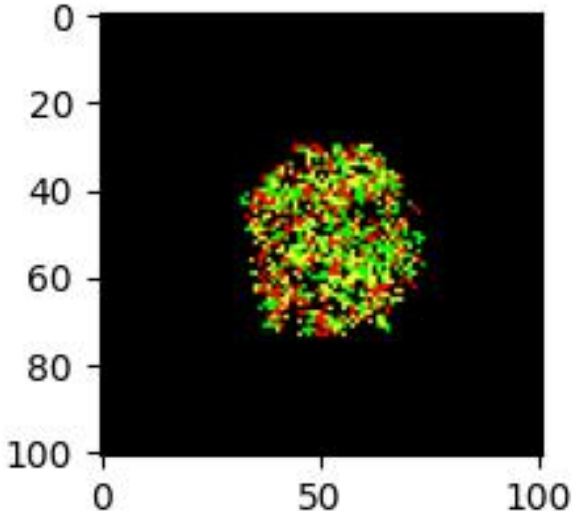
III. Simulation spatiotemporelle par des automates cellulaires de la croissance tumorale sous traitement

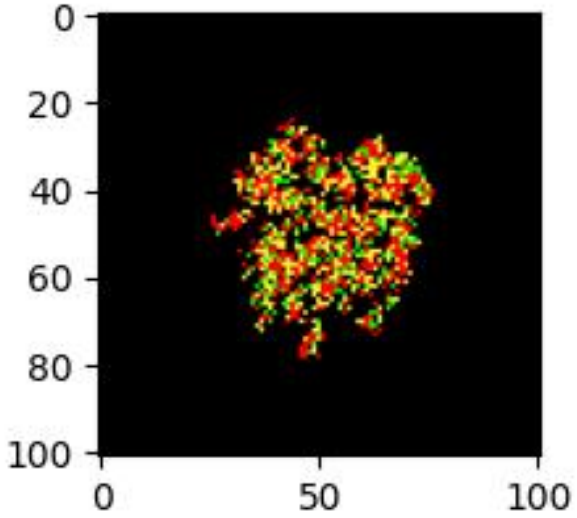


Résultats de la simulation









Annexe 1


```python
import math as m
import matplotlib.pyplot as plt
import numpy as np
import numpy.random as npr

def redimensionnement(b,d,c,T,k,n0 ):
    M=[n0]
    Temps=[0]
    delta_t=npr.exponential(1/((b+d+c*M[-1])*k*M[-1]))
    while Temps[-1]+delta_t<=T:
        plus_taux=M[-1]*b*k
        minus_taux=M[-1]*(d*k+c*k*M[-1])
        plus=plus_taux/(plus_taux+minus_taux)
        Temps.append(Temps[-1]+delta_t)
        jump=npr.rand()
        if jump < plus:
            M.append(M[-1]+1/k)
        else:
            M.append(M[-1]-1/k)
        if M[-1]==0:
            break
        delta_t=npr.exponential(1/((b+d+c*M[-1])*k*M[-1]))
    f=lambda n,t:(b-d-c*n)*n
    plt.step(Temps,M)

redimensionnement(0.1,0.03,0.005,250,1,1)
redimensionnement(0.1,0.03,0.005,250,10,1)
redimensionnement(0.1,0.03,0.005,250,100,1)
redimensionnement(0.1,0.03,0.005,250,1000,1)
plt.show()

```

Annexe 2


```
import scipy.integrate as scint
import numpy as np
import matplotlib.pyplot as plt
m=lambda M,D,T,A : M*(1.9-0.3*M  - 4*A - 28*T )*M + D
d=lambda M,D,T,A : (1.9 - 0.3*D)*D + 0.1*M + 4*A*M
t=lambda M,D,T,A : (8*M - 3 )*T
a=lambda M,D,T,A :-15*A +28*M*T
def ACT(h,tf,M0,D0,T0,A0):
    temps=np.arange(0,tf,h)
    M=[M0]
    D=[D0]
    A=[A0]
    T=[T0]
    n=len(temps)
    for i in range(1,n):
        M.append(M[-1]+h*m(M[-1],D[-1],T[-1],A[-1]))
        D.append(D[-1]+h*d(M[-1],D[-1],T[-1],A[-1]))
        T.append(T[-1]+h*t(M[-1],D[-1],T[-1],A[-1]))
        A.append(A[-1]+h*a(M[-1],D[-1],T[-1],A[-1]))
    X=np.ndarray.tolist(temps)
    plt.plot(X,M)
    plt.plot(X,D)
    plt.plot(X,T)
    plt.plot(X,A)
    plt.show()

```

Annexe 3


```python
import numpy as np
import random as r
import matplotlib.pyplot as plt

def choix_aleatoir(M,TUM,p):
    while True:
        for x in range(len(M)):
            for y in range(len(M)):
                if r.random()>=p and  (x,y) in TUM:
                    return(x,y)
##vérifié
def liste_prolif(M):#liste des cellules proliférantes
    L=[]
    for x in range(len(M)):
        for y in range(len(M)):
            if list(M[x,y])==[255,0,0]:
                L.append((x,y))
    return(L)
def liste_T(M):#liste des cellules T
    L=[]
    for x in range(len(M)):
        for y in range(len(M)):
            if list(M[x,y])==[0,255,0]:
                L.append((x,y))
    return(L)
def liste_Dead(M):#liste des cellules mortes
    L=[]
    for x in range(len(M)):
        for y in range(len(M)):
            if list(M[x,y])==[250,250,50]:
                L.append((x,y))
    return(L)

```

```
34 def cellular_automata(rprolif,rbinding,rescape,rlysis,rdecay,K,n,p):
35     """p: probabilité de choix des cellules proliférantes
36        n:nombre des instants avant la simulation"""
37     M=np.zeros((101,101,3),np.uint8)
38     M[51,51]=[255,0,0]
39     M[51,50]=[255,0,0]
40     M[51,52]=[255,0,0]
41     M[50,51]=[255,0,0]
42     M[52,51]=[255,0,0]
43     f=lambda x:rprolif*(1-x/K)
44     for i in range(n):
45         Prolif=liste_prolif(M)
46         T=liste_T(M)
47         D=liste_Dead(M)
48         TUM=Prolif+T+D
49         T1=TUM[:]
50         e=len(Prolif)
51         while T1!=[] :
52             (x,y)=choix_aleatoir(M,T1,p)
53             T1.remove((x,y))
54             r1=r.random()
55             if (x,y) in Prolif:
56                 if r1>f(e) and r1<=1-rbinding :
57                     M[x,y]=[0,255,0]
58                 else:
59                     V=[]#voisinage de la cellule, voisinage de MOORE
60                     if (x+1,y) not in TUM  :
61                         V.append((x+1,y))
62                     if (x-1,y)not in TUM  :
63                         V.append((x-1,y))
64                     if (x,y+1) not in TUM  :
65                         V.append((x,y+1))
66                     if (x,y-1) not in TUM  :
67                         V.append((x,y-1))
68                     if (x+1,y+1) not in TUM  :
69                         V.append((x+1,y+1))
70                     if (x-1,y-1) not in TUM  :
71                         V.append((x-1,y-1))
72                     if (x-1,y+1) not in TUM  :
73                         V.append((x-1,y+1))
74                     if (x+1,y-1) not in TUM  :
75                         V.append((x+1,y-1))
76                     if V!=[]:#rule for invasion
77                          (a,b)=r.choice(V)
78                         TUM.append((a,b))
79                         M[a,b]=[255,0,0]
```

```
            if (x,y) in T :
                if r1<=rescape:
                    M[x,y]=[255,0,0]
                elif r1>=1-rlysis:
                    M[x,y]=[250,250,50]
            if (x,y) in D :
                if r1<=rdecay :
                    M[x,y]=[0,0,0]
                    TUM.remove((x,y))
            if i ==15:
                plt.subplot(2,2,1)
                plt.imshow(M)
            if i ==30:
                plt.subplot(2,2,2)
                plt.imshow(M)
            if i ==50:
                plt.subplot(2,2,3)
                plt.imshow(M)
            if i ==79:
                plt.subplot(2,2,4)
                plt.imshow(M)
    plt.show()
cellular_automata(0.85,0.1,0.5,0.35,0.35,550,80,0.1)

```

Annexe 4

Soient $M_k(t) = \frac{M(t)}{k}$, $D_k(t) = \frac{D(t)}{k}$, $T_k(t) = \frac{T(t)}{k}$ pour $k \in \mathbb{N}^*$. On suppose que:

$$lim_{k \mapsto \infty}(M_k(0), D_K(0), T_k(0)) \overset{d}{=} (n_{M_0}, n_{D_0}, n_{T_0})$$

où $(n_{M_0}, n_{D_0}, n_{T_0})$ constitue une limite en loi supposée déterministe. On suppose aussi que $b_M, d_M, c_{MM}, c_{MD}, b_D, d_D, c_{DM}, c_{DD}, s_{MD}, s_{DM}, b_T, d_T, s_A, d_M^{[T]}$ sont des réels non nuls. Alors, sur tout segment $[0, t_F]$, la suite des processus $(M_k(t), D_k(t), T_k(t))_{k \in \mathbb{N}^*}$ converge en loi :

$$lim_{k \mapsto \infty}(M_k(t), D_k(t), T_k(t))_{t \in [0, t_F]} \overset{d}{=} (n_M(t), n_D(t), n_T(t))_{t \in [0, t_F]}$$

tel que $(n_M(t), n_D(t), n_T(t))$ soit la solution du système déterministe :

$$\begin{cases} \dot{n_M}(t) = (b_M - d_M - c_{MM} n_M(t) - c_{MD} n_D(t) - s_{MD} - (s_A + d_M^{[T]}) n_T) n_M(t) + s_{DM} n_D(t) \\ \dot{n_D}(t) = (b_D - d_D - c_{DD} n_D(t) - c_{DM} n_M(t) - s_{DM}) n_D(t) + s_{MD} n_M(t) + s_A n_M(t) \\ \dot{n_T}(t) = (b_T n_M(t) - d_T) n_T(t) \end{cases}$$

avec la condition initiale $(n_{M_0}, n_{D_0}, n_{T_0})$.