Reaction-Diffusion Model for Propagation of Invasive Cancer

Adam Syammas Zaki Purnomo

Tohoku University, School of Engineering

Department of Robotics

1. Introduction

Cancer is a broad term that covers many types of disease. It involves abnormal cells growth with the ability to invade and spread from one part of the body to the others. There are approximately over 100 types of cancers that affect humans [1]. Cancer happens due to genetic mutation that can be caused by many factors such as chemical and radiation exposure, infections, and genetic defects. Normally, our body will dispose of any cells that have damaged DNA. However, our ability to eliminate cells with defective DNA goes down as we age. This is one explanation why cancer is such a mainstream disease in developed countries where life expectancy is usually higher.

While technological advancement has enabled us to gain a deeper understanding of cancer, there are still many things that we do not know yet about cancer, especially its behavior in different tissue environments in the body. Nowadays, many treatment options exist, including surgery, chemotherapy, radiation therapy, and immunotherapy. However, these treatments are not bulletproof, which means their cure is not permanent. This is even true for the case where cancer has spread from its primary tumor to other parts of the body or called metastatic cancer. These treatments are usually aimed to control cancer since getting rid of cancer entirely is unattainable. In fact, the majority of cancer deaths are due to metastases of the primary tumor [2].

In this report, a mathematical model of cancer growth is presented. Mathematical modeling of cancer is a useful and inexpensive approach to determine and predict the stage, size, and progression of cancer. Many models have been presented and have been used to get an insight into how cancer grows and invades other tissues. In this report, we will focus on the reaction-diffusion model for cancer growth. In section 2, the reaction-diffusion model, which also considers chemotherapy treatment, will be discussed. Section 3 describes the derivation of the finite difference method for this simulation is presented. In section 4, boundary conditions and initial conditions for computer simulation are explained. In section 5, the result of the simulation will be presented and discussed. Finally, section 6 presents the conclusion of this report. The simulation code, written in python, is given in Appendix A, and the computer specification used during simulation is given in Appendix B.

2. Mathematical Modelling

This report presents a reaction-diffusion model of cancer propagation based on [3] with some modifications. In this model, the mechanism of cancer invasion is facilitated by the diffusion of excess acid production from cancer cells to the surrounding healthy tissue. This is supported by [4], showing that normal cells have an optimal pH of 7.4 while cancer cells thrive at 6.8. Healthy cells are unable to survive in this acidic environment and eventually die. The loss of healthy cells will even facilitate the progress of cancer invasion to the surrounding area. This model is a system of three coupled reaction-diffusion equations that determine the spatial and temporal evolution of three variables: the density of normal cells $H(x, y, t)$, cancerous cells $C(x, y, t)$, and the excess acid concentration $L(x, y, t)$. Since we are simulating it only on 2D space, the dimension of H and C are cells/cm2, and the dimension of L is molar (M). The equation that describes the healthy cells is given by

$$\frac{\partial H}{\partial t} = r_H H \left(1 - \frac{H}{k_H} - \alpha \frac{C}{k_C}\right) - \beta LH + \nabla(D_H \nabla H), \quad (1)$$

where r_H is the growth rate of Verhulst logistic growth model, k_H is the carrying capacity, α is Lotka-Volterra competition strength parameter, β is interaction coefficient with the excess acid concentration, D_H is diffusion coefficient. The equation that describes the cancerous cells is almost the same as (1) except for the lack of acid interaction term since the cancerous cells do not die in the acidic condition. The equation is given by

$$\frac{\partial C}{\partial t} = r_C C \left(1 - \frac{C}{k_C} - \alpha \frac{H}{k_H}\right) + \nabla(D_C \nabla C), \quad (2)$$

where r_C and k_C are the growth rate and carrying capacity of cancerous cells, α is Lotka-Volterra competition strength parameter, and D_C is diffusion coefficient of cancerous cells. In this report, we focus on a special case where the competition between both healthy can cancerous cells is infinitesimal; thus, we can set α = 0. Healthy tissue is usually well regulated to be part of a specific tissue or organ; it is reasonable to assume that it does not diffuse. Hence, we can set D_H = 0. On the other hand, we can assume that the ability of cancer cells to spread depends on the density of healthy cells. The diffusion of cancerous cells is given by

$$D_C = D_{C'} \left(1 - \frac{H}{k_H}\right) \quad (3)$$

where $D_{C'}$ is the diffusion coefficient of cancerous cells in the absence of healthy cells. From equation (3), we can see that if healthy cells operate at their carrying capacity, the diffusion coefficient of cancerous cells is zero, which means that the tumor is confined. The cancerous cells cannot diffuse without the surrounding healthy cells diminished first from their carrying capacity. In this model, cancerous cells develop only in the region where the healthy cells are greatly compromised.

The excess acid concentration is produced at a rate proportional to the density of cancer cells. Our body has the acid buffer ability to maintain homeostasis to reabsorb the excess acid concentration. The equation that describes the spatial and temporal evolution of excess acid concentration is given by

$$\frac{\partial L}{\partial t} = r_L C - \gamma L + D_L \nabla^2 L \quad (4),$$

where r_L is acid production rate, γ is reabsorption rate, and D_L is diffusion coefficient of the excess acid concentration. We can further modify the equation (1) and (2) if we also consider the effect of chemotherapy treatment. Chemotherapy treatment affects both healthy and cancer cells. However, it is more deadly to cancer cells since chemotherapy targets cells that rapidly multiply. We can assume that the death rate due to chemotherapy is proportional to the growth rate of the cells multiplied by their sensitivity coefficient toward the chemo drug and the drug variable $T(t)$. The modified equation (1) and (2) is shown as follows.

$$\frac{\partial H}{\partial t} = r_H H\left(1-\frac{H}{k_H}\right) - \beta LH - \ d_H TH\left(1-\frac{H}{k_H}\right) \quad (1)$$

$$\frac{\partial C}{\partial t} = r_C C\left(1-\frac{C}{k_C}\right) - \ d_C TC\left(1-\frac{C}{k_C}\right) \ + \nabla(D_C \nabla C) \quad (2)$$

3. Finite Difference Method

In this section, the derivation of finite difference method for this problem will be shown. In this report, explicit method is used. For simplicity of notation, we introduce several new terms as follows.

$$Reaction_H = r_H H\left(1-\frac{H}{k_H}\right) - \beta LH - \ r_H TH \quad (5)$$

$$Reaction_C = \ r_C C\left(1-\frac{C}{k_C}\right) - \ r_C TC \quad (6)$$

$$Reaction_L = \ r_L C - \gamma L \quad (7)$$

The indices i, j represents coordinate on 2D space while the index n represents time-step. The derivation using the explicit method of equation (1) is shown as follows.

$$\frac{H_{i,j}^{n+1} - H_{i,j}^{n}}{\Delta t} = {Reaction_H}_{i,j}^{n} \quad (8)$$

$$H_{i,j}^{n+1} = H_{i,j}^{n} + \ \Delta t {Reaction_H}_{i,j}^{n} \quad (8)$$

The derivation of the explicit method of equation (2) is given below.

$$\frac{C_{i,j}^{n+1} - C_{i,j}^{n}}{\Delta t} = {Reaction_C}_{i,j}^{n} + \frac{{D_C}_{i,j+1/2}^{n}\frac{dC_{i,j+1/2}^{n}}{dx} - {D_C}_{i,j-1/2}^{n}\frac{dC_{i,j-1/2}^{n}}{dx}}{\Delta x}$$
$$+ \frac{{D_C}_{i+1/2,j}^{n}\frac{dC_{i+1/2,j}^{n}}{dx} - {D_C}_{i-1/2,j}^{n}\frac{dC_{i-1/2}^{n}}{dx}}{\Delta y} \quad (9)$$

$${D_C}_{i,j+1/2}^{n} = \frac{{D_C}_{i,j}^{n} + {D_C}_{i,j+1}^{n}}{2} \quad (10)$$

$${D_C}_{i,j-1/2}^{n} = \frac{{D_C}_{i,j-1}^{n} + {D_C}_{i,j}^{n}}{2} \quad (11)$$

$$\frac{dC^{n}_{i,j+\frac{1}{2}}}{dx} = \frac{C_{i,j+1}^{n} - C_{i,j}^{n}}{\Delta x} \quad (12)$$

$$\frac{dC_{i,j-1/2}^{n}}{dx} = \frac{C_{i,j}^{n} - C_{i,j-1}^{n}}{\Delta x} \quad (13)$$

Substituting equation (10-12) to equation (9), we get

$$\frac{C_{i,j}^{n+1}-C_{i,j}^{n}}{\Delta t} = Reaction_{C_{i,j}}^{n} + \frac{\left(D_{C_{i,j}}^{n}+D_{C_{i,j+1}}^{n}\right)\left(C_{i,j+1}^{n}-C_{i,j}^{n}\right)-\left(D_{C_{i,j-1}}^{n}+D_{C_{i,j}}^{n}\right)\left(C_{i,j}^{n}-C_{i,j-1}^{n}\right)}{2\Delta x^{2}} + \frac{\left(D_{C_{i,j}}^{n}+D_{C_{i+1,j}}^{n}\right)\left(C_{i+1,j}^{n}-C_{i,j}^{n}\right)-\left(D_{C_{i-1,j}}^{n}+D_{C_{i,j}}^{n}\right)\left(C_{i,j}^{n}-C_{i-1,j}^{n}\right)}{2\Delta y^{2}} \quad (9)$$

$$C_{i,j}^{n+1} = C_{i,j}^{n} + \Delta t Reaction_{C_{i,j}}^{n} + \frac{\Delta t}{2\Delta x^{2}}\Big(\left(D_{C_{i,j}}^{n}+D_{C_{i,j+1}}^{n}\right)\left(C_{i,j+1}^{n}-C_{i,j}^{n}\right)-\left(D_{C_{i,j-1}}^{n}+D_{C_{i,j}}^{n}\right)\left(C_{i,j}^{n}-C_{i,j-1}^{n}\right) + \left(D_{C_{i,j}}^{n}+D_{C_{i+1,j}}^{n}\right)\left(C_{i+1,j}^{n}-C_{i,j}^{n}\right)-\left(D_{C_{i-1,j}}^{n}+D_{C_{i,j}}^{n}\right)\left(C_{i,j}^{n}-C_{i-1,j}^{n}\right)\Big) \quad (9)$$

From equation (3), $D_C = D_{C'}\left(1-\frac{H}{k_H}\right)$. For notation simplification, we introduce κ where

$$\kappa = 1-\frac{H}{k_H} \quad (14)$$

Therefore, equation (9) can be simplified as follows.

$$C_{i,j}^{n+1} = C_{i,j}^{n} + \Delta t Reaction_{C_{i,j}}^{n} + \frac{\Delta t D_{C'}}{2\Delta x^{2}}\Big(\left(\kappa_{i,j}^{n}+\kappa_{i,j+1}^{n}\right)\left(C_{i,j+1}^{n}-C_{i,j}^{n}\right)-\left(\kappa_{i,j-1}^{n}+\kappa_{i,j}^{n}\right)\left(C_{i,j}^{n}-C_{i,j-1}^{n}\right) + \left(\kappa_{i,j}^{n}+\kappa_{i+1,j}^{n}\right)\left(C_{i+1,j}^{n}-C_{i,j}^{n}\right)-\left(\kappa_{i-1,j}^{n}+\kappa_{i,j}^{n}\right)\left(C_{i,j}^{n}-C_{i-1,j}^{n}\right)\Big) \quad (9)$$

The derivation of the explicit method of equation (4) is given below.

$$\frac{L_{i,j}^{n+1}-L_{i,j}^{n}}{\Delta t} = Reaction_{L_{i,j}}^{n} + D_L\left(\frac{\frac{dL_{i,j+1/2}^{n}}{dx}-\frac{dL_{i,j-1/2}^{n}}{dx}}{\Delta x}+\frac{\frac{dL_{i+1/2,j}^{n}}{dy}-\frac{dL_{i-1/2,j}^{n}}{dy}}{\Delta y}\right) \quad (14)$$

$$\frac{L_{i,j}^{n+1}-L_{i,j}^{n}}{\Delta t} = Reaction_{L_{i,j}}^{n} + D_L\left(\frac{L_{i,j+1}^{n}-2L_{i,j}^{n}+L_{i,j-1}^{n}}{\Delta x^{2}}+\frac{L_{i+1,j}^{n}-2L_{i,j}^{n}+L_{i-1,j}^{n}}{\Delta y^{2}}\right) \quad (14)$$

$$L_{i,j}^{n+1} = L_{i,j}^{n} + \Delta t Reaction_{L_{i,j}}^{n} + \frac{\Delta t D_L}{\Delta x^{2}}\left(L_{i,j+1}^{n}+L_{i,j-1}^{n}+L_{i+1,j}^{n}+L_{i-1,j}^{n}-4L_{i,j}^{n}\right) \quad (14)$$

The downside of the explicit method is that it can be numerically unstable if the time step is not chosen carefully. For 1D heat conduction with no reaction term, the stability condition is when

$$\lambda = \frac{\kappa\Delta t}{\Delta x^{2}} \leq 0.5 \Rightarrow \Delta t \leq \frac{\Delta x^{2}}{2\kappa} \quad (15)$$

For 2D, the stability condition is

$$\lambda = \frac{\kappa \Delta t}{\Delta x^2} \leq 0.25 \Rightarrow \Delta t \leq \frac{\Delta x^2}{4\kappa} \quad (16).$$

For our model on 2D space, we have reaction terms and non-constant diffusion constant. Just to make sure that the solution is numerically stable, Δt is chosen to be the smallest value between the following inequalities.

$$\Delta t \leq \frac{\Delta x^2}{4D_L} \quad (17)$$

$$\Delta t \leq \frac{\Delta x^2}{2D_{C'}} \quad (18)$$

4. Parameters, Boundary Condition, and Initial Condition

The parameters of the simulation are given in the table below.

Parameter	**Estimated Value**	**Reference**
r_H	$1 \times 10^{-6} [\mathrm{s}^{-1}]$	[5]
r_C	$1 \times 10^{-6} [\mathrm{s}^{-1}]$	[5]
r_L	$2.2 \times 10^{-17} [\mathrm{M} \cdot \mathrm{cm}^2 \cdot \mathrm{s}^{-1}]$	[3]
K_H	$5 \times 10^{7} [\mathrm{cm}^{-2}]$	[5]
K_C	$5 \times 10^{7} [\mathrm{cm}^{-2}]$	[5]
β	$0 \sim 10 [\mathrm{M}^{-1}\mathrm{s}^{-1}]$	[3]
γ	$1.1 \times 10^{-4} [\mathrm{s}^{-1}]$	[3]
$D_{C'}$	$2 \times 10^{-10} [\mathrm{cm}^2\mathrm{s}^{-1}]$	[6]
D_L	$5 \times 10^{-6} [\mathrm{cm}^2\mathrm{s}^{-1}]$	[7]

In this case, we are considering the propagation of cancer on 5x5 cm^2 of healthy tissue. Cancer originally starts from the center of the tissue and begins to spread to the surrounding tissue. The initial condition of the cancer cells is given by

$$C(x, y, 0) = 3 \times 10^7 e^{-5(x^2+y^2)} \quad (19),$$

while the initial condition of healthy cells is given by

$$H(x, y, 0) = RandomUniform(4.91 \times 10^7, 4.99 \times 10^7) \quad (20).$$

For the cancer cells to be able to spread, the healthy cells should operate below their carrying capacity; otherwise, the cancer spread will be contained.

In this simulation, since we are only examining the behavior of how cancer spreads in a limited space, we assume that there are no cells that can go inside and outside the area that we examine. Therefore, the flux of all three variables is zero, leading us to use the Neumann boundary condition given by

$$\frac{\partial C}{\partial n} = 0 \quad (22),$$

$$\frac{\partial L}{\partial n} = 0 \quad (23),$$

where n is the normal vector in the boundary line, for example, at the boundary lines $x = 0$, the diffusion term is evaluated as follows

$$\frac{\partial^2 C}{\partial x^2} = \frac{D_{C_{i,1+1/2}} \frac{dC_{i,1+1/2}}{dx} - D_{C_{i,1+1/2}} \frac{dC_{i,1}}{dx}}{(\Delta x/2)} = \frac{D_{C_{i,1+1/2}} \frac{dC_{i,1+1/2}}{dx}}{(\Delta x/2)}$$
$$= \frac{\left(D_{C_{i,1}} + D_{C_{i,2}}\right)\left(C_{i,2} - C_{i,1}\right)}{\Delta x^2} \quad (24)$$

$$\frac{\partial^2 L}{\partial x^2} = \frac{\frac{dL_{i,1+1/2}}{dx} - \frac{dL_{i,1}}{dx}}{(\Delta x/2)} = \frac{\frac{dL_{i,1+1/2}}{dx}}{(\Delta x/2)} = \frac{2\left(L_{i,2} - L_{i,1}\right)}{\Delta x^2}. \quad (25)$$

At other boundary lines, the diffusion term is also evaluated in the same manner.

5. Simulation

In this section, simulation results are presented. A cancer propagation simulation on 5x5 cm^2 tissue in 2 months period is conducted. The simulation parameter is given as follows.

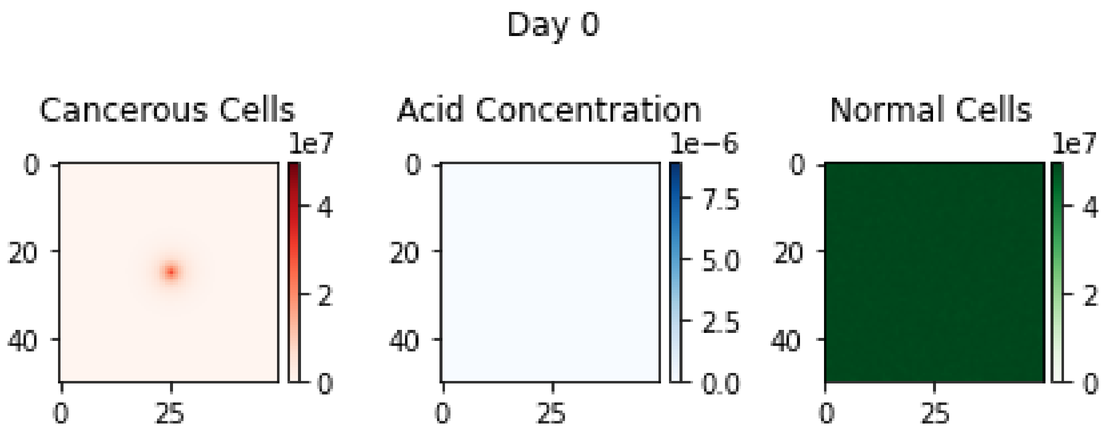
$$\Delta x = 0.1 \text{ cm} \quad (26)$$

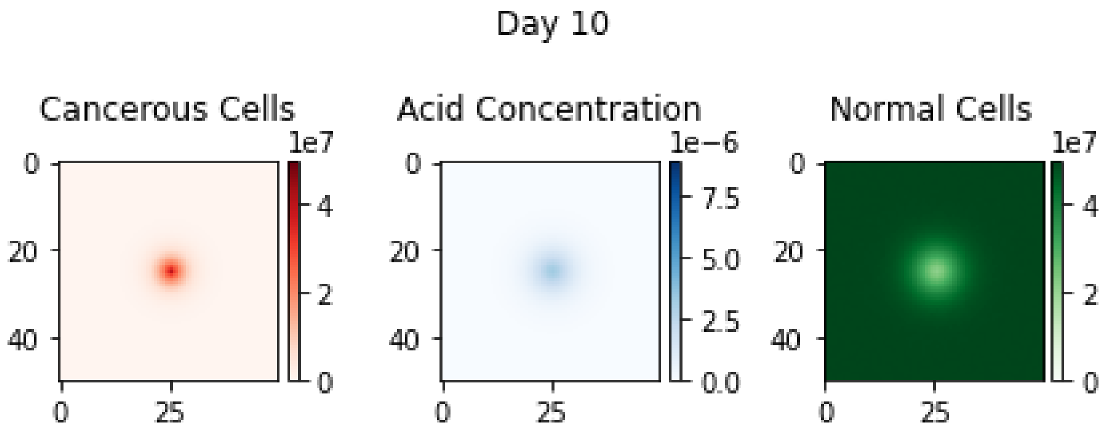
$$\Delta t = 300 \text{ s} \quad (27)$$

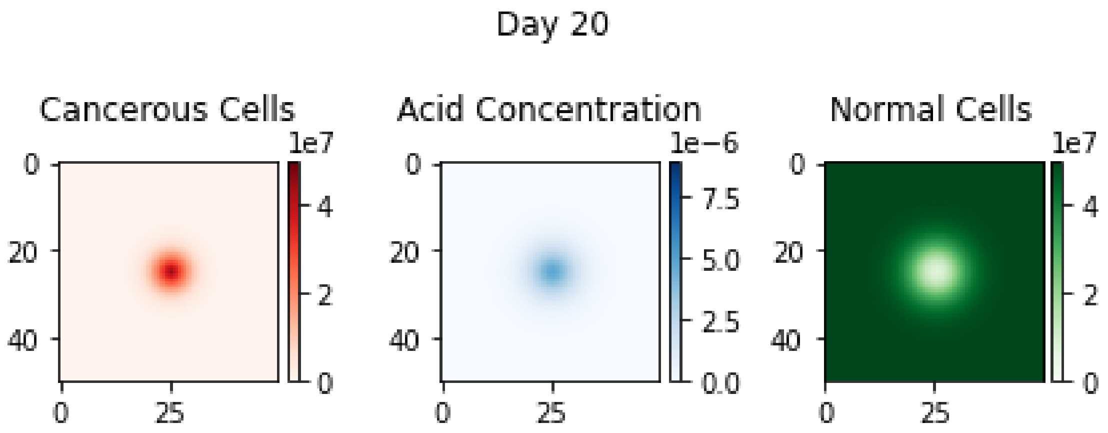
The variable Δt is chosen so that it satisfies equations (17) and (18) at the same time. There are four kinds of simulation conducted, which will be explained in the following subsection.

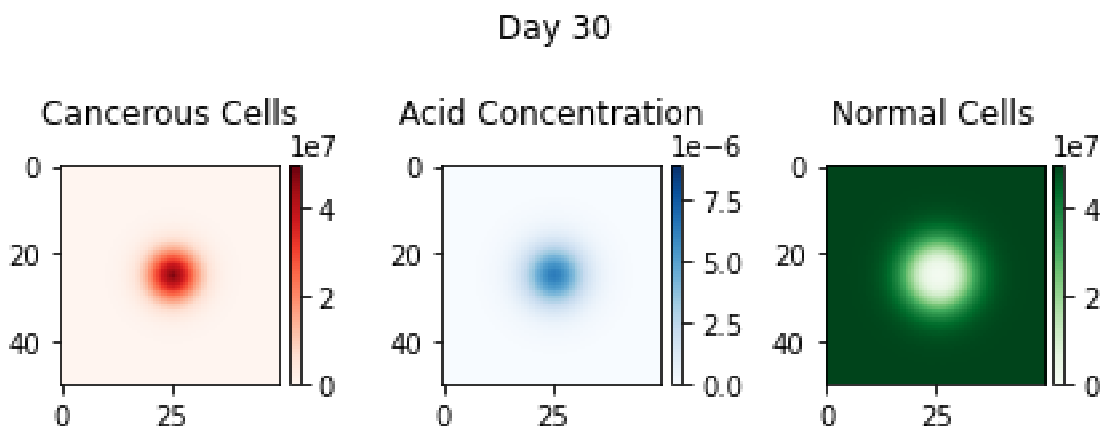
5.1 Without Treatment

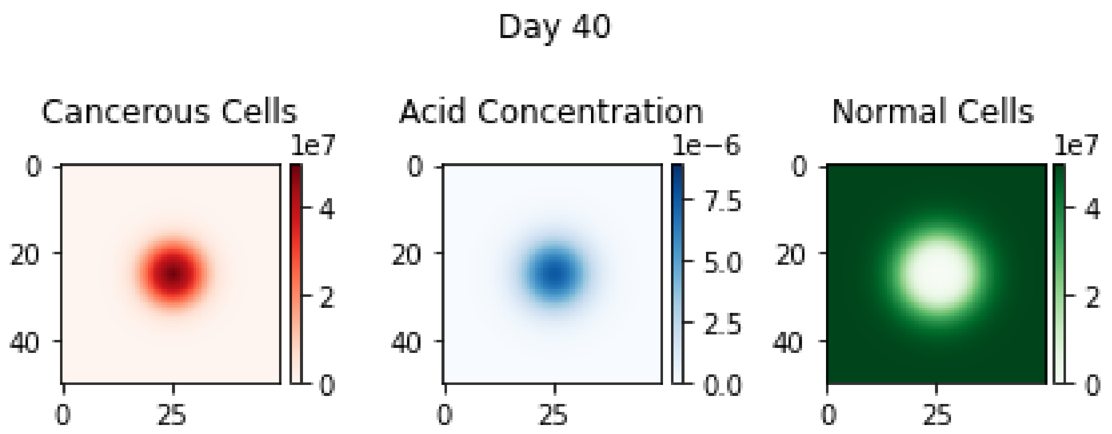
The first one is cancer propagation without any treatment. We let cancer spread to the surrounding tissue for 60 days and examine how fast cancer propagates. The total CPU time taken to conduct the simulation is 251.726 seconds. The result of the simulation is shown below.

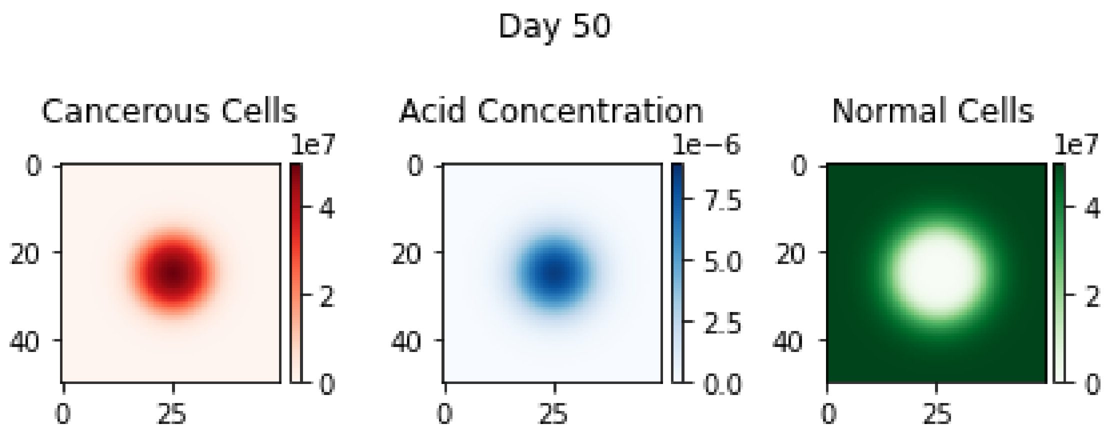

Day 0
Cancerous Cells
1e7
0
20
40
0
25
4
2
0
Acid Concentration
1e−6
0
20
40
0
25
7.5
5.0
2.5
0.0
Normal Cells
1e7
0
20
40
0
25
4
2
0

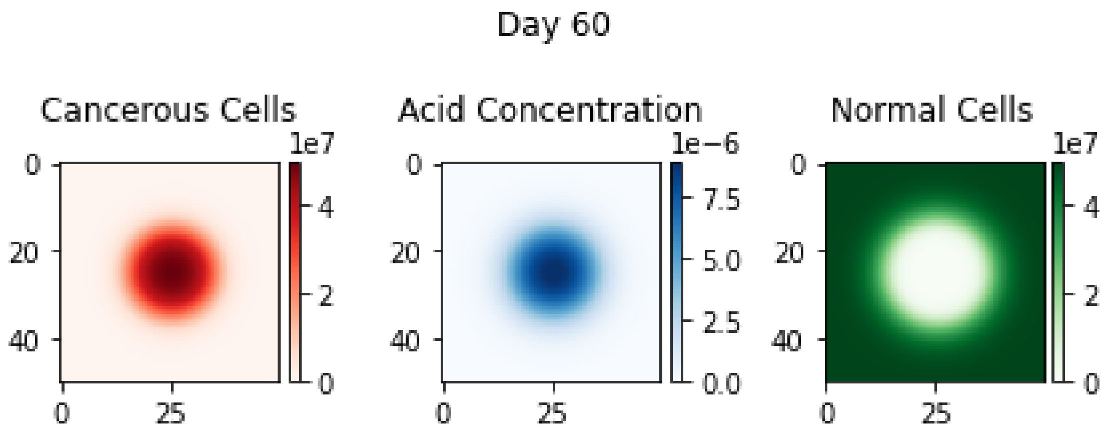

Day 10
Cancerous Cells
1e7
0
20
40
0
25
4
2
0
Acid Concentration
1e−6
0
20
40
0
25
7.5
5.0
2.5
0.0
Normal Cells
1e7
0
20
40
0
25
4
2
0


Day 20
Cancerous Cells
1e7
0
20
40
0
25
4
2
0
Acid Concentration
1e−6
0
20
40
0
25
7.5
5.0
2.5
0.0
Normal Cells
1e7
0
20
40
0
25
4
2
0


Day 30
Cancerous Cells
1e7
0
20
40
0
25
4
2
0
Acid Concentration
1e−6
7.5
5.0
2.5
0.0
Normal Cells
1e7
4
2
0


Day 40
Cancerous Cells
1e7
0
20
40
0
25
4
2
0
Acid Concentration
1e−6
7.5
5.0
2.5
0.0
Normal Cells
1e7
4
2
0

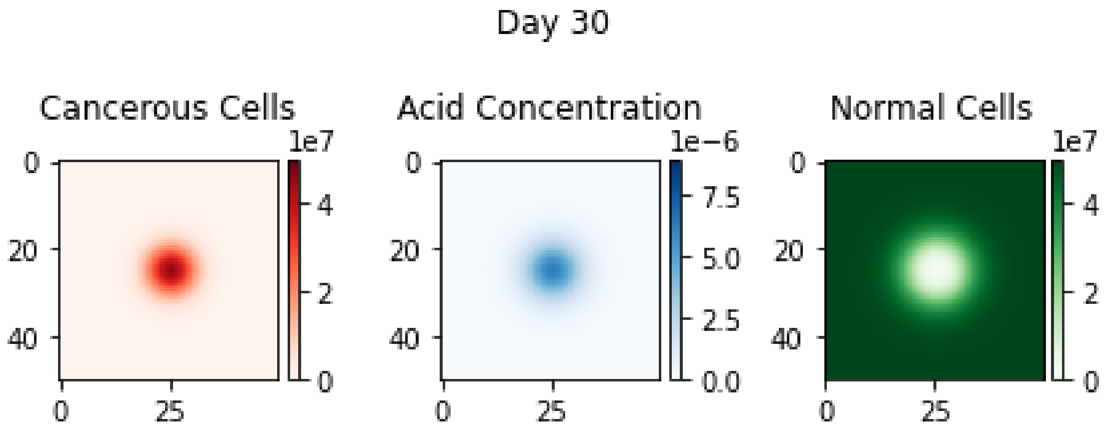

Day 50
Cancerous Cells
1e7
0
20
40
0
25
4
2
0
Acid Concentration
1e−6
7.5
5.0
2.5
0.0
Normal Cells
1e7
4
2
0

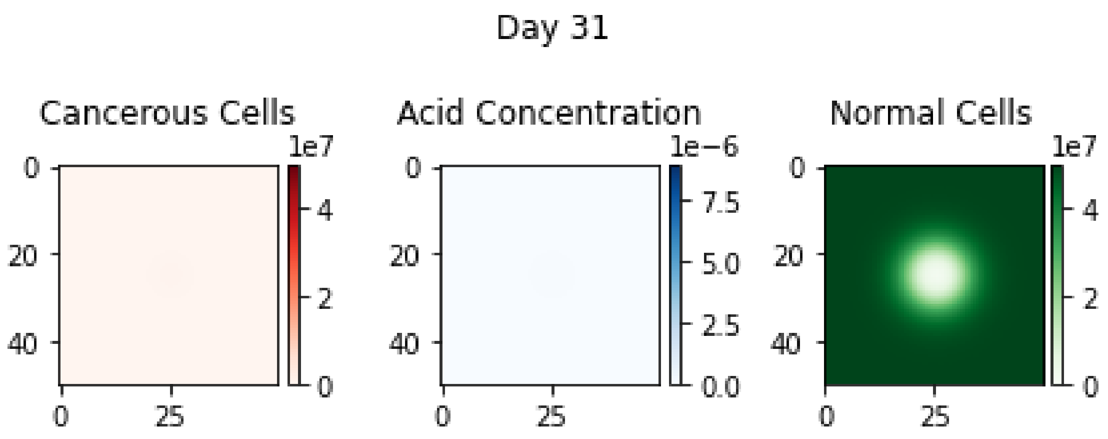


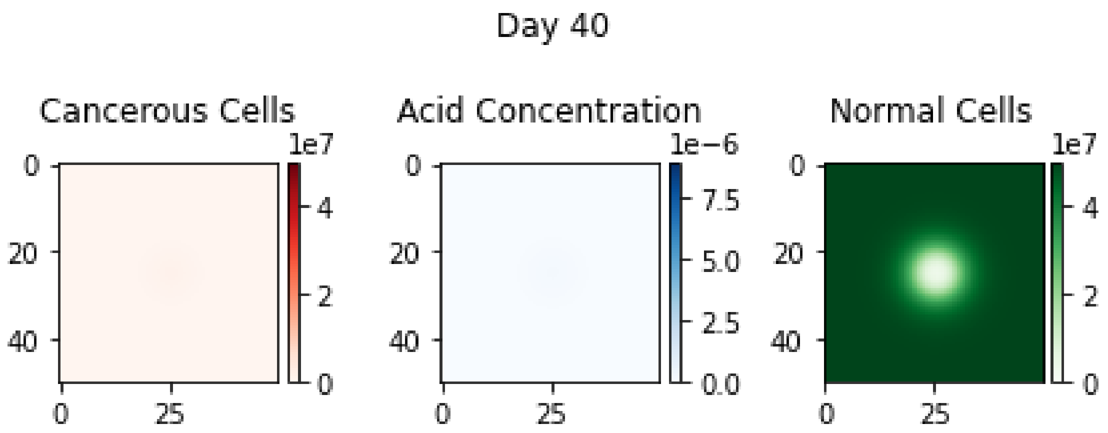
As it can be seen from the simulation results, cancer originally starts small at the center of the tissue. It produces excess acid resulting in the death of the surrounding healthy tissue. With the healthy tissue being greatly diminished, it allows the cancer cells to diffuse to the surrounding area. By day 60, we can see that the cancer cells have grown with a diameter of almost 2.5 cm, as shown in the figure above.

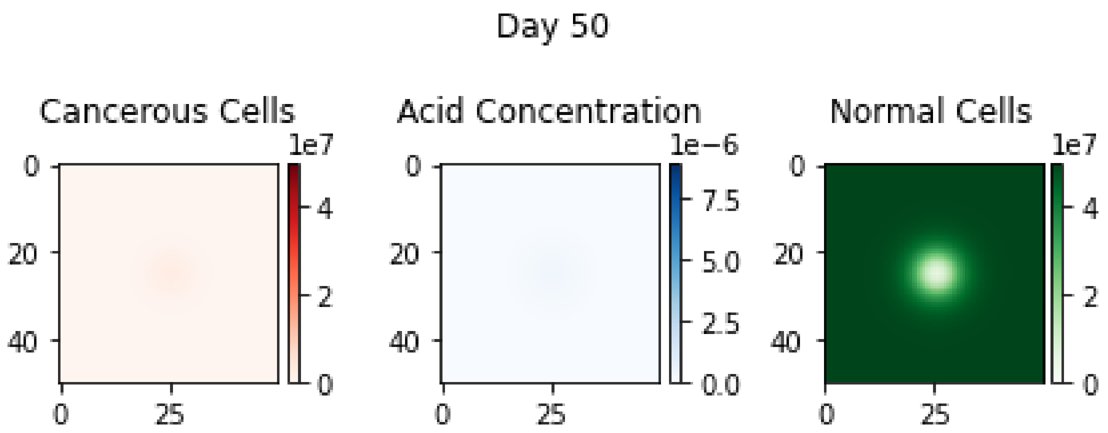
5.2 With Surgical Treatment

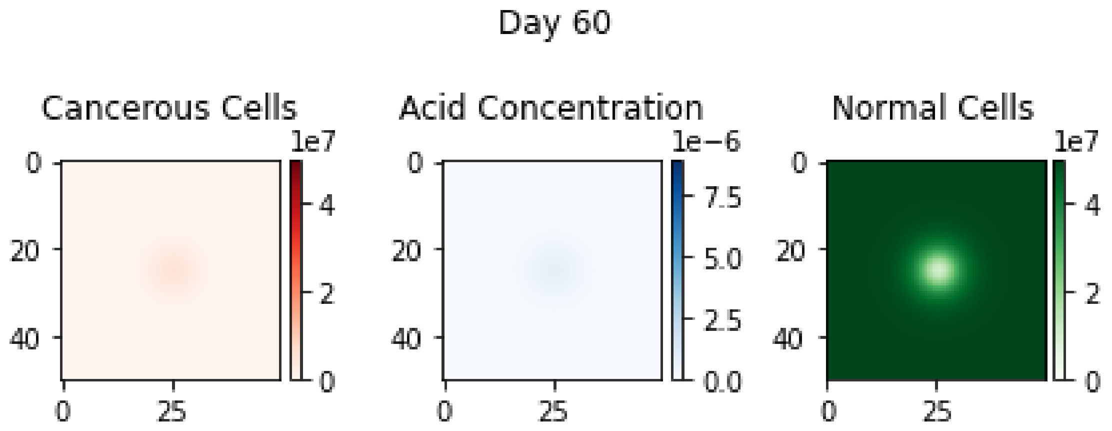
The second one is a simulation with surgical treatment. We let cancer spread for as long as one month, and a surgical treatment, which is assumed to remove 99% of the cancer cells, is conducted at the end of the month (day 30). In this simulation, we evaluate the model for up to 60 days, and we then examine whether cancer will grow back or will be contained after the surgery. The total CPU time taken to conduct this simulation is 333.234 seconds. The result of the simulation is shown as follows.

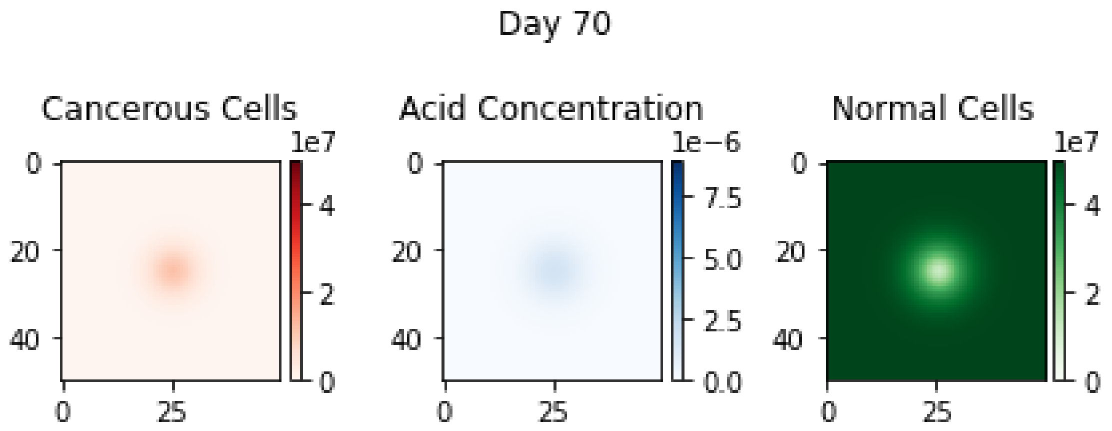


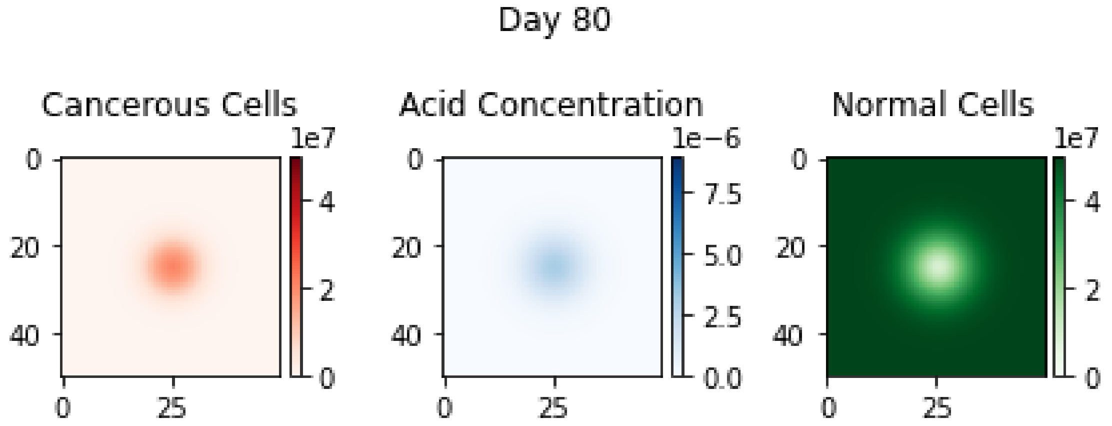

Day 31
Cancerous Cells
Acid Concentration
Normal Cells
1e7
1e−6
1e7
0
20
40
0
25
0
2
4
0.0
2.5
5.0
7.5


Day 40
Cancerous Cells
Acid Concentration
Normal Cells
1e7
1e−6
1e7
0
20
40
0
25
0
2
4
0.0
2.5
5.0
7.5


Day 50
Cancerous Cells
Acid Concentration
Normal Cells
1e7
1e−6
1e7
0
20
40
0
25
0
2
4
0.0
2.5
5.0
7.5







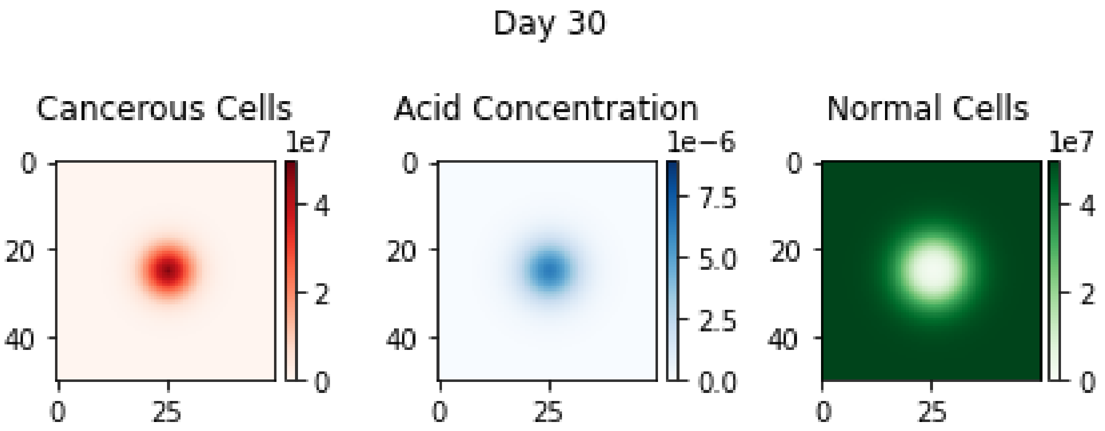
We can clearly see that right after the surgery on day 31, no visible cancer is seen. From day 31 to day 60, we can also see that the healthy cells start to grow back in the region previously occupied by the cancer cells as the white region shrinks. But it turns out that we can start to see a visible cancerous region by day 60. Even if the surgery only left one percent of the cancer cells' population at the time before the surgery, the cancer cells can grow back in time. As cancer grows back, the white region on the leftmost figure starts to get bigger again, indicating the death of the surrounding healthy cells.

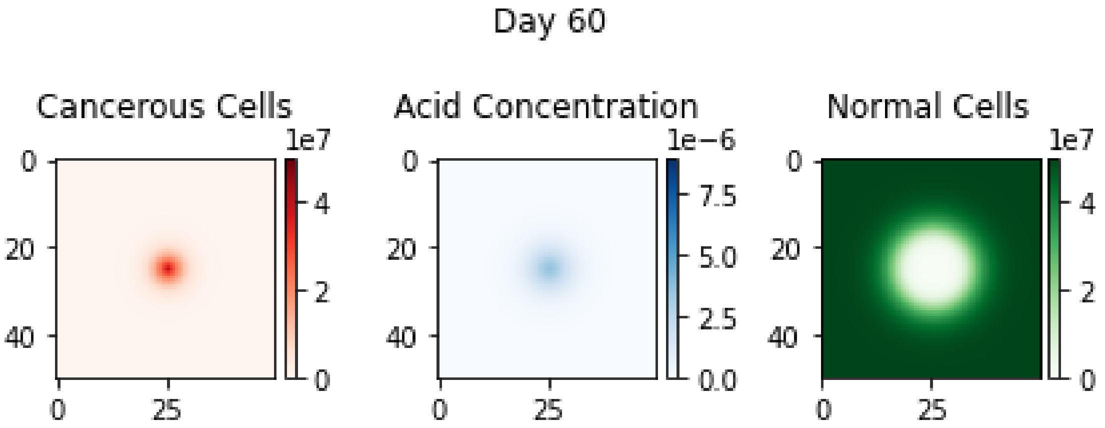
5.3 With Chemotherapy Treatment

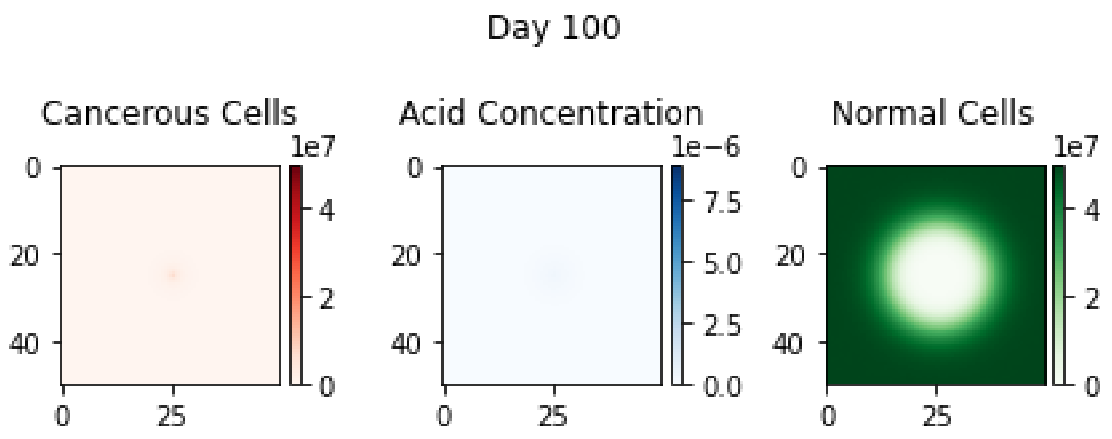
Next, we consider the chemotherapy treatment. We let cancer spread for as long as one month, and the chemotherapy treatment starts at the beginning of the second month. The function of the chemotherapy drug that is used in this model is given by

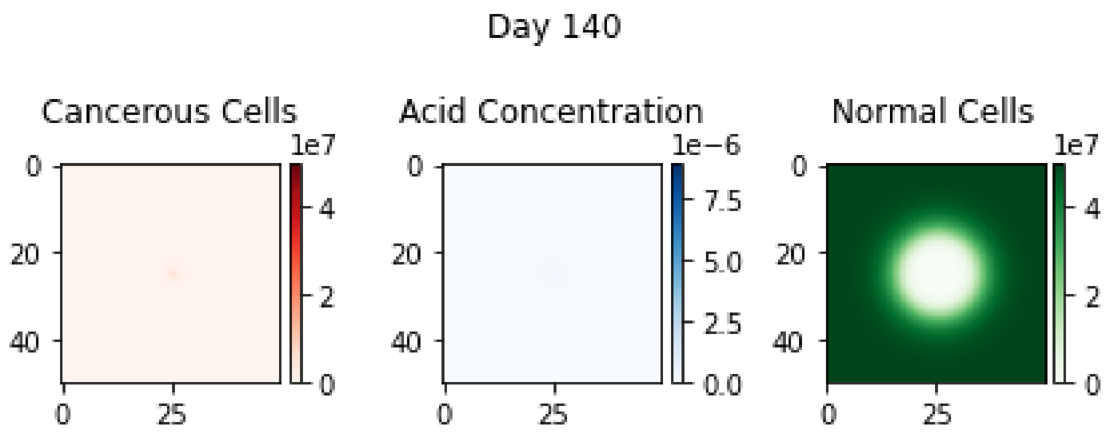
$$T(t) = \begin{cases} k, & t \in T_{on} - \text{chemotherapy on} \\ 0, & t \in T_{off} - \text{chemotherapy off} \end{cases}, \quad (26)$$

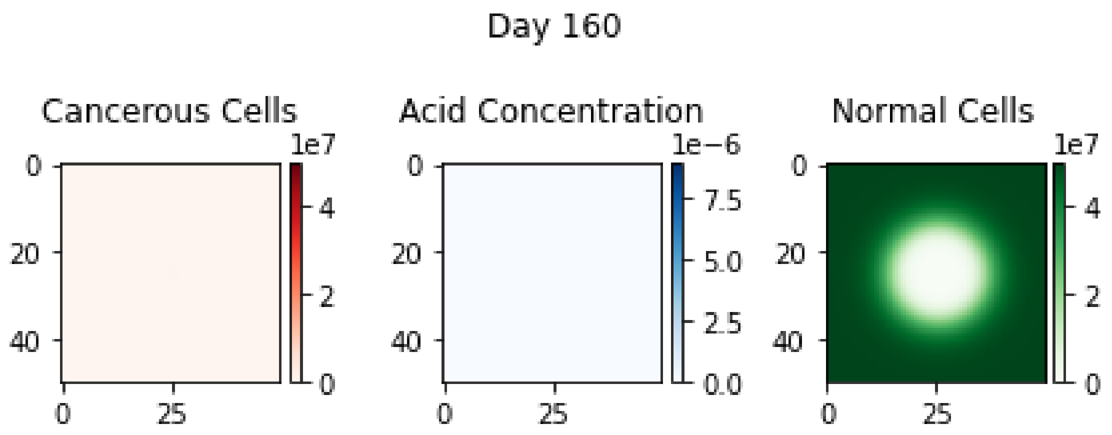
where k is a constant and T_{on}, T_{off} follow a periodic pattern. In this simulation, we set the cycle to be four weeks, consisting of 2 weeks of chemotherapy followed by two weeks of resting period. We set the value of k = 1. In total, six cycles of chemotherapy are given. The chemotherapy starts on day 31 and ends on day 156 after six cycles. According to [8], fairly effective chemotherapy happens when the ratio of d_C and r_C is more than 2.25. In this simulation, we set $d_C = 3r_C$ and $d_H = 2r_H$ with an assumption that the chemotherapy drug is more potent to cancer cells compared to normal cells. We evaluate this simulation for 300 days, and the total CPU time taken to conduct this simulation is 1252.605 seconds. The result of the simulation is shown as follows.

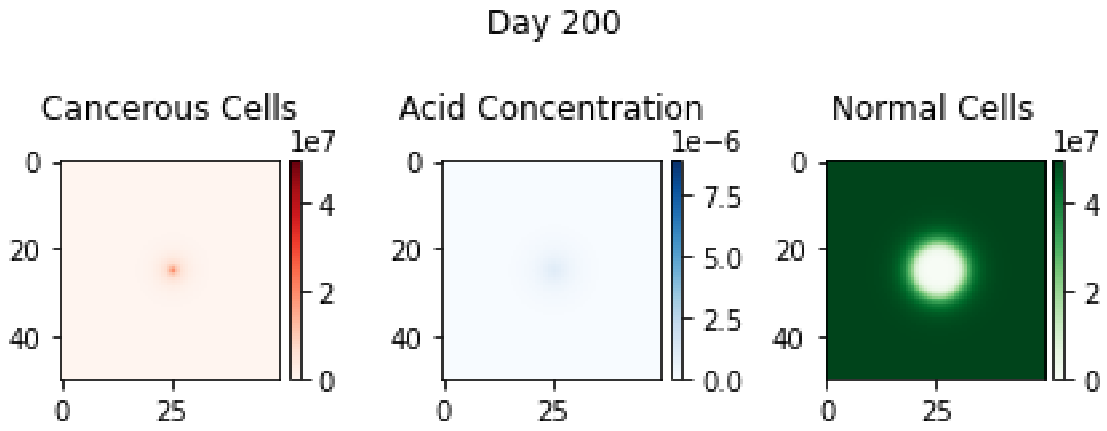


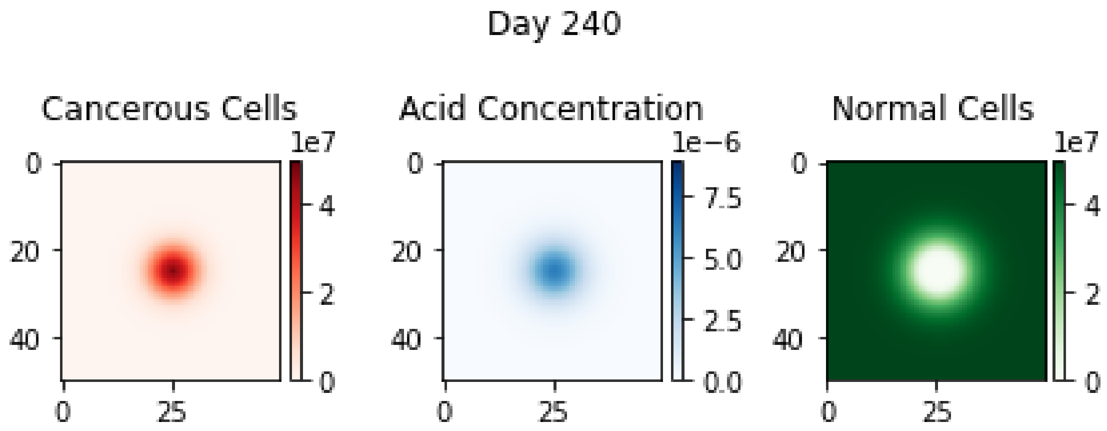


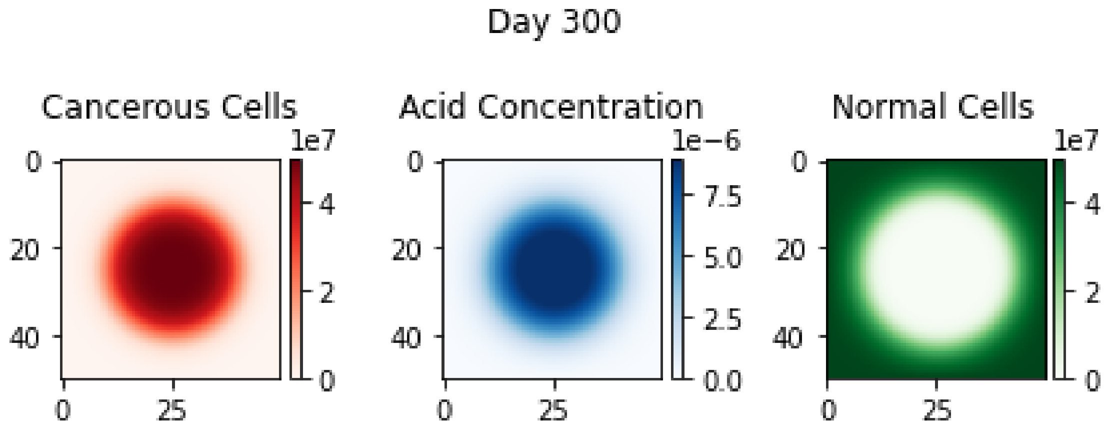

Day 100
Cancerous Cells
Acid Concentration
Normal Cells
1e7
1e−6
1e7
0
20
40
0
25
0
2
4
0.0
2.5
5.0
7.5


Day 140
Cancerous Cells
Acid Concentration
Normal Cells
1e7
1e−6
1e7
0
20
40
0
25
0
2
4
0.0
2.5
5.0
7.5


Day 160
Cancerous Cells
Acid Concentration
Normal Cells
1e7
1e−6
1e7
0
20
40
0
25
0
2
4
0.0
2.5
5.0
7.5



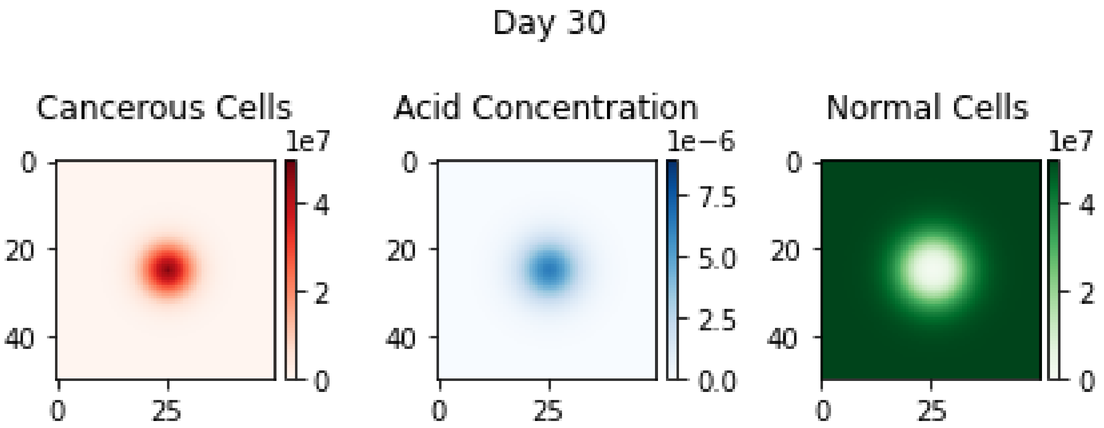


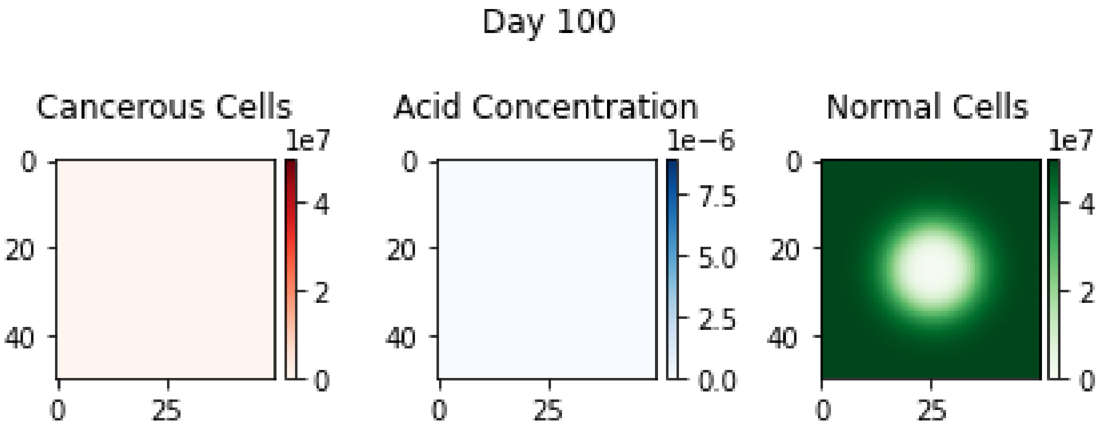


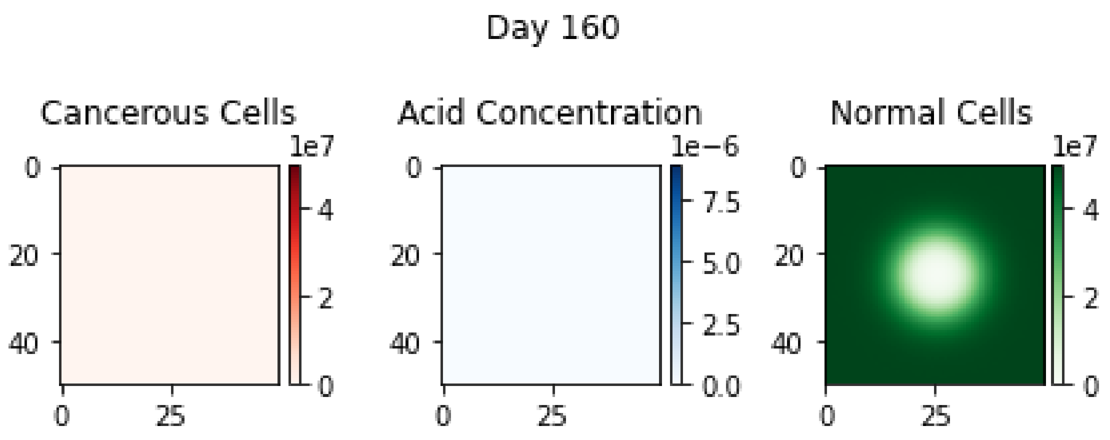
From the figures above, we can see that during the chemotherapy period, the cancerous region starts to shrink slowly. By day 160, there is no visible cancerous region seen. However, we can also see that the white region in the leftmost figure also grows and shrinks periodically. During chemotherapy weeks, it grows, and during the resting period, it shrinks. This is because healthy tissue that is actively multiplying is also affected by the chemotherapy drugs, albeit less potent. Once the chemotherapy is over, the healthy tissue starts growing back, and the white region shrinks. However, at the same time, we can see that by day 200, a small cancerous region starts to appear again. From then, the size of the cancerous region keeps growing, and by day 300, the size of the cancerous region is even bigger than day 30. From this simulation, we can conclude that six cycles of chemotherapy treatment alone are not likely to be enough for invasive cancer.

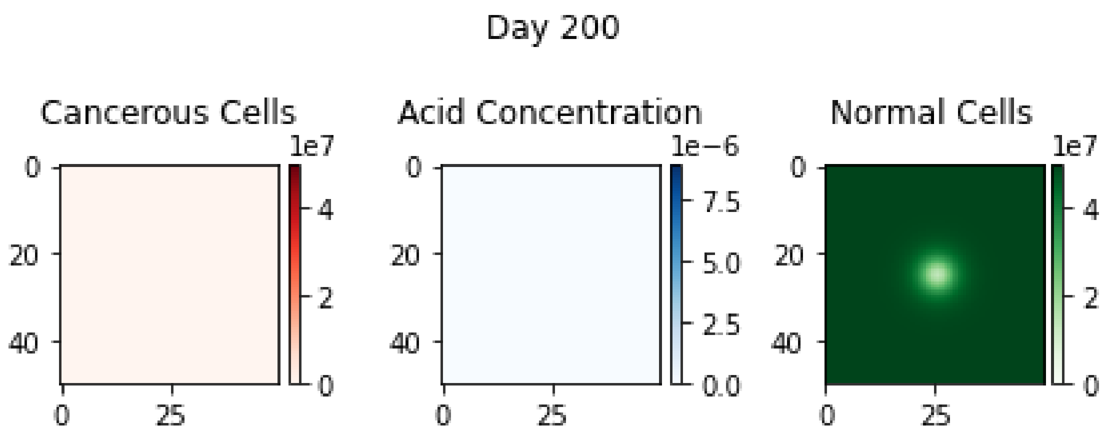
5.4 With Surgical and Chemotherapy Treatment

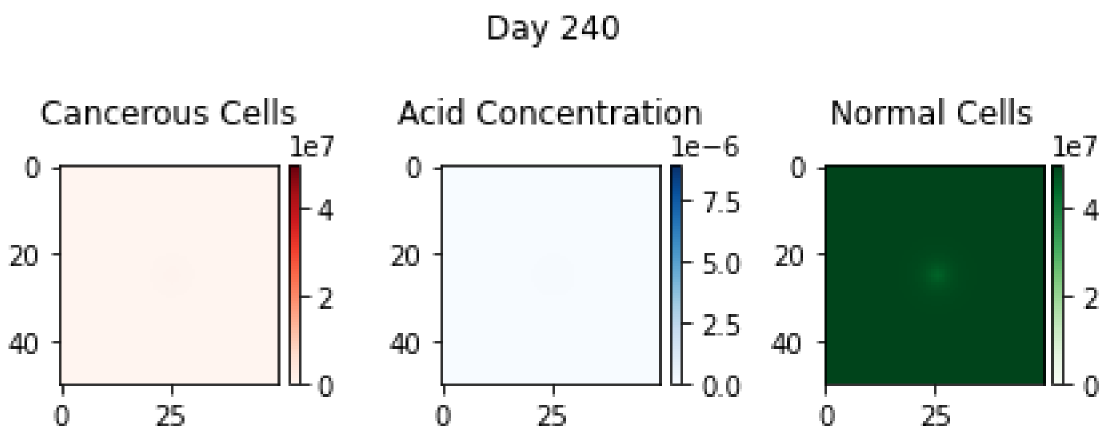
Finally, the last one we consider is a combination of surgical and chemotherapy treatment. The surgical treatment is conducted on day 31, followed by six cycles of chemotherapy treatments. We evaluate this simulation for 300 days, and the total CPU time taken to conduct this simulation is 1263.881 seconds. The result of the simulation is given by the following figures.

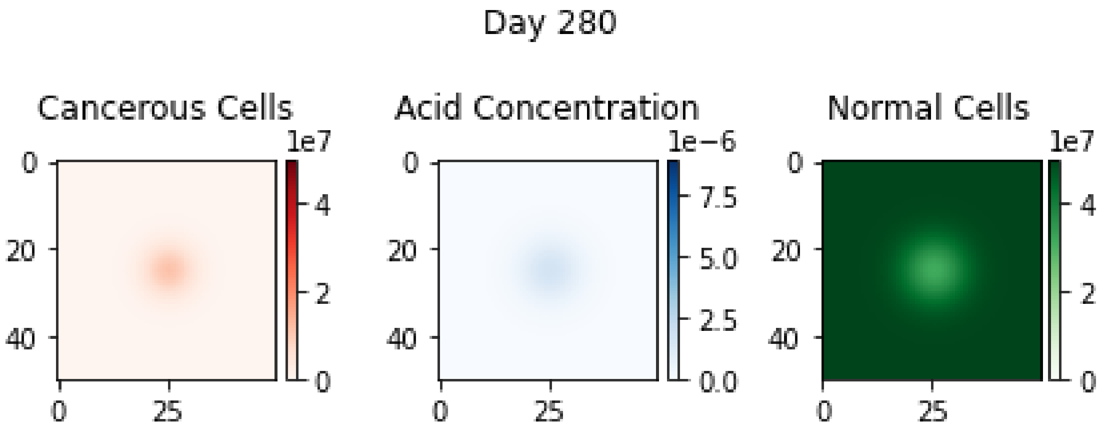


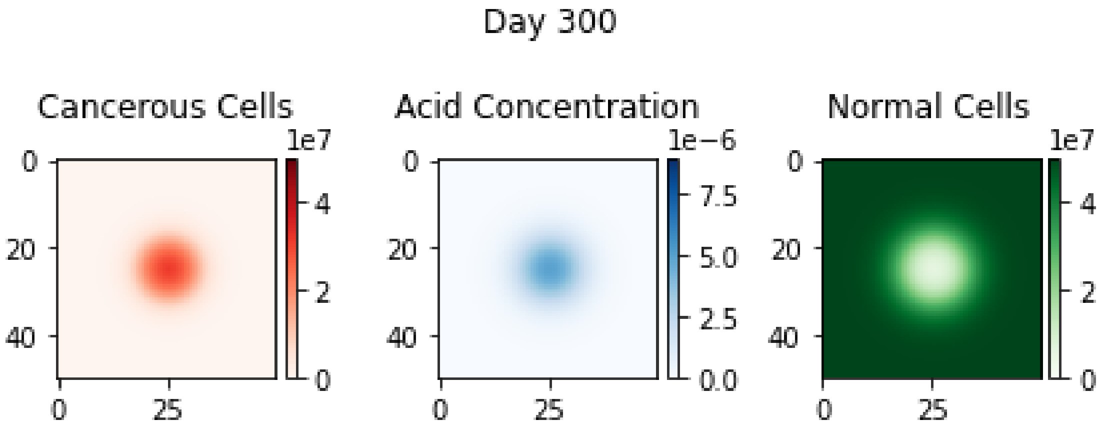



Day 160
Cancerous Cells
Acid Concentration
Normal Cells
1e7
1e−6
1e7
0
20
40
0
25
0
2
4
0.0
2.5
5.0
7.5


Day 200
Cancerous Cells
Acid Concentration
Normal Cells
1e7
1e−6
1e7
0
20
40
0
25
0
2
4
0.0
2.5
5.0
7.5


Day 240
Cancerous Cells
Acid Concentration
Normal Cells
1e7
1e−6
1e7
0
20
40
0
25
0
2
4
0.0
2.5
5.0
7.5





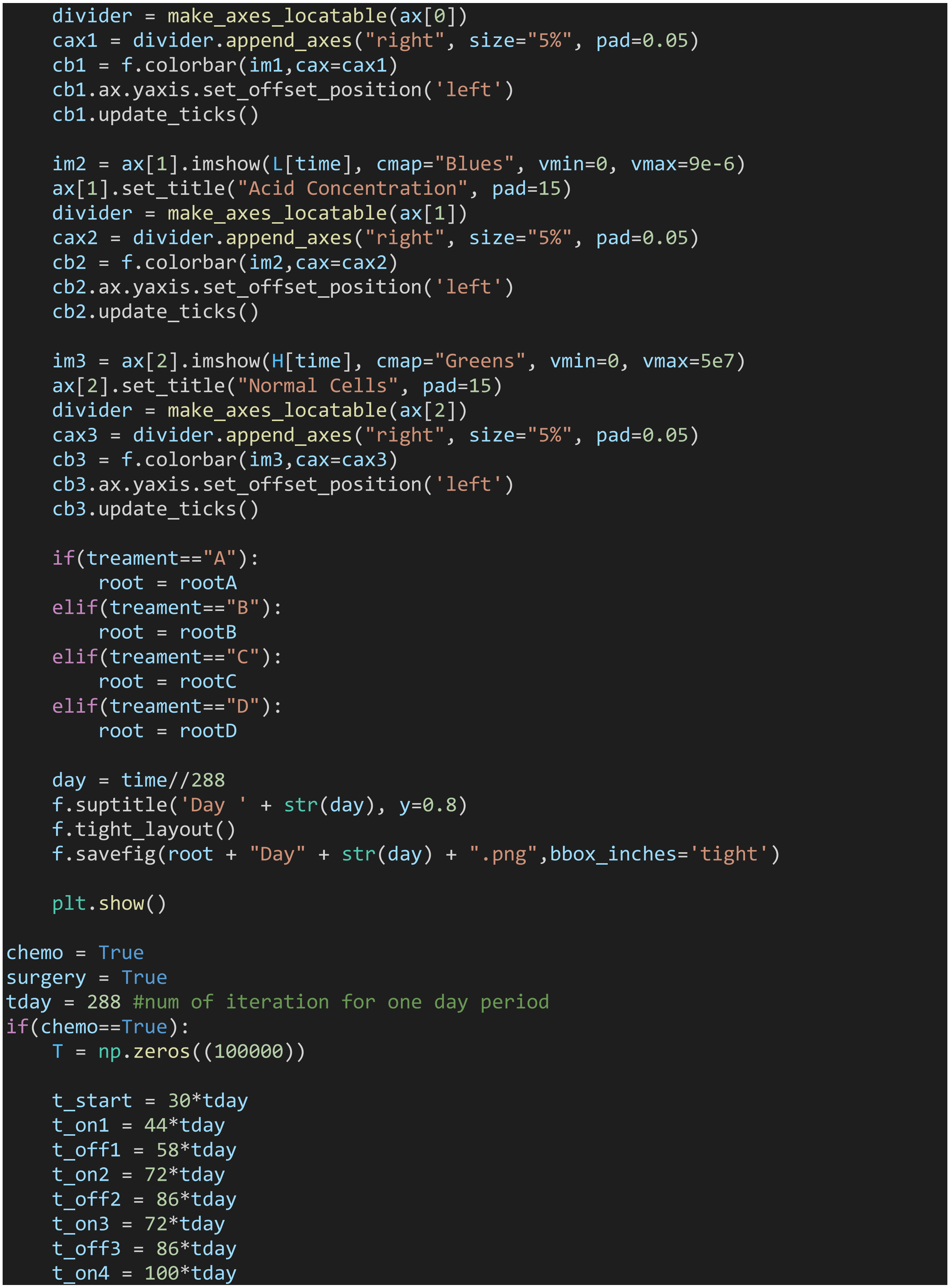
The simulation result above shows that the cancerous region still grows back even after a combination of surgery and chemotherapy treatment. However, we can also see that there is no visible cancerous region until day 280. Even if it grows back, the size and the density of the cancerous region at day 300 is much lesser compared to when only chemotherapy treatment is given. This simulation demonstrates how a combination of surgery and chemotherapy treatment is more potent than surgery or chemotherapy alone. However, six cycles of chemotherapy are also likely not enough to eradicate all cancer cells. Thus, we can conclude that surgery and more than six cycles of chemotherapy treatment are likely needed to eradicate all the invasive cancerous regions.

6. Conclusion

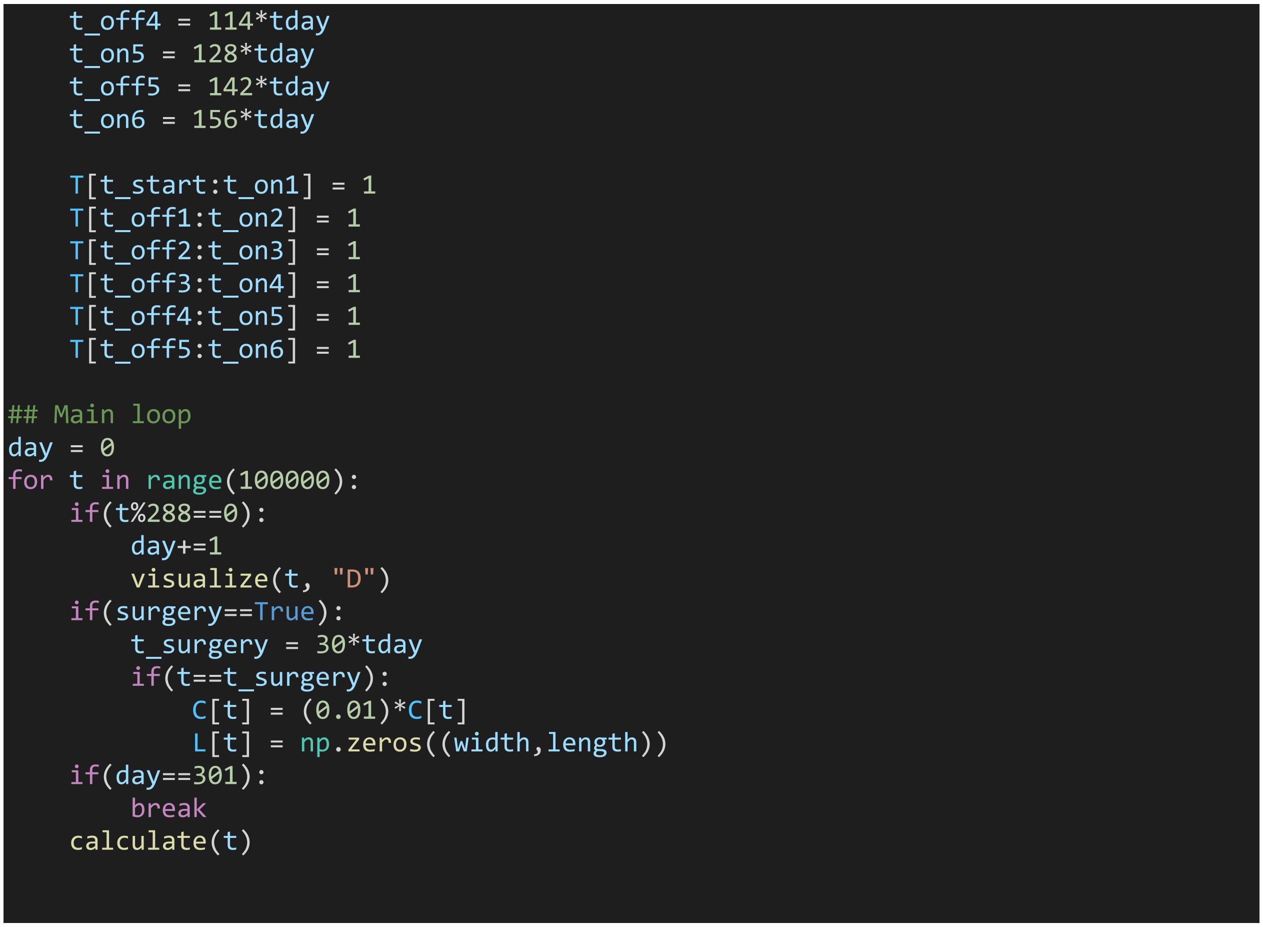
In this report, the propagation of invasive cancer has been modeled and simulated. In this model, the propagation of cancer is mediated by the production of excess acid concentration in which healthy cells are unable to survive. Cancer cells can only propagate if the density of healthy cells in the surrounding has been greatly diminished. The initial condition used in this simulation is an exponential function of position starting from the center of the region of interest, and the boundary condition used in this simulation is the Neumann boundary condition under the assumption that no cells can get in or get out from the region of interest. There are four types of simulation conducted for different lengths in this report: without treatment (60 days), with surgical treatment (80 days), with chemotherapy treatment (300 days), with a combination of surgical and chemotherapy treatment (300 days). The simulation shows that even with a combination of surgical and six cycles of chemotherapy treatment, the cancerous region still grows back by day 300. Therefore, we can conclude a combination of surgery and six cycles of chemotherapy treatment is not enough. A combination with other therapy or more cycles of chemotherapy is likely needed to completely eradicate invasive cancer cells.

Appendix A. Code

```
import numpy as np
import matplotlib.pyplot as plt

rootA = "Results/A/"
rootB = "Results/B/"
rootC = "Results/C/"
rootD = "Results/D/"

x,y = np.mgrid[-2.5:2.5:0.1, -2.5:2.5:0.1]
length, width = x.shape

H = np.zeros((100000,width,length))
C = np.zeros((100000,width,length))
L = np.zeros((100000,width,length))

#initialize cancer cell, healthy cell and acid concentration
H[0] = np.random.uniform(low=4.91e7, high=4.99e7, size=(width,length))
C[0] = 3e7*np.exp(-5*np.sqrt(x**2 + y**2))

# Reaction Parameters
r_H = 1e-6
r_C = 1e-6
r_L = 2.2e-17
d_H = 2*r_H
d_C = 3*r_C

k_H = 5e7
k_C = 5e7
alpha = 0.0
beta = 0.5
gamma = 1.1e-4

#Diffusion Parameters
D_C = 2e-10
D_L = 5e-6
T = 0

#time and space parameters
delta_space = 0.1
delta_time = 300

def calculate(t):
    #reaction terms
    reac_H = r_H*H[t]*(1 - (H[t]/k_H) - (alpha*C[t]/k_C)) - beta*L[t]*H[t] - d_H*T[t]*H[t]*(1 - (H[t]/k_H))
    reac_C = r_C*C[t]*(1 - (C[t]/k_C) - (alpha*H[t]/k_H)) - d_C*T[t]*C[t]*(1 - (C[t]/k_C))
    reac_L = r_L*C[t] - gamma*L[t]

    ##Diffusion terms

    #healthy cell
    diff_H = 0

    #cancer cell
    kappa = D_C*(1 - H[t]/k_H) #diffusion coefficient of cancer as a function of healthy cell
    diff_C = np.zeros((width,length))
    for i in range(width):
        for j in range(length):
            if((i==0) and (j==0)):
                diff_C[i,j] = (1/(delta_space**2))*((kappa[i,j] + kappa[i,j+1])*(C[t,i,j+1] - C[t,i,j]) + ((kappa[i,j] + kappa[i+1,j])*(C[t,i+1,j] - C[t,i,j])))
            elif((i==0) and (j==length-1)):
                diff_C[i,j] = (1/(delta_space**2))*((kappa[i,j] + kappa[i,j-1])*(C[t,i,j-1] - C[t,i,j]) + ((kappa[i,j] + kappa[i+1,j])*(C[t,i+1,j] - C[t,i,j])))
            elif((i==width-1) and (j==length-1)):
                diff_C[i,j] = (1/(delta_space**2))*((kappa[i,j] + kappa[i,j-1])*(C[t,i,j-1] - C[t,i,j]) + ((kappa[i,j] + kappa[i-1,j])*(C[t,i-1,j] - C[t,i,j])))
            elif((i==width-1) and (j==0)):
                diff_C[i,j] = (1/(delta_space**2))*((kappa[i,j] + kappa[i,j+1])*(C[t,i,j+1] - C[t,i,j]) + ((kappa[i,j] + kappa[i-1,j])*(C[t,i-1,j] - C[t,i,j])))
            elif(j==0):
                diff_C[i,j] = (1/(2*delta_space**2))*(2*(kappa[i,j+1] + kappa[i,j])*(C[t,i,j+1] - C[t,i,j]) + (kappa[i+1,j] + kappa[i,j])*(C[t,i+1,j] - C[t,i,j]) - (kappa[i,j] + kappa[i-1,j])*(C[t,i,j] - C[t,i-1,j]))
            elif(j==length-1):
                diff_C[i,j] = (1/(2*delta_space**2))*(2*(kappa[i,j] + kappa[i,j-1])*(C[t,i,j-1] - C[t,i,j]) + (kappa[i+1,j] + kappa[i,j])*(C[t,i+1,j] - C[t,i,j]) - (kappa[i,j] + kappa[i-1,j])*(C[t,i,j] - C[t,i-1,j]))
            elif(i==0):
                diff_C[i,j] = (1/(2*delta_space**2))*((kappa[i,j+1] + kappa[i,j])*(C[t,i,j+1] - C[t,i,j]) - (kappa[i,j] + kappa[i,j-1])*(C[t,i,j] - C[t,i,j-1]) + 2*(kappa[i,j] + kappa[i+1,j])*(C[t,i+1,j] - C[t,i,j]))
            elif(i==width-1):
                diff_C[i,j] = (1/(2*delta_space**2))*((kappa[i,j+1] + kappa[i,j])*(C[t,i,j+1] - C[t,i,j]) - (kappa[i,j] + kappa[i,j-1])*(C[t,i,j] - C[t,i,j-1]) + 2*(kappa[i-1,j] + kappa[i,j])*(C[t,i-1,j] - C[t,i,j]))
            else:
                diff_C[i,j] = (1/(2*delta_space**2))*((kappa[i,j+1] + kappa[i,j])*(C[t,i,j+1] - C[t,i,j]) - (kappa[i,j] + kappa[i,j-1])*(C[t,i,j] - C[t,i,j-
1]) + (kappa[i+1,j] + kappa[i,j])*(C[t,i+1,j] - C[t,i,j]) - (kappa[i,j] + kappa[i
-1,j])*(C[t,i,j] - C[t,i-1,j]))

    #acid concetration
    diff_L = np.zeros((width,length))
    for i in range(width):
        for j in range(length):
            if((i==0) and (j==0)):
                diff_L[i,j] = (2/(delta_space**2))*(L[t,i,j+1] - L[t,i,j] + L[t,i
+1,] - L[t,i,j])
            elif((i==0) and (j==length-1)):
                diff_L[i,j] = (2/(delta_space**2))*(L[t,i,j-
1] - L[t,i,j] + L[t,i+1,] - L[t,i,j])
            elif((i==width-1) and (j==length-1)):
                diff_L[i,j] = (2/(delta_space**2))*(L[t,i,j-
1] - L[t,i,j] + L[t,i-1,] - L[t,i,j])
            elif((i==width-1) and (j==0)):
                diff_L[i,j] = (2/(delta_space**2))*(L[t,i,j+1] - L[t,i,j] + L[t,i
-1,] - L[t,i,j])
            elif(j==0):
                diff_L[i,j] = (1/(delta_space**2))*(2*(L[t,i,j+1] - L[t,i,j]) + (
L[t,i+1,j] - 2*L[t,i,j] + L[t,i-1,j]))
            elif(j==length-1):
                diff_L[i,j] = (1/(delta_space**2))*(2*(L[t,i,j-
1] - L[t,i,j]) + (L[t,i+1,j] - 2*L[t,i,j] + L[t,i-1,j]))
            elif(i==0):
                diff_L[i,j] = (1/(delta_space**2))*((L[t,i,j+1] - 2*L[t,i,j] + L[
t,i,j-1]) + 2*(L[t,i+1,j] - L[t,i,j]))
            elif(i==width-1):
                diff_L[i,j] = (1/(delta_space**2))*((L[t,i,j+1] - 2*L[t,i,j] + L[
t,i,j-1]) + 2*(L[t,i-1,j] - L[t,i,j]))
            else:
                diff_L[i,j] = (1/(delta_space**2))*((L[t,i,j+1] - 2*L[t,i,j] + L[
t,i,j-1]) + (L[t,i+1,j] - 2*L[t,i,j] + L[t,i-1,j]))


    C[t+1] = C[t] + delta_time*(D_C*diff_C + reac_C)
    H[t+1] = H[t] + delta_time*(diff_H + reac_H)
    L[t+1] = L[t] + delta_time*(D_L*diff_L + reac_L)

    #make sure that it does not go negative
    invalid_idx = np.where(C[t+1]<0)
    C[t+1,invalid_idx] = 0

from mpl_toolkits.axes_grid1 import make_axes_locatable

def visualize(time, treament):
    plt.clf()
    f,ax = plt.subplots(1,3)

    im1 = ax[0].imshow(C[time], cmap="Reds", vmin=0, vmax=5e7)
    ax[0].set_title("Cancerous Cells", pad=15)
```



```
    divider = make_axes_locatable(ax[0])
    cax1 = divider.append_axes("right", size="5%", pad=0.05)
    cb1 = f.colorbar(im1,cax=cax1)
    cb1.ax.yaxis.set_offset_position('left')
    cb1.update_ticks()

    im2 = ax[1].imshow(L[time], cmap="Blues", vmin=0, vmax=9e-6)
    ax[1].set_title("Acid Concentration", pad=15)
    divider = make_axes_locatable(ax[1])
    cax2 = divider.append_axes("right", size="5%", pad=0.05)
    cb2 = f.colorbar(im2,cax=cax2)
    cb2.ax.yaxis.set_offset_position('left')
    cb2.update_ticks()

    im3 = ax[2].imshow(H[time], cmap="Greens", vmin=0, vmax=5e7)
    ax[2].set_title("Normal Cells", pad=15)
    divider = make_axes_locatable(ax[2])
    cax3 = divider.append_axes("right", size="5%", pad=0.05)
    cb3 = f.colorbar(im3,cax=cax3)
    cb3.ax.yaxis.set_offset_position('left')
    cb3.update_ticks()

    if(treament=="A"):
        root = rootA
    elif(treament=="B"):
        root = rootB
    elif(treament=="C"):
        root = rootC
    elif(treament=="D"):
        root = rootD

    day = time//288
    f.suptitle('Day ' + str(day), y=0.8)
    f.tight_layout()
    f.savefig(root + "Day" + str(day) + ".png",bbox_inches='tight')

    plt.show()

chemo = True
surgery = True
tday = 288 #num of iteration for one day period
if(chemo==True):
    T = np.zeros((100000))

    t_start = 30*tday
    t_on1 = 44*tday
    t_off1 = 58*tday
    t_on2 = 72*tday
    t_off2 = 86*tday
    t_on3 = 72*tday
    t_off3 = 86*tday
    t_on4 = 100*tday
```



```
    t_off4 = 114*tday
    t_on5 = 128*tday
    t_off5 = 142*tday
    t_on6 = 156*tday

    T[t_start:t_on1] = 1
    T[t_off1:t_on2] = 1
    T[t_off2:t_on3] = 1
    T[t_off3:t_on4] = 1
    T[t_off4:t_on5] = 1
    T[t_off5:t_on6] = 1

## Main loop
day = 0
for t in range(100000):
    if(t%288==0):
        day+=1
        visualize(t, "D")
    if(surgery==True):
        t_surgery = 30*tday
        if(t==t_surgery):
            C[t] = (0.01)*C[t]
            L[t] = np.zeros((width,length))
    if(day==301):
        break
    calculate(t)
```

Appendix B. Computer Specification and CPU Time

Operating System	Windows 10
Processor	Intel(R) Core(TM) i7-10700KF CPU @ 3.80GHz (16 CPUs), ~3.8GHz
Memory	32768MB RAM
Language	Python 3

Simulation	Evaluation Period (days)	Running Time (s)
Without Treatment	60	251.726
With Surgical Treatment	80	333.234
With Chemotherapy Treatment	300	1252.605
With Surgical and Chemotherapy Treatment	300	1263.881

References

[1] *What is cancer?* National Cancer Institute. (n.d.). https://www.cancer.gov/about-cancer/understanding/what-is-cancer.

[2] Tammela, T., & Sage, J. (2020). Investigating tumor heterogeneity in mouse models. *Annual Review of Cancer Biology*, *4*(1), 99–119. https://doi.org/10.1146/annurev-cancerbio-030419-033413

[3] Gatenby RA, Gawlinski ET. (1996). A reaction-diffusion model of cancer invasion. *Cancer Research*, 56(24):5745-53. PMID: 8971186.

[4] Stubbs M, Rodrigues L, Howe FA, Wang J, Jeong KS, Veech RL, Griffiths JR. Metabolic consequences of a reversed pH gradient in rat tumors. Cancer Res. 1994 Aug 1;54(15):4011-6. PMID: 8033132.

[5] Tracqui, P., Cruywagen, G. C., Woodward, D. E., Bartoo, G. T., Murray, J. D., & Alvord, E. C. (1995). A mathematical model of glioma growth: The effect of chemotherapy on spatio-temporal growth. *Cells Proliferation*, *28*(1), 17–31. https://doi.org/10.1111/j.1365-2184.1995.tb00036.x

[6] Dale, P. D., Sherratt, J. A., & Maini, P. K. (1994). The speed of corneal epithelial wound healing. *Applied Mathematics Letters*, *7*(2), 11–14. https://doi.org/10.1016/0893-9659(94)90022-1

[7] CRC Handbook of chemistry and PHYSICS, 92nd Edition. (2011). https://doi.org/10.1201/b17379

[8] Swanson, K.R., Alvord, E.C. & Murray, J.D. (2002). Quantifying Efficacy of Chemotherapy of Brain Tumors with Homogeneous and Heterogeneous Drug Delivery. *Acta Biotheor* 50, 223–237. https://doi.org/10.1023/A:1022644031905