THE SIMEONI MODEL FOR TUMOR GROWTH

SOPHIE CARTER, MORGAN NIELSEN, MICHELLE WANG, SARAH WINTERS

Abstract. The Simeoni Model is a system of ordinary differential equations that represents tumor growth as it receives different treatments. After researching the constants, variables, and functions used in this model, we created graphs to see how changing the parameters affected the model and what it implied about tumor growth. We conclude that this model successfully represents immunotherapeutic and chemotherapeutic treatment on tumor cells, and we are curious about how the different constants are related to the various cancer treatments.

1. Background/Motivation

As a team, our search for a suitable situation or phenomenon to model encountered some obstacles. Initially, many concepts we considered were better suited for a Partial Differential Equation (PDE) rather than an Ordinary Differential Equation (ODE), which we aimed to avoid based on advice from our professor. Additionally, we sought a topic with a significant impact on people's lives. This led us to focus on analyzing the Simeoni Model (1.1), specifically its application in modeling cancerous tumor growth within the body [3].

Cancer is a complex group of diseases characterized by the uncontrolled division and growth of abnormal cells, which can infiltrate and destroy normal tissues within the body. Tumor growth, a hallmark of cancer, poses a critical threat as it can lead to the disruption of essential physiological processes by cutting off the oxygen and nutrient supply to surrounding tissues [1]. As cancer progresses, it often metastasizes, spreading to other parts of the body and compounding the challenges of treatment. The profound impact of cancer on individuals' lives, coupled with the urgent global quest for improved solutions and treatments, underscores the importance of studying and modeling tumor growth.

Our motivation for selecting the Simeoni Model as the focus of our research stems from its application in addressing the intricate dynamics of cancerous tumor growth within the body. By delving into this modeling framework, we aim to contribute to existing research efforts seeking to better understand solutions and treatments for cancer. The Simeoni Model provides a valuable platform for simulating and understanding the intricate processes involved in tumor development. Analyzing and modifying this

Date: December 7, 2023.

model can offer insights into the underlying mechanisms of cancer progression, potentially leading to more effective strategies for intervention and treatment. Ultimately, our research endeavors aspires to contribute to the broader goal of preventing the devastating consequences of uncontrolled tumor growth and, in turn, improving the prognosis and quality of life for individuals facing cancer diagnoses.

2. Modeling

While the Simeoni model and the SIR model represent different phenomena, they share similarities in certain aspects, such as their mathematical structures and underlying conceptual framework, which illustrate the relevance of our project to our classwork. In particular, both models utilize a system of differential equations to describe the dynamics of populations. In both cases, a portion of the population undergoes a transition, such as infection or initialization of cell death, in one equation, and this affected population is then incorporated into the subsequent equation. This sequential flow of individual entities from one compartment to another reflects the progression of the affected population through distinct stages.

$$\begin{aligned}
\frac{dZ_1}{dt} &= TGF(t) - k_1 c(t) Z_1(t), \\
\frac{dZ_2}{dt} &= k_1 c(t) Z_1(t) - k_2 Z_2(t), \\
\frac{dZ_3}{dt} &= k_2 Z_2(t) - k_2 Z_3(t), \\
\frac{dZ_4}{dt} &= k_2 Z_3(t) - k_2 Z_4(t)
\end{aligned} \tag{1.1}$$

$$TGF(t) = \frac{\lambda_0 Z_1(t)}{[1 + (\frac{\lambda_0}{\lambda_1} V(t))^{\psi}]^{\frac{1}{\psi}}}, \tag{1.2}$$

where $V(t)$ is the total tumor volume; λ_0, λ_1, and ψ are constants.

Remark. *Cell growth in Z_1 is first exponential followed by a linear growth phase, described by λ_0 and λ_1, respectively.* [4] *The meaning of ψ was not mentioned in any reference material.*

The Simeoni model (1.1) is a transit compartment model (TCM) of perturbed tumor growth, which describes the delay in the effect of drug administration on tumor volume, as drug-induced apoptosis is a delayed process [2]. The central compartment, denoted as Z_1, represents the actively growing tumor that increases according to the tumor growth function (TGF) (1.2). The peripheral compartments Z_2, Z_3, and Z_4 represent states of cell death following anticancer treatment. The concentration of a chemotherapeutic or immunotherapeutic agent in Z_1 at time t is represented by $c(t)$. This

concentration triggers a portion of tumor cells to undergo cell death, with a corresponding killing constant k_1. These tumor cells enter successively into Z_2, Z_3, and Z_4 with a constant rate k_2. The $-k_2 * Z_4(t)$ term represents the volume of tumors that leave Z_4 at time t; in other words, those tumor cells die completely. The cumulative tumor volume observed results from the combined cell count in compartments Z_1, Z_2, Z_3, Z_4.

It is important to note that the number of peripheral compartments is arbitrary as these compartments represent stages of cell death due to the inherently delayed process of treatment. For example, there could be only one peripheral compartment Z_2, as seen in Figure 2.

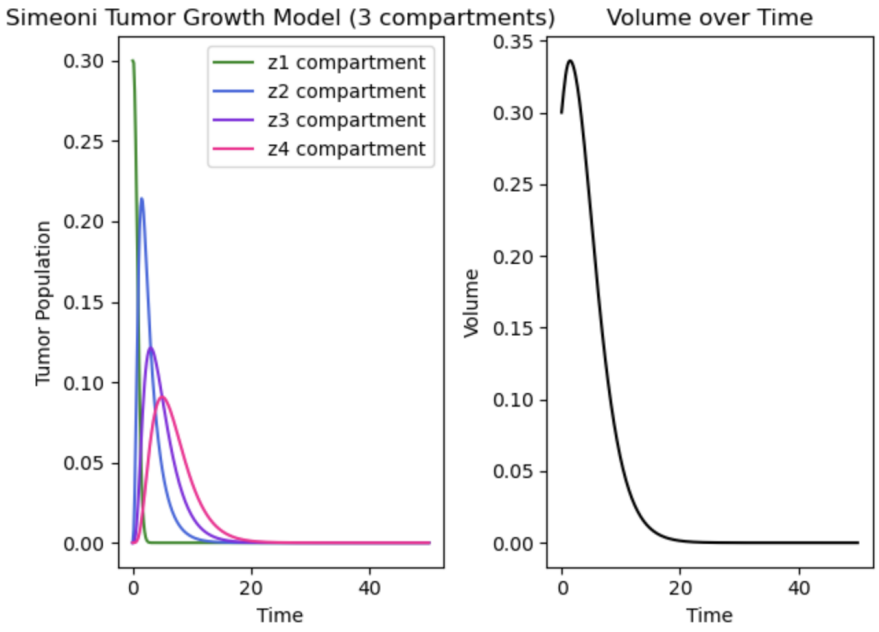


FIGURE 1. This graph shows our Simeoni graphs created with parameters: $\lambda_0 = .5, \lambda_1 = .2, \psi = .5, k_1 = .4, k_2 = .5, c = t, V_0 = .3$. These parameters were arbitrarily chosen, but show that tumors could be treated.

Various techniques and methods have been historically employed to tackle the challenge of modeling tumor growth and decay. Prior studies have explored models such as Mendelsohn, Exponential, Linear, Logistic, and Bertalanffy, among others [5]. Each of these models brings a unique perspective to the understanding of tumor dynamics. While Simeoni is one of the models in this repertoire, we recognize the importance of critically evaluating its suitability and effectiveness compared to alternative models.

The intricacies of the Simeoni model motivate a fundamental question: is the existing Simeoni model the best model for the decay and growth of tumor

cells, or are there adjustments that could enhance its precision in describing this phenomenon? A study conducted in 2020 showed that compared to other existing tumor growth functions, such as the generalized logistic, Gompertz, and Von Bertalanffny tumor growth functions, the Simeoni tumor growth function describes experimental data best, with minimal AIC, AICc, and BIC values[3] The weights that were found also underscore the preference for the Simeoni tumor growth function.

As such, we did not alter the Simeoni tumor growth function or the number of peripheral compartments. Instead, we decided to focus on the Simeoni model as it is. We first examined its equilibria and the stability of the system at each equilibrium. An initial look at the system of ODEs shows that there is an equilibrium when each $Z_i = 0$. This occurs at $t = 0$ an , so we have $Z_1(0) = Z_2(0) = Z_3(0) = Z_4(0)$. To confirm that this was the only equilibrium, we created a `numpy` array representing the system, and calculated the null space. This proved that for the original Simeoni model, there is one equilibrium at the origin. To examine the stability of the origin, we calculated the eigenvalues of the matrix representing the model, corresponding to each $Z_i = 0$. If we use $k_1 = 0.4$, $k_2 = 0.5$, $c(t) = 0.5$, and $TGF(t) \approx 0.075$, our eigenvalues are $\lambda_1 = -0.5$, $\lambda_2 = -0.5$, $\lambda_3 = -0.5$, and $\lambda_4 \approx -0.125$. Since these values are all less than zero, we found that the origin is a stable equilibrium.

Next, we examined the effects of variation of the model's parameters, such as k_1, k_2, λ_0, λ_1, psi, and V_0 on tumor population. We wanted to determine if by altering the parameters, we could significantly decrease the growth of tumor cells or even completely annihilate all tumor cells. We used `python` to define our system of ODEs (1.1) and `matplotlib` to visualize the graphs. We solved our IVP with Scipy `solve_ivp` and then plotted these solutions to observe each compartment's behavior. We explored various parameter combinations and found a set of parameter values that model the tumor growth most realistically and similarly to existing research [3]. See figure 2 for a visual representation of that model.

Remark. *The graph in Figure 2 shows that the overall tumor volume continues to increase, although the rate slightly changes during the plotted time frame, which can be attributed to the delayed effect of treatment. Tumor cells are indeed moving from Z_1 to Z_2 and some have continued to Z_3. Given more time, we expect to see tumors continuously moving to Z_3 and Z_4 and eventually dying off. Depending on the specific treatment values as well as the stage of cancer, we can expect the tumor volume to continue to increase at slower rates or even begin decreasing. In the latter case, the effect of the birth of new tumor cells according to the tumor growth function is suppressed by the killing constants.*

Allowing for $c(t)$, the concentration of the treatment agent in Z_1, to equal zero at all values of t gives us tumor growth in the absence of any treatment. This is modeled in Figure 3 to visualize untreated tumor growth over time.

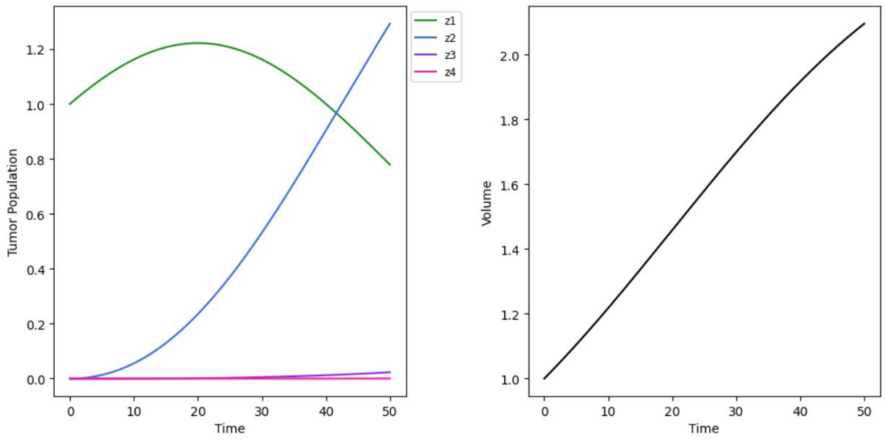


FIGURE 2. Realistic valued model, with parameters: $\lambda_0 = 0.02, \lambda_1 = 0.001, \psi = -2, c = t, k_1 = 0.001, k_2 = 0.001, V_0 = 1$. This model is realistic because it models treatment of the tumor, with decreased Z1 values.

Compare with Figure 2, which has the same given parameters. It is worth noting that we do observe what we would expect to observe. Under the same given parameters, there is higher tumor volume over time in the absence of treatment than there is under treatment.

Just as no model can claim absolute perfection in its descriptive capabilities, the Simeoni model has certain limitations as well. Tumor growth in real-life scenarios, like any other real data, is characterized by noise and uncertainty. There is always a degree of randomness that the differential equations of the Simeoni model do not capture. Moreover, the model's relevance and accuracy are examined through data from animal experimentation rather than human experimentation; this introduces another level of uncertainty. In addition, while the Simeoni tumor growth function captures experimental animal data best as compared with other growth functions, it is not the most biologically realistic growth law for application to humans [3]. With continuous research and comparison with other models and datasets, the Simeoni model and its tumor growth function can be further adapted or serve as a baseline for an eventual optimized model that can be realistically applied to humans. A 2020 study [3] found that the optimal number of peripheral compartments within the model remains uncertain concerning the goodness of fit. As the number of peripheral compartments is arbitrary yet also a crucial factor in modeling the results of treatment, potential weaknesses in this uncertainty should be acknowledged. As of this writing, there is no consensus as to which model best characterizes tumor growth. Despite its weaknesses, the Simeoni model remains one of the most robust tumor growth models in the field.

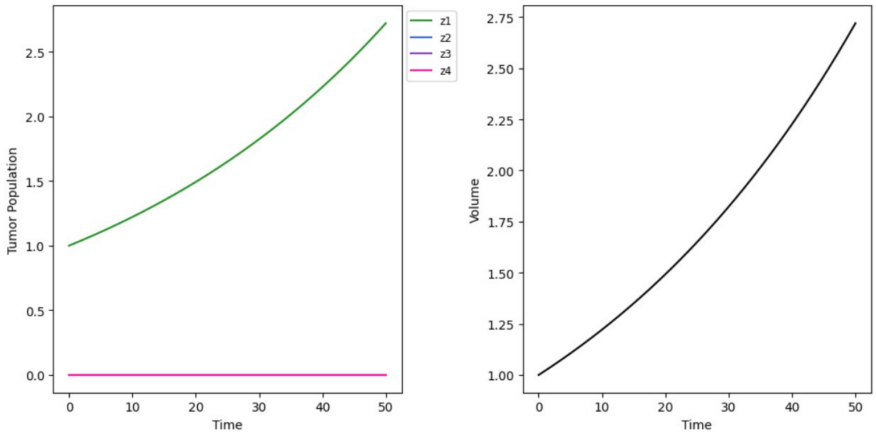


FIGURE 3. Realistic valued model, with no treatment

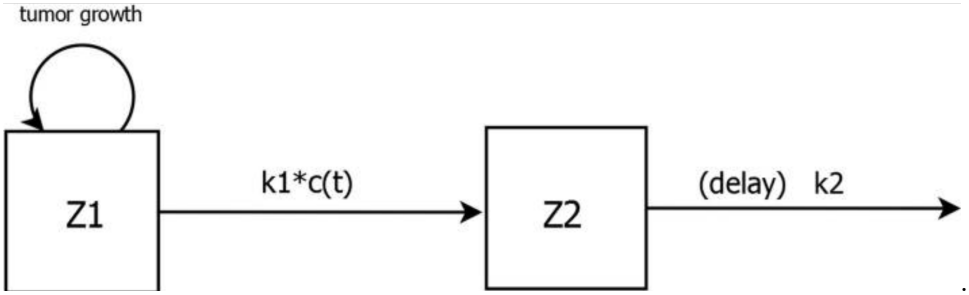


FIGURE 4. This figure is from "Different ODE models of tumor growth can deliver similar results"[3]. It shows the tumor growth happening in Z1 and the delayed growth from tumor treatment. The fact that there are four compartments in the model (Z_1, Z_2, Z_3, and Z_4) is arbitrary, and models different stages of . Our model fits this figure's depiction of the stages of tumor growth.

Figure 1 and Figure 2 illustrate the rapid decrease of tumor volume upon increasing the values of the constant parameters. With specific parameter values as outlined in the respective captions, we find that we can not only slow tumor growth rate but also completely eradicate all tumors. As seen in Figure 1, all tumors are essentially gone within 20 days.

3. RESULTS

Concerning the model's stability, we found that it is based on the existence of one stable equilibrium at the origin. This equilibrium satisfies the

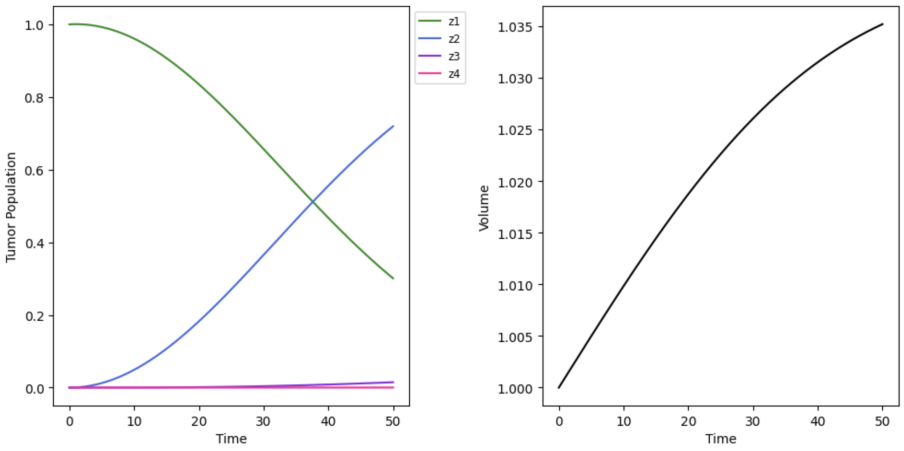


FIGURE 5. Our altered-parameters-valued model with parameters $\lambda_0 = 0.02, \lambda_1 = 0.001, \psi = 2, k_1 = 0.001, k_2 = 0.001, V_0 = 1$

definition of a uniformly stable system and is crucial for maintaining the system's stability and consistent behavior.

After conducting deep parameter tuning, we can affirm that the Simeoni model can realistically reflect the chemotherapeutic and immunotherapeutic treatment on tumor growth. In the absence of such treatment agents, tumor growth is increased.

We also found that when each constant parameter is increased relative to the constant parameter setting of the realistic model (see Figure 2), tumor growth increases at a slower rate. With parameter values high enough, the overall tumor volume even eventually returns to zero, signifying a state void of any remaining tumor cells. This potentially has very significant implications: if the constant parameters can be realistically and ethically manipulated to attain the values as in our altered model, the efficacy of the same levels of chemotherapeutic and immunotherapeutic agents can be greatly increased. Nonetheless, these findings suggest that even minimally increasing the values of the constant parameters can have positive effects on slowing tumor growth.

Figure 1 and Figure 2 illustrate the rapid decrease of tumor volume upon increasing the values of the constant parameters. With specific parameter values as outlined in the respective captions, we find that we can not only slow tumor growth rate but also completely eradicate all tumors. As seen in Figure 1, all tumors are essentially gone within 20 days.

Again, it is important to explore the feasibility of directly manipulating these parameters in real-world scenarios to determine the optimal, realistic combination for achieving the most effective cancer treatment.

4. Analysis/Conclusions

The techniques and methods employed in our modeling endeavors were crucial in gaining insights into the dynamics of cancer progression. We chose to model treated tumors within the body, which we feel the Simeoni model does well. Through our analysis, we observed both successes and challenges in capturing the intricacies of the chosen phenomenon. The appropriateness of the methods became evident as they allowed us to simulate and understand the growth patterns of cancerous tumors.

However, we acknowledge that the modeling process presented its own set of complexities, and while our group made significant strides, there remains room for improvement. Given more time, we would like to do more research on tumors to be more informed and to be able to modify the model by adding new variables and functions as well as find a better explanation of the types of constants that model tumor growth appropriately. In addition, we would like to explore and dissect the Simeoni model further by altering the c(t) function. Findings from these particular alterations can lend insight into optimal amounts of chemotherapeutic and immunotherapeutic agent concentrations at different times. Experimenting with different initial values of Z1 can lend valuable insight into the efficacy of treatment at earlier vs later and slower vs rapid onset stages of cancer.

In the event of a successful modification of the Simeoni Model, our findings could have far-reaching implications for cancer treatment strategies. The ability to predict and understand tumor growth patterns more accurately could inform personalized treatment plans and optimize the allocation of resources in healthcare. Our research emphasizes the significance of dynamic modeling in the context of cancer research and the potential impact on treatment efficiency and patient outcomes.

Our goal was to educate ourselves and any potential audience of an example of a system of ODEs that appropriately models how a tumor grows and depletes as it is treated. Though models are never perfect, we believe we can learn more about tumors and how to properly treat them by observing what different parameters are most effective in stopping tumor growth. The lessons learned from our research methodology can also be applied to other areas of scientific inquiry, providing a framework for tackling complex biological processes. We encourage others to appreciate the interdisciplinary nature of modeling and its potential to drive advancements in healthcare and beyond.

References

[1] How cancers grow. *Cancer Research UK*, October 2023.
[2] Jong Hyuk Byun, In-Soo Yoon, Song Yi Lee, Hyun-Jong Cho, and Il Hyo Jung. Extended transit compartment model to describe tumor delay using coxian distribution. *Scientific Reports*, 12(1):10086, 2022.
[3] James A Koziol, Theresa J Falls, and Jan E Schnitzer. Different ode models of tumor growth can deliver similar results. *BMC cancer*, Mar 2020.
[4] Emma Martin, Leon Aarons, and James Yates. Accounting for dropout in xenografted tumour efficacy studies: integrated endpoint analysis, reduced bias and better use of animals. *Cancer Chemotherapy and Pharmacology*, 78, 07 2016.
[5] Hope Murphy, Hana Jaafari, and Hana M. Dobrovolny. Differences in predictions of ode models of tumor growth: a cautionary example. *BMC cancer*, Feb 2016.

Simeoni Model Jupyter Notebook

December 7, 2023

1 Adjusted SIR model lab

[1]:
```
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import odeint, solve_ivp, solve_bvp
from scipy.optimize import minimize
import numpy.linalg as la
```

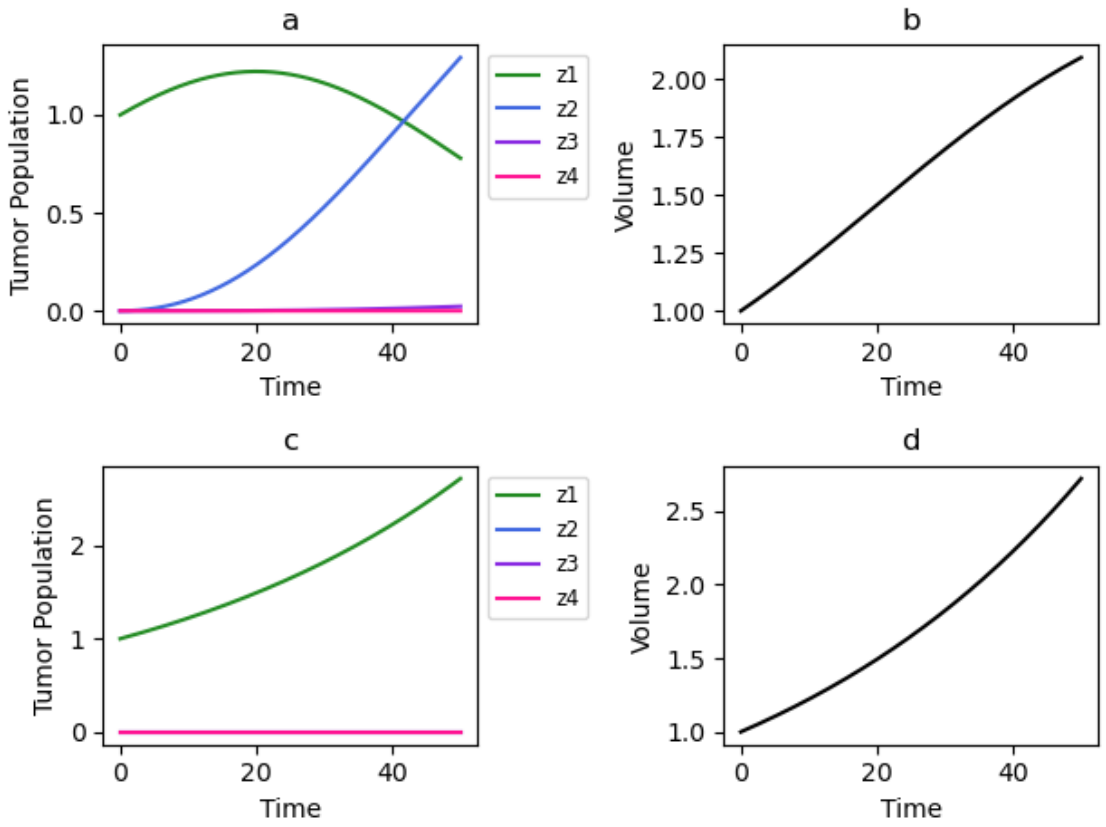
[7]:
```
def simeoni_3(lambda0=.5, lambda1=.2, psi=.5, k1 = .4, k2=.5, c=lambda t: t*k1,␣
↪V0 = 3, show_volume=True, show_growth=True, show_base=True):
    '''
    Adjusting the the SIR model, model the spread cancer in a body
    it plots the results of cancer cells over time

    Parameters:
    Tumor growth function values:
        lambda0 - default .5
        lambda1 - default .2
        psi - default .5

    c(t) denotes the concentration of a chemotherapeutic or immunotherapeutic␣
↪agent
    V0 - default = 3
    '''

    volume = []
    no_chemo_volume = []

    # the simeoni model - ode system
    def ode(t, Y):

        Z1 = Y[0]
        Z2 = Y[1]
        Z3 = Y[2]
        Z4 = Y[3]

        V_t = Z1 + Z2 + Z3 + Z4
```

```
        volume.append(V_t)

        def TGF(t):
            return ((lambda0 * Y[0]) /(1+(lambda0 * V_t /lambda1)**psi)**(1/
↪psi))

        dZ1 = TGF(t) - k1*c(t)*Z1
        dZ2 = k1*c(t)*Z1 - k2*Z2
        dZ3 = k2*Z2 - k2*Z3
        dZ4 = k2*Z3 -k2*Z4

        return np.array([dZ1, dZ2, dZ3, dZ4])#, V_t

    # Time domain
    t0 = 0
    tf = 50
    t = np.linspace(t0,tf,300)

    # total cells and inital conditions
    #initial cancer level = Y0
    Y0 = np.array([V0,0,0,0])


    # solve it
    sol = solve_ivp(ode, (t0,tf), Y0, t_eval=t).y

    fig, axs = plt.subplots(2, 2)

    if show_growth:
      # create colors for the graph
      plot_colors = ['forestgreen', 'royalblue', 'blueviolet', 'deeppink']

      line1, = axs[0, 0].plot(t,sol[0], label='z1 compartment',␣
↪c=plot_colors[0])
      line2, = axs[0, 0].plot(t,sol[1], label='z2 compartment',␣
↪c=plot_colors[1])
      line3, = axs[0, 0].plot(t,sol[2], label='z3 compartment',␣
↪c=plot_colors[2])
      line4, = axs[0, 0].plot(t,sol[3], label='z4 compartment',␣
↪c=plot_colors[3])

      axs[0, 0].set_title('a')

      axs[0, 0].set_xlabel('Time')
      axs[0, 0].set_ylabel('Tumor Population')
```

```
    legend = axs[0, 0].legend(handles=[line1, line2, line3, line4],␣
↪labels=['z1', 'z2', 'z3', 'z4'], loc='upper left', bbox_to_anchor=(1, 1),␣
↪fontsize='small')

    print('a = Simeoni Tumor Growth Model (3 periphery compartments)')

  if show_volume:
    axs[0, 1].plot(t, sol[0] + sol[1] + sol[2] + sol[3], c='k')
    axs[0, 1].set_title('b')
    axs[0, 0].set_xlabel('Time')
    axs[0, 0].set_ylabel('Tumor Population')
    axs[0, 1].set_xlabel("Time")
    axs[0, 1].set_ylabel("Volume")

    print('b = Volume over Time')

  # no chemo
  def base_ode(t, Y):

    Z1 = Y[0]
    Z2 = Y[1]
    Z3 = Y[2]
    Z4 = Y[3]

    V_t = Z1 + Z2 + Z3 + Z4
    no_chemo_volume.append(V_t)

    def TGF(t):
        return ((lambda0 * Y[0]) /(1+(lambda0 * V_t /lambda1)**psi)**(1/psi))

    dZ1 = TGF(t)
    dZ2 = -k2*Z2
    dZ3 = k2*Z2 - k2*Z3
    dZ4 = k2*Z3 -k2*Z4

    return np.array([dZ1, dZ2, dZ3, dZ4])#, V_t

  base_sol = solve_ivp(base_ode, (t0,tf), Y0, t_eval=t).y

  if show_base: # no chemo
    bline1, = axs[1, 0].plot(t,base_sol[0], label='z1 compartment',␣
↪c=plot_colors[0])
    bline2, = axs[1, 0].plot(t,base_sol[1], label='z2 compartment',␣
↪c=plot_colors[1])
    bline3, = axs[1, 0].plot(t,base_sol[2], label='z3 compartment',␣
↪c=plot_colors[2])
```

```
    bline4, = axs[1, 0].plot(t,base_sol[3], label='z4 compartment',␣
↪c=plot_colors[3])
    axs[1, 0].set_title('c')
    axs[1, 0].set_xlabel('Time')
    axs[1, 0].set_ylabel('Tumor Population')

    legend1 = axs[1, 0].legend(handles=[bline1, bline2, bline3, bline4],␣
↪labels=['z1', 'z2', 'z3', 'z4'], loc='upper left', bbox_to_anchor=(1, 1),␣
↪fontsize='small')


    axs[1, 1].plot(t, base_sol[0] + base_sol[1] + base_sol[2] + base_sol[3],␣
↪c='k')
    axs[1, 1].set_title('d')
    axs[1, 1].set_xlabel('Time')
    axs[1, 1].set_ylabel('Volume')

    print('c = Tumor Growth Model w/o Treatment (3 periphery compartments)')
    print('d = Volume over Time w/o Treatment')

  print('Parameters: lambda0=' + str(lambda0) + ', lambda1=' + str(lambda1) +␣
↪', psi=' + str(psi) +
        ', k1=' + str(k1) + ', k2=' + str(k2) + ', V0=1')
  plt.tight_layout()
  plt.show()

# chosen constants
lambda0 = .5
lambda1=.2
psi = .5
k1 = .4
k2=.5
V0 = 3

def c(t):
    """induces a fraction of tumor cells to commit
        to cell death with a killing constant k1"""
    return t


simeoni_3(lambda0=.02, lambda1=.001, psi=-2, k1=.001, k2=.001, c=c, V0 = 1)
```

```
a = Simeoni Tumor Growth Model (3 periphery compartments)
b = Volume over Time
c = Tumor Growth Model w/o Treatment (3 periphery compartments)
d = Volume over Time w/o Treatment
Parameters: lambda0=0.02, lambda1=0.001, psi=-2, k1=0.001, k2=0.001, V0=1
```



Then after creating the simeoni_3 function, we tested with random parameters to try to find the best model. We did that with the following code, which we commented out after our analysis.

[9]:
```
#plot a bunch and see what happens
#all constants are the same and are less than 1
# for i in range(1, 10):
    # print("consts =", i/10)
    # simeoni_3(lambda0=i/10, lambda1=i/10, psi=i/10, k1 = i/10, k2=i/10, V0 =␣
 ↪100)
```

[10]:
```
#plot a bunch and see what happens
#all constants are the same and are less than 1
# for i in range(1, 10):
#     lambda0 = np.random.rand()
#     lambda1 = np.random.rand()
#     psi = np.random.rand()
#     k1 = np.random.rand()
#     k2 = np.random.rand()
#     print(lambda0, lambda1, psi, k1, k2)
#     simeoni_3(lambda0=lambda0, lambda1=lambda1, psi=psi, k1 =k1, k2=k2, V0 =␣
 ↪100)
```

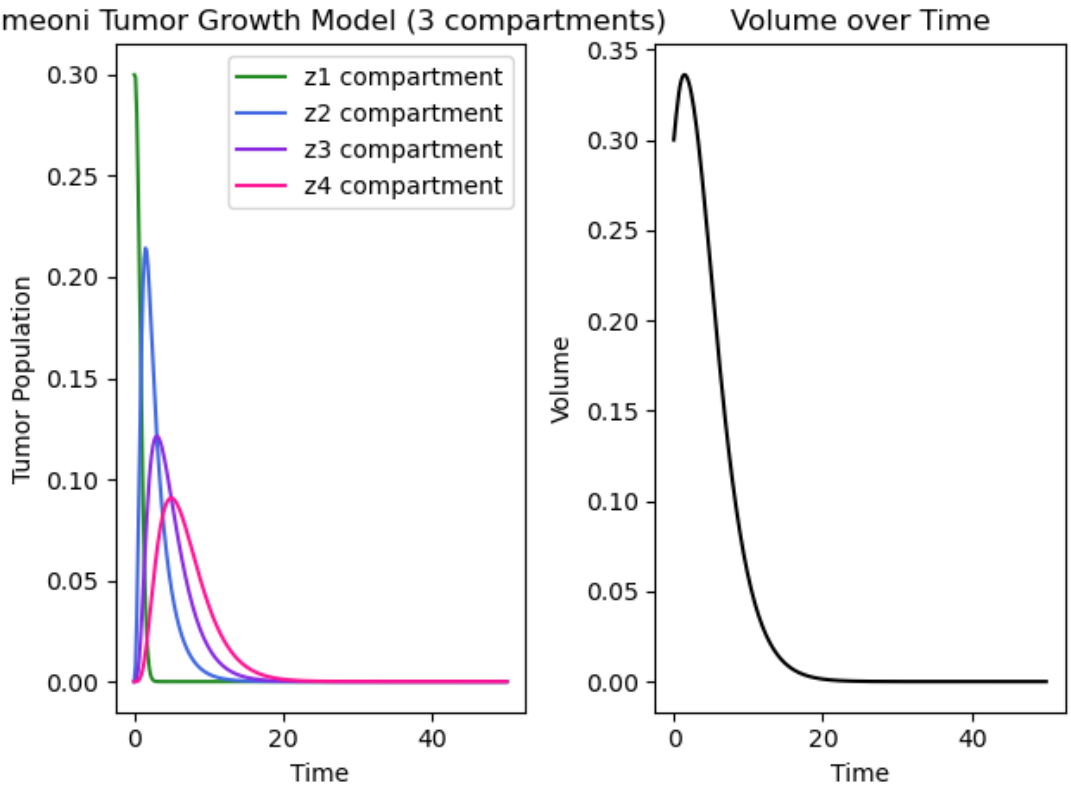
After looking at each graph, we picked the best parameters and slightly modified the simeoni_3() function below:

[12]:
```
# Our modified model
# we have lambdas and phis changing above
# we can change c(t) = concentration in Z1 compartment at time t
# we can also show effects of chemo from being introduced when initial Z1 is very low  vs very high
# done -> also do a base non-chemo tumor growth model
def simeoni_modified(lambda0=.5, lambda1=.2, psi=.5, k1 = .4, k2=.5, c=lambda t: t*k1, V0 = 3, show_volume=True, show_growth=True):
    '''
    Adjusting the Simeoni model, model the spread cancer in a body
    it plots the results of cancer cells over time

    Parameters:
    Tumor growth function values:
        lambda0 - default .5
        lambda1 - default .2
        psi - default .5

    c(t) denotes the concentration of a chemotherapeutic or immunotherapeutic agent
    V0 - default = 3
    '''

    volume = []

    #simeoni model - ode system
    def ode(t, Y):

        Z1 = Y[0]
        Z2 = Y[1]
        Z3 = Y[2]
        Z4 = Y[3]

        V_t = Z1 + Z2 + Z3 + Z4
        volume.append(V_t)

        def TGF(t):
            return ((lambda0 * Y[0]) /(1+(lambda0 * V_t /lambda1)**psi)**(1/psi))

        dZ1 = TGF(t) - k1*c(t)*Z1
        dZ2 = k1*c(t)*Z1 - k2*Z2
        dZ3 = k2*Z2 - k2*Z3
        dZ4 = k2*Z3 -k2*Z4
```

```
        return np.array([dZ1, dZ2, dZ3, dZ4])#, V_t

    # Time domain
    t0 = 0
    tf = 50
    t = np.linspace(t0,tf,300)

    # total cells and inital conditions
    #initial cancer level = Y0
    Y0 = np.array([V0,0,0,0])


    # solve it
    sol = solve_ivp(ode, (t0,tf), Y0, t_eval=t).y

    if show_growth:
      # create colors for the graph
      plot_colors = ['forestgreen', 'royalblue', 'blueviolet', 'deeppink']

      plt.subplot(1, 2, 1)
      plt.plot(t,sol[0], label='z1 compartment', c=plot_colors[0])
      plt.plot(t,sol[1], label='z2 compartment', c=plot_colors[1])
      plt.plot(t,sol[2], label='z3 compartment', c=plot_colors[2])
      plt.plot(t,sol[3], label='z4 compartment', c=plot_colors[3])

      plt.xlabel('Time')
      plt.ylabel('Tumor Population')
      plt.title("Simeoni Tumor Growth Model (3 compartments)")
      plt.legend()
      plt.tight_layout()

    if show_volume:
      plt.subplot(1, 2, 2)
      plt.plot(t, sol[0] + sol[1] + sol[2] + sol[3], c='k')
      plt.title('Volume over Time')
      plt.xlabel('Time')
      plt.ylabel('Volume')
      plt.tight_layout()
      plt.show()

# good constants
lambda0 = .5
lambda1=.2
psi = .5
k1 = .4
k2=.5
```

```
V0 = 3

def c(t):
    """induces a fraction of tumor cells to commit
        to cell death with a killing constant k1"""
    return 5*t

simeoni_modified(lambda0=.5, lambda1=.2, psi=.5, k1 =.4, k2=.5, c=c, V0 =.3)
```

```
/var/folders/6f/tvgwmzxn47d92btq5ry1q_b40000gn/T/ipykernel_3239/642102557.py:34:
RuntimeWarning: invalid value encountered in double_scalars
  return ((lambda0 * Y[0]) /(1+(lambda0 * V_t /lambda1)**psi)**(1/psi))
```



As time goes on the higher the consts the less cancer there is. We need to look at the different constants though to figure out what affects it the most.

Stability

[13]:
```
# to find the equilibria, we want to find the Z1-Z4 where each dZ = 0
lambda0=.5
lambda1=.2
psi=.5
k1 = .4
```

```
k2=.5
c=.5
V0 = 3

# define our Z matrix
# each row represents dZi, and each column represents Zi
Z_matrix = np.array([[(lambda0)/(1+(lambda0/lambda1)**psi)**(1/psi) - k1*c, 0,␣
 ↪0, 0],
                     [k1*c, -k2, 0, 0],
                     [0, k2, -k2, 0],
                     [0, 0, k2, -k2]])

# now solve for zeros
b = np.zeros(4)
Z_vals = la.solve(Z_matrix, b)
print(Z_vals)

print((lambda0)/(1+(lambda0/lambda1)**psi)**(1/psi))

# now solve for the eigenvalues of Z_matrix
eigenvals = np.linalg.eigvals(Z_matrix)
print(eigenvals)
```

```
[ 0.  0.  0. -0.]
0.07504940885147124
[-0.5        -0.5        -0.5        -0.12495059]
```

Now we know for sure that the only equilibrium is when each Z is equal to 0