Yi Lin

Senior Design

Final report

A Tumor Growth Model

1 INTRODUCTION

There are several ways to modeling the tumor growth, a traditional way to modeling the tumor is using the continuous model, the continuous model is just based on a system of differential equation, and the continuous model is suitable for simulate population of simulation of tumor growth. Another way to modeling the tumor is using the discrete model, which is based on cellular automata. It first divide a 2D or 3D space into grid, and each cancer can be represented in a cell of grid, and each cells has a certain state, and different state would affect different result in its neighbor cell. This method can well describe cancer cell's spatial spreading. Another interesting way which is a sort of expanding of discrete modeling, it uses the agent-based modeling, and it can very well simulate the cell interaction based on the cell's energy system.

As we can see, these models have some limitations in tumor simulation, so in this project, I want to use a new more general way to approach the tumor growth. Since tumor growth is uncertain, all tumor models are based on probability and stochastic, and the models that I introduce is also stochastic, it randomly simulates the cells born and cells death process, and it is using an idea that borrow from molecular dynamics to further simulate the cancer cells dynamics. And since the model is a general way to approach modeling of cancer, it has a lot extendibility.

2 CELL POPULATION AND CELL LIFE CYCLE

In the model, the cell life cycle is defined by using Monte-Carlo methods as a random discrete value:

$$\zeta = \begin{cases} \text{Death}, P_{death} \\ \text{Interphase}, P_{rest} \\ \text{Metaphase}, P_{div} \end{cases}$$

Where in this case we need to guarantee that $P_{death} + P_{rest} + P_{div} = 1$. And we can also represent our life cycle as the following way:

Death	Interphase	Metaphase
$\zeta < P_{death}$	$P_{death} < \zeta \leq 1 - P_{div}$	$\zeta > 1 - P_{div}$

Above is how the life cycle process defined in our model, usually, at most of the time, cells is in the interphase (resting, neither dividing nor death). in practice, it first generates a random value ζ to be a uniform random value from [0,1), and compare to the user's defined probability P_{death} and P_{div}, to stochastically modeling the cells life cycle.

The cells life cycle can be implemented by the code below:

```
import random

lifetime = 30
CLT = [lifetime, lifetime, lifetime] #initial 3 cells
p_death = 0.1
p_div = 0.2

for i in range(0, len(CLT)):

    # if current cell is alive
    if CLT[i]>0:
        tmp = random.random() # roll a die

        # cell death
        if tmp<p_death:
            CLT[i] = 0
```

```
        # cell division
        elif tmp>=1-p_div:
            CLT.append(lifetime)

print(CLT)
```

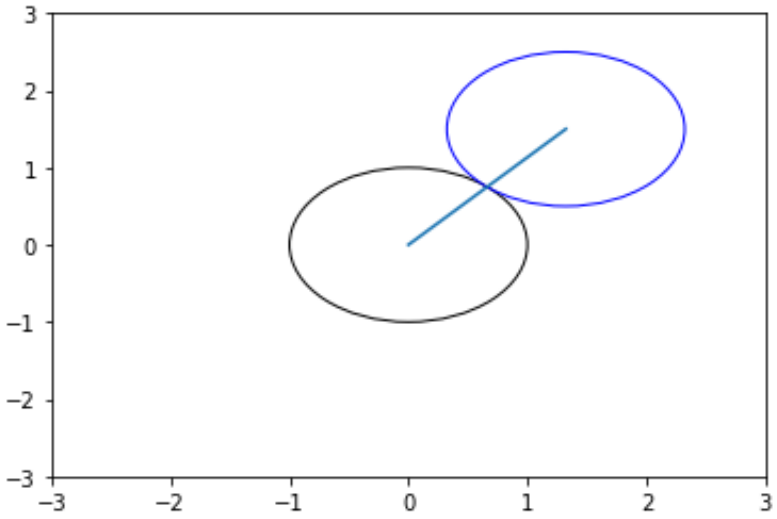
Where CLT is the list of our cells life cycle, and tmp is our ζ.

3 CELL DYNAMICS

As we mentioned, the cell dynamics of this model borrow some ideas from molecular dynamics, it first consider the cell spatial distribution as a gas cloud, and each cell's movement is determined by 2 type of force: Adhension for and Haptotaxis force. We first consider the Cell dynamics of in a 2D case.

3.1 NEWBORN CELL IMPLEMENTATION

We consider the new divided process as new divided cell (child) appears in the surrounding of its parent:



As we can see from above, the blue circle denotes for a newborn cell from the black cell. And this process could be easily implemented by the polar coordinate system:

$$\begin{cases} \psi = \text{random}(0,1),\ \psi \text{ is random angle} \\ x_{initial} = 2r_{cell} \cdot \cos(2\pi \cdot \psi) \\ y_{initial} = 2r_{cell} \cdot \sin(2\pi \cdot \psi) \end{cases}$$

Then we can just simply add the two lines of code into the cell division part:

```
x.append(2*r_cell*math.cos(2*math.pi*psi)) # initial x axis
y.append(2*r_cell*math.sin(2*math.pi*psi)) # initial y axis
```

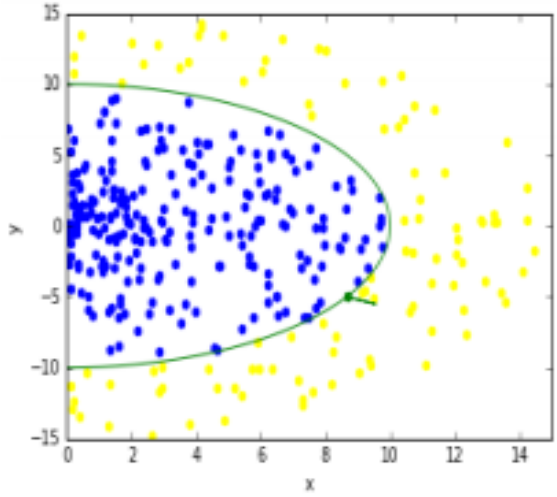
where x, y here stores for the list of cells' x coordinate and y coordinate.

3.2 HAPTOTAXIS FORCE

The haptotaxis force is a repulsive force between cells under haptotaxis effects, cell motility is caused by the pressure difference between inner and outer cell of the tumor. Given by a modified Fick's first law:

$$\vec{F}_{Haptotaxis} = -\alpha(c_{outer} - c_{inner})\frac{\vec{r}}{r}$$

Where α here is a constant, c_{outer} is number of outer cells, and c_{inner} is number of inner cells. $\frac{\vec{r}}{r}$ is normalized radius vector of current cells. Here is an example:



From the figure above, the green cells haptotaxis force is given by the number of yellow cells as c_{outer} and the number of blue cells as c_{inner}, and the directions is given by $\left(\frac{x}{\sqrt{x^2+y^2}}, \frac{y}{\sqrt{x^2+y^2}}\right)$. Here is how the haptotaxis force implemented in code:

```
# haptotaxis force

import math

x = [0,1,2,3]
y = [0,1,2,3]
c_out = 0
c_in = 0
CLT = [30,30,30,30]
alpha = 10

# count number of innner and outer cells
for j in range(0,len(CLT)):
    for k in range(0,len(CLT)):
        if CLT[k]>0:
            if x[k]**2+y[k]**2 > x[j]**2+y[j]**2:
                c_out += 1
            elif x[k]**2+y[k]**2 < x[j]**2+y[j]**2:
                c_in += 1

    Fx = -alpha*(c_out-c_in)*x[j]/(math.sqrt(x[j]**2+y[j]**2)+0.01)
    Fy = -alpha*(c_out-c_in)*y[j]/(math.sqrt(x[j]**2+y[j]**2)+0.01)
    print("Haptotaxis force for cell ",j)
    print(Fx)
    print(Fy)
```

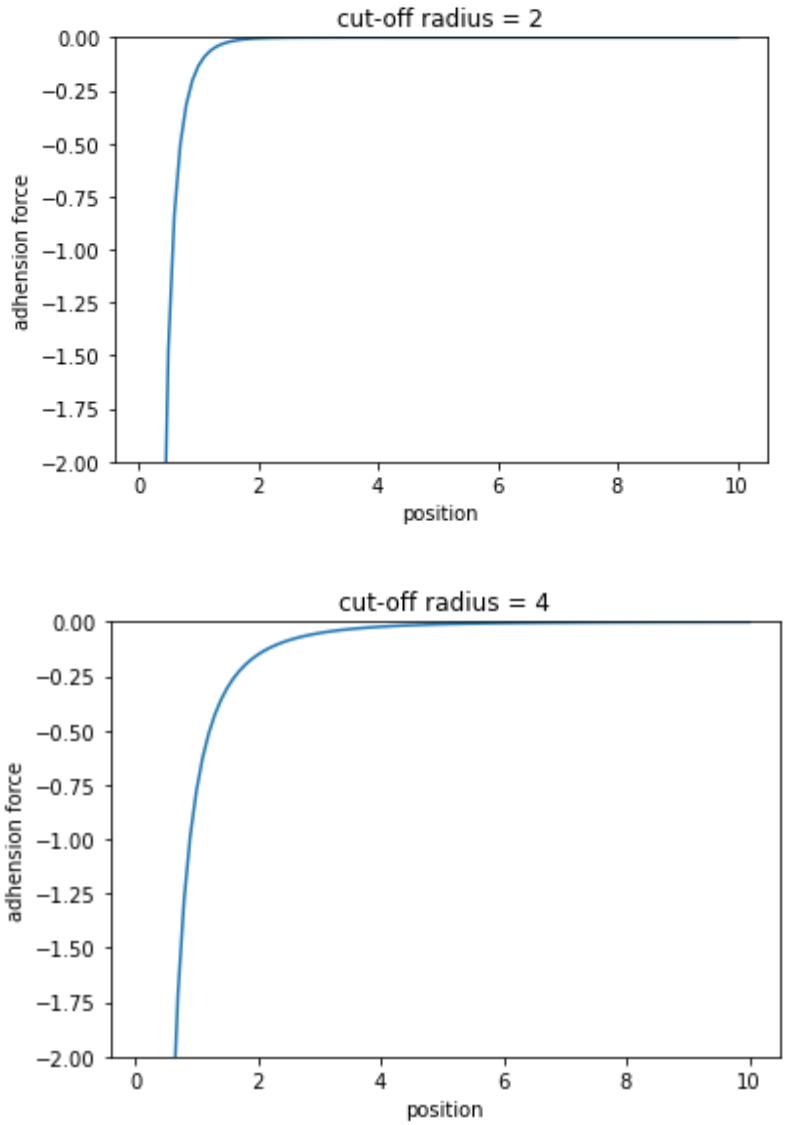
3.3 ADHENSION FORCE

The adhension force in the cell system is caused by the cell-cell mechanical interaction throw special components of their membranes or skeletons. Notice that the action radius of the adhension force should be short enough. So the formula for the adhension force is given by:

$$\vec{F}_{Adhension} = -\beta \frac{\vec{r}}{r^3} e^{-\gamma r} = -\beta \frac{\vec{r}}{r^3} e^{-\frac{r}{R_{cut-off}}}$$

Where β is a constant, $\vec{r}$ is the vector between two cell, and $R_{cut-off}$ is the cut-off radius. The cut-off radius is that any cell that the distance between that cell to current cell is greater than cut-off radius would not have the adhension force between them.

Below two figures show that how the cut-off radius affect the adhension force:



As we can see, the larger cut-off radius would gives us a larger convergence rate for the adhension force to 0. That means the larger cut-off radius would let the action radius of the adhension force to be larger.

Then we can implement the adhension force by given formula using the code below:

```
# adhension force for j-th cell

x = [0,1,2,3]
y = [0,1,2,3]
CLT = [30,30,30,30]
beta = 10
r_cell = 1
```

```
for j in range(0,len(CLT)):
    for i in range(0,len(CLT)):
        if i != j and CLT[i]>0:
            dx = x[j]-x[i]
            dy = y[j]-y[i]
            dr = math.sqrt(dx**2+dy**2)
            Fx += -beta*dx/(dr+1)**3*math.exp(-dr/(4*r_cell))
            Fy += -beta*dy/(dr+1)**3*math.exp(-dr/(4*r_cell))

    print("Adhension force for cell ",i)
    print(Fx)
    print(Fy)
```

3.4 CELL MOTION

We already discuss the adhension force and the haptotaxis force for the cancer cell, then we can simply combine them to get the cells motion for every single j^{th} cell:

$$\vec{F}^{j}_{Total} = \vec{F}^{j}_{Haptotaxis} + \vec{F}^{j}_{Adhension}$$

$$= -\alpha\left(c^{j}_{outer} - c^{j}_{inner}\right)\frac{\vec{r}^{j}}{r^{j}} - \beta\sum_{i\neq j}\frac{\vec{r}^{j}_{i}}{\left(r^{j}_{i} + R_{Banned}\right)^{3}}e^{-\frac{r^{j}_{i}}{R_{cut-off}}}$$

The R_{Banned} here is a artificial radius in avoidance of the $\vec{F}^{j}_{Adhension}$ to be too large, since a extreme large net force for a single cell would that the acceleration to be too large. And it also made for represent cell hardness. Then we can use the newton's second law:

$$\vec{a}^{j} = \frac{\vec{F}^{j}_{Total}}{mass}$$

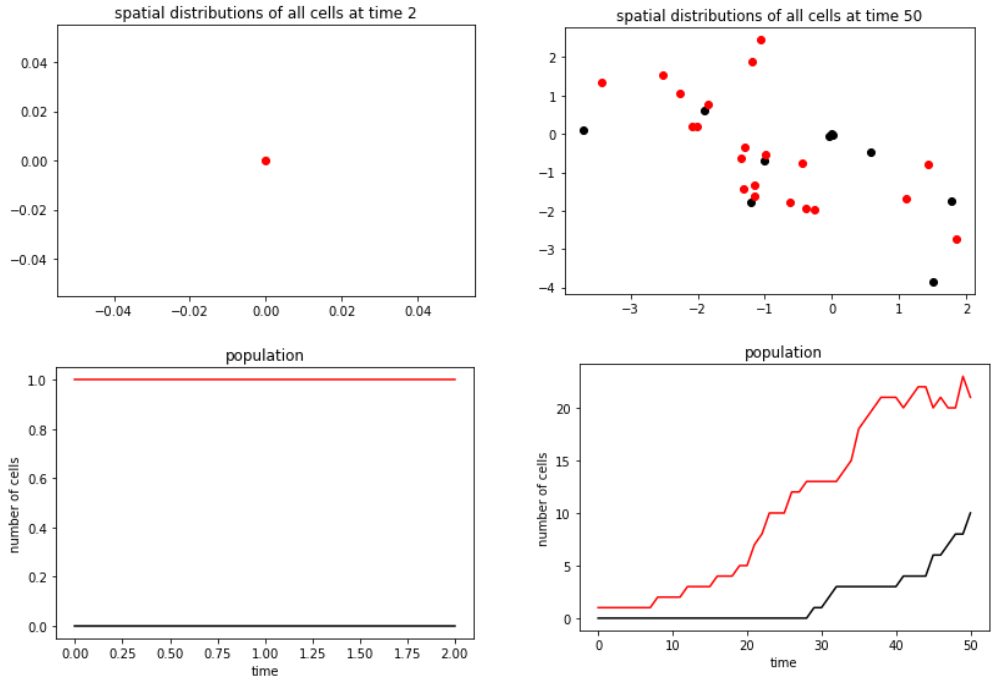
To get the acceleration for the j^{th} cell. Here is the formula for the cell motion system expand in x-coordinate:

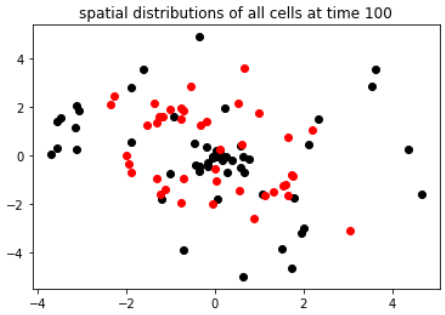
$$\begin{cases} a_x^j = \dfrac{dv_x^j}{dt} = \dfrac{\alpha\left(c_{outer}^j - c_{inner}^j\right)\dfrac{x^j}{r^j} - \beta\sum_{i\neq j}\dfrac{x^j - x^i}{\left(r_i^j + R_{Banned}\right)^3}e^{-\frac{r_i^j}{R_{cut-off}}}}{mass} \\ \dfrac{dx^j}{dt} = v_x^j \end{cases}$$

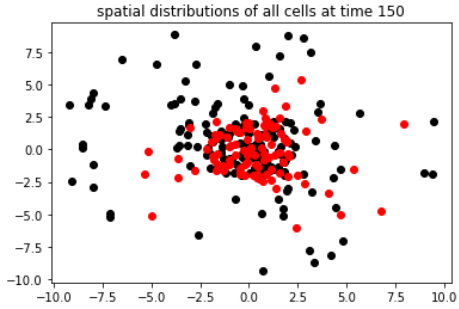
Then we can further solve the system using Euler's methods, for a 2D cases, it only requires 4 formula to represent the dynamic cell motion.

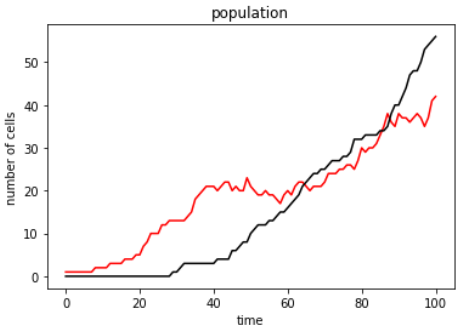
4 SIMULATION RESULT

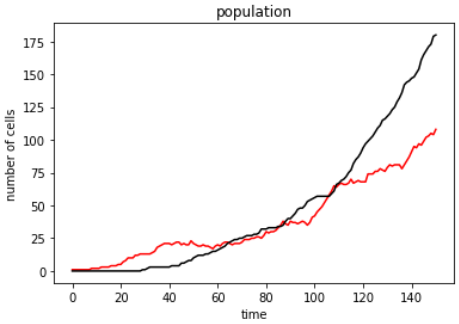
Then we can combine each part above to get a basic tumor model. And as the tumor growth, we keep tracking the population at each time and we can get the graph of population – time dependencies, here are the result for the simulation for $P_{death} = 0.01$, $P_{div} = 0.05$, lifetime $= 30$, mass $= 1$, $r_{cell} = 1$, and 1 initial cells:

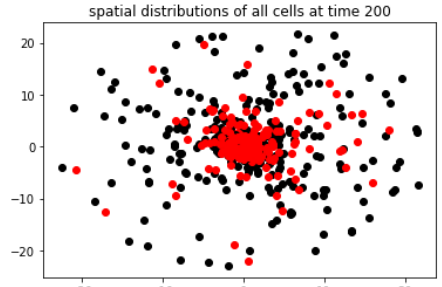


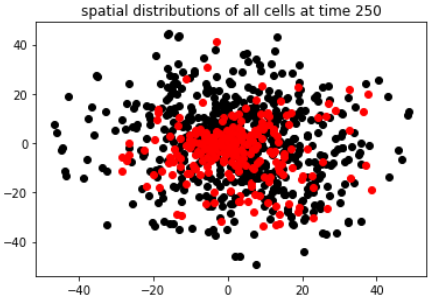

spatial distributions of all cells at time 100

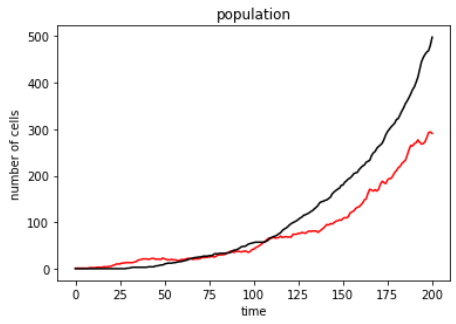

spatial distributions of all cells at time 150

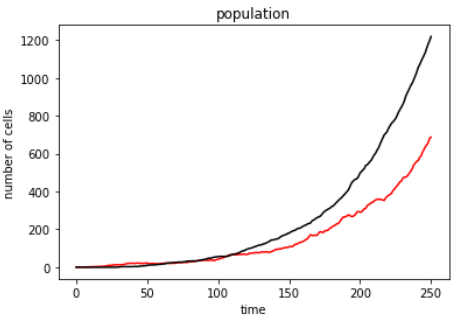

population
number of cells
time

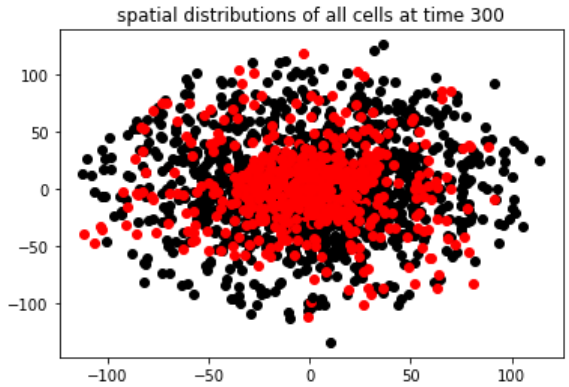

population
number of cells
time

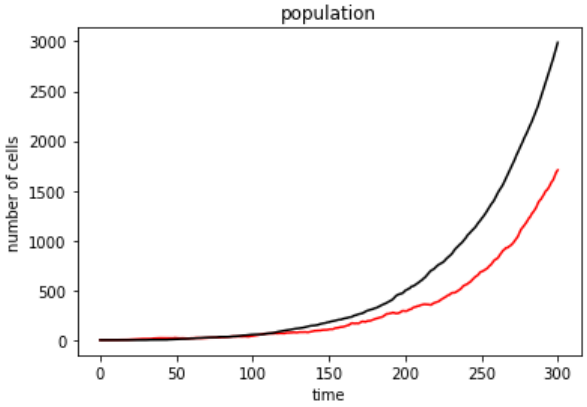

spatial distributions of all cells at time 200


spatial distributions of all cells at time 250


population
number of cells
time

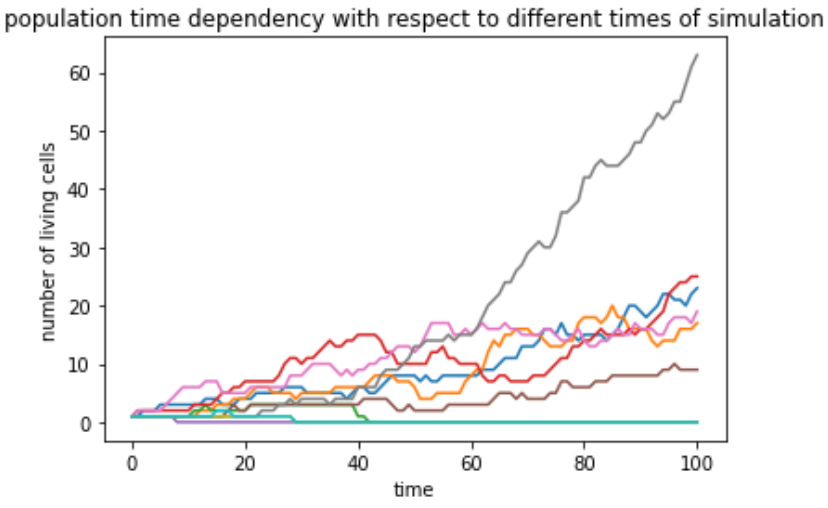

population
number of cells
time





As we can see, at time 300, the system generates a circle tumor with radius about 200.

Another character of this hybrid model is uncertainty. Uncertainty is manifested in different results of simulations with the same vector of initial parameters. Here we do the simulation 10 times with initial 1 cell and show the result of population growth during time 0 to 100:



We can see that most of the simulation result at time 100 will end up 100 living cells, but there is 1 tumor growth explosive with 60 cells at time 100 and on the other hand, there are several “unlucky tumor” dies out very early. But these differences in populations become smaller with rising of simulation time. Also, we will obtain the same in average results if we make many simulations.

5 MODIFICATION

5.1 LIFETIME MODIFICATION

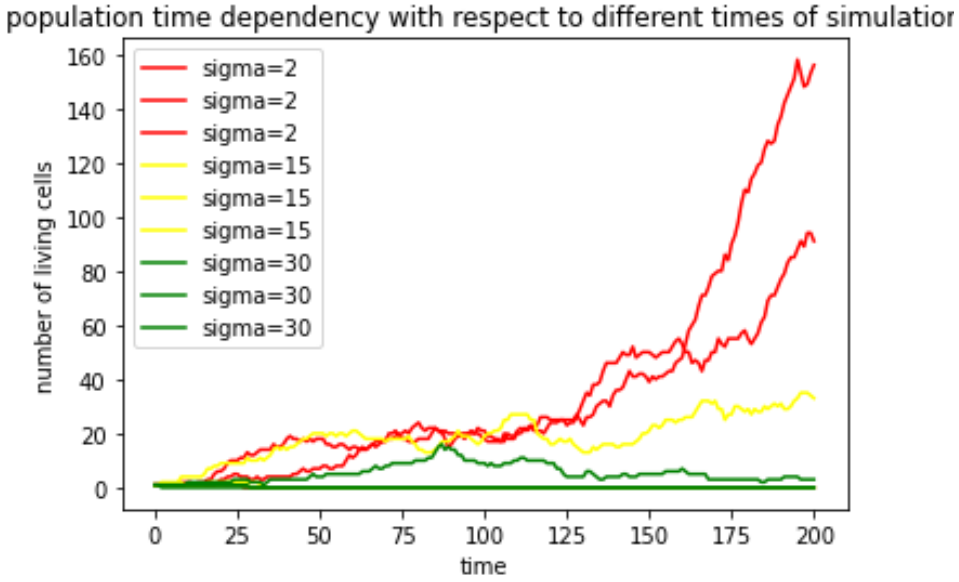
In previous basic model, we predefine the lifetime of each newborn cell to be a fixed value. But in real life, each cell has own lifetime which can be different from other’s one. We will need to take it in account if we want to do our model more natural. So we can create one more random value in our model. This value is continuous. We can obtain it from numerous age distribution models, which depend on definite cell type. Here is a simple way to implement it is using normal distributed random value. Here is the gaussian distribution formula for cell’s lifetime:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

Where in this case, the average lifetime of our cancer cells would be μ, which is predefined by user, and σ is the standard deviation, in this case, should also be user predefined. The larger σ will give us a larger instability of each cell’s lifetime. Here in our code, we can just simply append the normal distributed random lifetime for each newborn cell:

```
CLT.append(random.gauss(lifetime, sigma_list[line_num]))
```

And then we simulate several times to see how the standard deviation affect population since statistically, the larger σ gives the smaller growing rate of tumor.

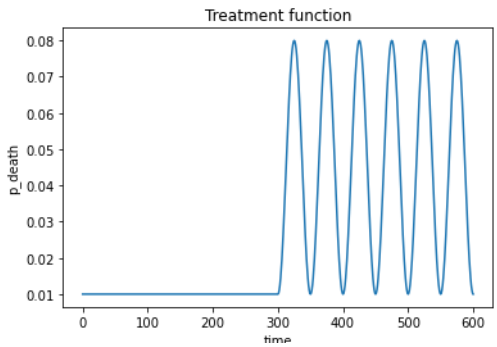


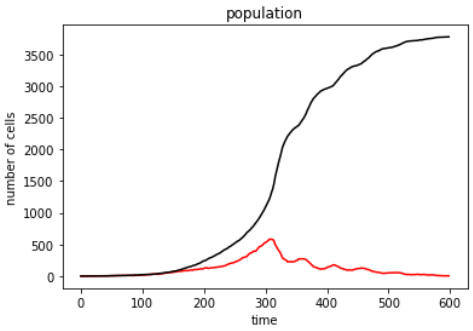
5.2 TREATMENT MODELING

In this model, we can also simulate the therapy of both chemical or/and radiation throw changing in lifecycle. The cancer treatment is a process that killing tumor cells. Then we can consider treatment as rising P_death. Here we select a reasonable treatment function:

$$P_{death} = \begin{cases} P^0_{death}, \text{ if } t \le T_{treatment} \\ P^0_{death} + P_s\left(\sin^2\left(\frac{2\pi}{T_s}(t - T_{treatment})\right)\right), \text{ if } t > T_{treatment} \end{cases}$$

Here we say the function is “reasonable” since in real life, most of the cancer treatment, either radiation or chemical, would be periodic. And as we know, the $\sin^2$ function is periodic. In the formula $T_{treatment}$ denotes for the time that start the treatment and P^0_{death} denotes for the initial cells death rate, and P_s is the increased death rate, and T_s is the treatment period.

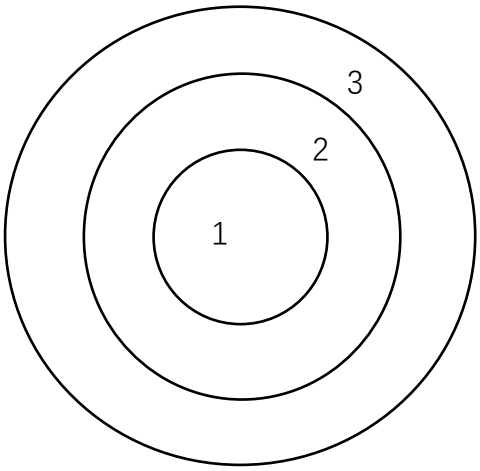




From the result, we can see that the after time 300, the treatment started and the number of living cancer cells start decreasing periodically.

5.3 MULTILEVEL MODIFICATION

Some tumor is developed by full or semi absents of blood vessels. And cancer cells will necrotize without blood vessels. And it will cause the tumor to be act as multilevel structures. The basic idea of multilevel structures is assigning different probability values for different levels:



The most inner level is necrotic core, it contains a lot of death cells. The intermediate level consists of quiescent cells. And the most outer level is a is a layer of proliferating cells, because in this level, there is penetration of blood vessels.

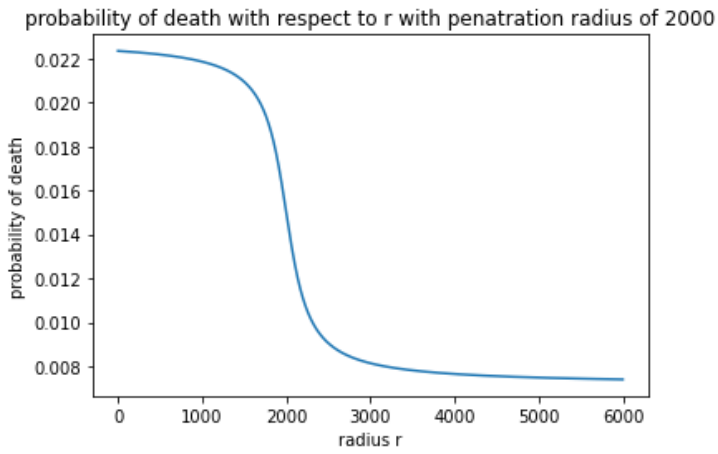
In this case, the growth of tumor obeys the logistic equation:

$$\frac{dP}{dt} = r \cdot P\left(1 - \frac{P}{K}\right)$$

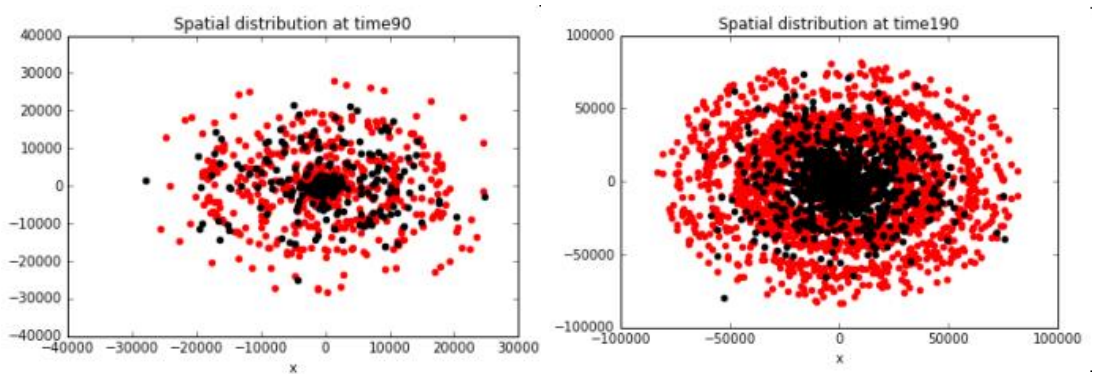
Then we can use the death probability given by the equation below with respect to the cell radius r:

$$P_{death}(r) = P^0_{death} + P_{level}\left(0.5 + 0.5 \cdot \arctan\left(\chi \cdot \left(r - r_{penetration}\right)\right)\right)$$

The P^0_{death} denotes for the initial death rate, and P_{level} denotes for increased death probability, and χ is negative constant that shows how fast the P_{death} change in the intermediate level and $r_{penetration}$ is the penetration radius for blood vessel. Notice that the penetration radius in implementation should have an upper bound. Here we create a probability death function with penetration radius of 2000:



And here is the simulation result for the multilevel structures:

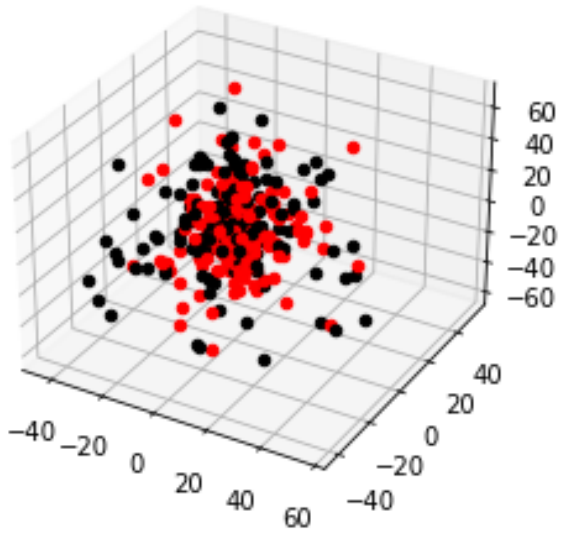


6 3D TUMOR

The last part of the model is just extends our model into 3D. We can simply add a z-coordinate and resolve the x,y,z value for the polar coordinate system:

$$\begin{cases} \psi = \text{random}(0,1),\ \psi \text{ is random angle} \\ \phi = \text{random}(0,1),\ \phi \text{ is a random angle} \\ x_{initial} = 2r_{cell} \cdot \cos(2\pi \cdot \psi) \cdot \sin(2\pi \cdot \phi) \\ y_{initial} = 2r_{cell} \cdot \sin(2\pi \cdot \psi) \cdot \sin(2\pi \cdot \phi) \\ z_{initial} = 2r_{cell} \cdot \cos(2\pi \cdot \phi) \end{cases}$$

Where in this case, we add another random value ϕ as the same purpose as ψ.



7 CONCLUSIONS

The model for tumor growth modeling using random values techniques and Newtonian dynamics in 2 and 3 space dimensions. It contains a part that modeling of cells death and cells division, and another part of cells dynamics. And the models has a lot of extensibility, it can be modified to represent many types of specific tumors, and it also support the treatment modeling. It would be consider as a practical model in tumor modeling.

REFERENCES

[1] Altman, D. G., & Bland, J. M. (1995). Statistics notes: the normal distribution. Bmj, 310(6975), 298.

[2] Dréau, D., Stanimirov, D., Carmichael, T., & Hadzikadic, M. (2009, April). An agent-based model of solid tumor progression. In International Conference on Bioinformatics and Computational Biology (pp. 187-198). Springer, Berlin, Heidelberg.

[3] Strang, G., & Herman, E. “Jed.” (2021, January 2). The Logistic Equation. Retrieved May 21, 2021, from https://math.libretexts.org/@go/page/2559

[4] Surkov, Y., Samofalov, I., & Anastasia, M. (2017). About new hybrid approach in cancer modeling. arXiv preprint arXiv:1705.00666.