UTM UNIVERSITI TEKNOLOGI MALAYSIA SCHOOL OF COMPUTING Faculty of Engineering

ASSIGNMENT 1

SECB4313 BIOINFORMATICS MODELING AND SIMULATION

SECTION 01

LECTURER:

DR. AZURAH BINTI A SAMAH

GROUP MEMBERS:

No.	Name	Matric No
1	CHANG MIN XUAN	A20EC0024
2	LEE JIA YEE	A20EC0063

TABLE OF CONTENT

INTRODUCTION

A simulation model is a computational construct that represents a real-world system or process. It is designed to imitate the behaviors and interactions of real-world systems using a model, enabling researchers, engineers and designers to understand, predict and analyze the system's performance under various conditions (What is simulation? what does it mean? (definition and examples)). Besides, the creation of a simulation model requires a conceptual physical model, in terms of being described with mathematical representation that generates numerical outcomes through experiments (Li et al., 2020). By further analyzing and interpreting the outcomes obtained, conclusions with justification can be drawn to know the system's behavior, learning the system design and improvement to be done. Meanwhile, it assesses how varying uncertain factors and the choices we implement affect the results that are significant to us. Essentially, it's a tool that helps predict the consequences of uncertainties and decisions on things we value. Although building a simulation model is considered time-consuming and expensive, it is still particularly valuable for solving real-world problems safely and efficiently (Gibson, 2002). They are used extensively across disciplines, from engineering to environmental science, to optimize processes, enhance safety, and drive innovation. Crucially, the simulation model is an indispensable tool that provides invaluable insights and decision-making impacts, driving advancement in various fields.

OBJECTIVE

The objectives of simulation model are:

- To evaluate how well the system performs under various scenarios and conditions
- To analyze the potential risks associated with changes in the system without affecting actual operations
- To demonstrate theoretical concepts and allow for the exploration of 'what-if' scenarios in a controlled environment

FLOW OF SIMULATION MODEL

Based on the codes, the simulation model begins with initialization of necessary python libraries such as Flask, NumPy, Matplotlib, and SciPy are the libraries being imported. Next, the mathematical model describing the system dynamics is defined in the function. The function measures how quickly or slowly these variables change over time, providing insights into the dynamics of the system's cellular components. Moreover, a homepage route is defined using @app.route(). This route handles both GET and POST requests. Default model parameters are specified. If the form is submitted via POST request, the parameters are updated accordingly. Subsequently, the model undergoes ODE Integration and Plotting. The differential equations constituting the model are solved through the 'odeint' function provided by SciPy, utilizing the newly adjusted parameters. The results obtained are graphically represented via Matplotlib, and the ensuing chart is preserved as a static image file within the directory 'static/plot.png'. Later, the homepage template ('index.html') is rendered, passing the current parameter values and the path to the graph image. Another route for the results page ('/results') is defined, which simply renders a template ('results.html') that displays the graph. Finally, the Flask application is run, with debug mode enabled.

MATHEMATICAL EQUATIONS

Equation 1:

$$dCdt = rC * C * (1 — (T/K)) * (1 — S) — dC * C$$

Description:

dCdt represents the change in the concentration of cancer cells over time and is dependent on the **rate of growth of cancer cells** (rC), the **competition for resources with other cells** (1-(T/K)), the **effect of treatment** (1-S), and the **death rate of cancer cells** (dC).

Equation 2 :

$$dHdt = rH * H$$

Description:

dHdt represents the changes in the concentration of healthy cells over time and is dependent on the **growth rate of healthy cells** (rH).

Equation 3:

$$dILdt = kIL * H$$

Description:

dILdt represents the changes in the concentration of interleukins over time and is dependent on the **production rate of interleukins** (kIL).

Equation 4:

$$dTdt = -kCT * C * T$$

Description:

dTdt represents the change in the concentration of tumor cells over time and is dependent on the competition for resources with other cells, the **death rate of tumor cells due to treatment** (kCT), and the **concentration of tumor cells themselves** (T).

Equation 5:

$$dSdt = s * T$$

Description:

dSdt represents the change in the effect of treatment over time and is dependent on the **concentration of tumor cells** (T) and the **effectiveness of the treatment** (s).

PYTHON LIBRARIES

The python libraries used are:

- Flask: Used for creating the online application.
- render_template: Used to render HTML templates within Flask in a web browser.
- request: Used to handle HTTP requests within Flask.
- numpy: Used for numerical computations, particularly arrays and matrices.
- matplotlib.pyplot: Used for creating plots and visualizations.
- scipy.integrate.odeint: Used for solving ordinary differential equations (ODEs) numerically.

INPUT OF SIMULATION MODEL

The input for the simulation model is shown below:

C = the concentration of cancer cells

H = the concentration of healthy cells

IL = the concentration of interleukins

T = the concentration of tumor cells

S = the effect of treatment

MODEL PARAMETERS

rC = the rate of growth of cancer cells

dC = the death rate of cancer cells

rH = the growth rate of healthy cells

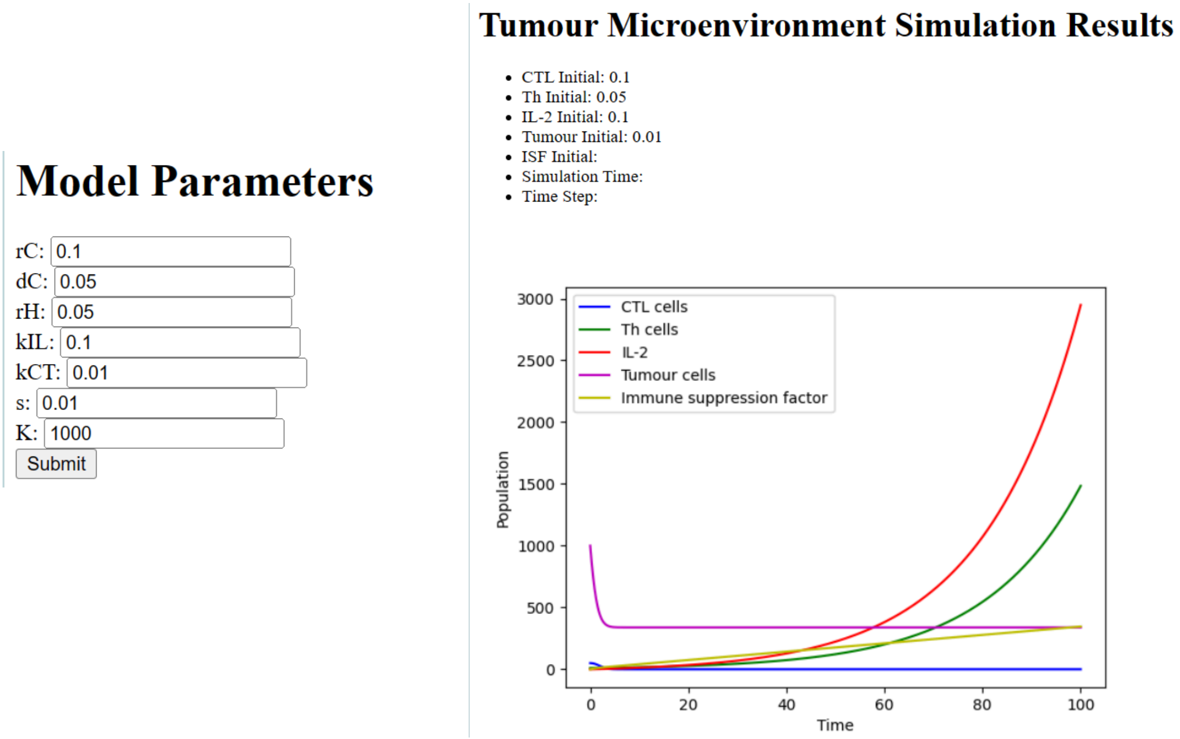
kIL = the production rate of interleukins

kCT = the death rate of tumor cells due to treatment

s = the effectiveness of the treatment

K = the capacity of the environment

DESCRIPTION OF SIMULATION OUTPUT & GENERATED GRAPH



Model Parameters

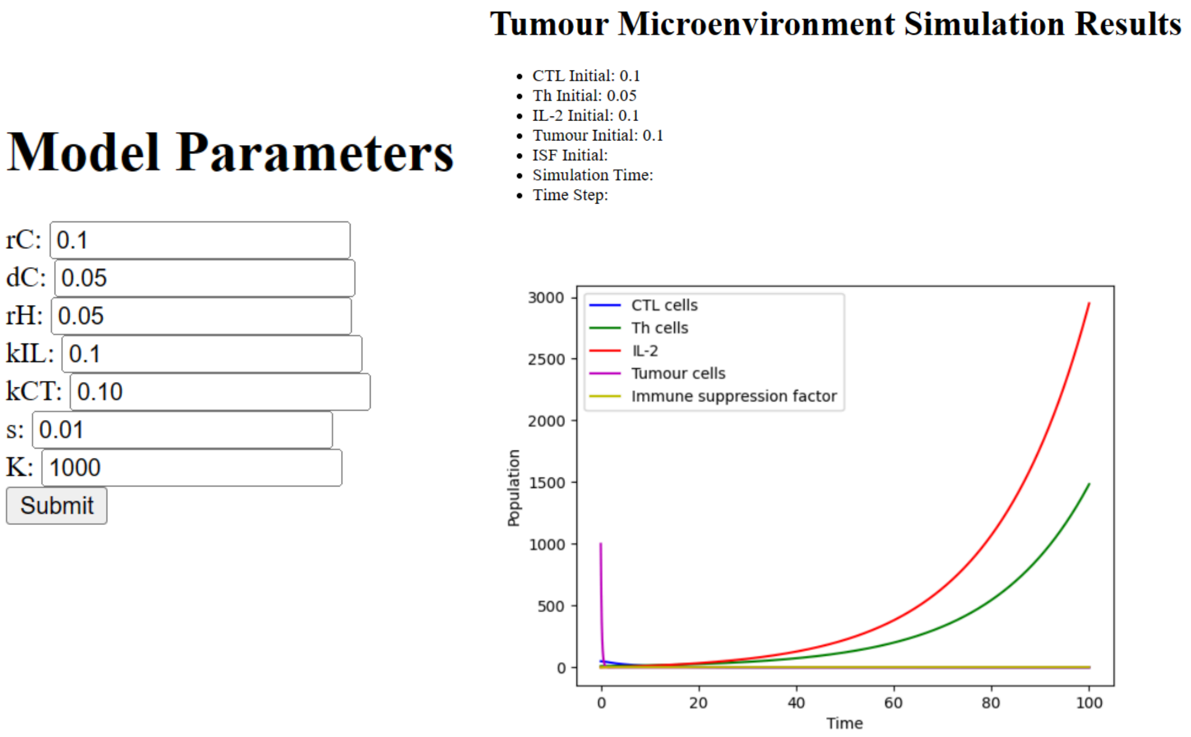
rC: 0.1
dC: 0.05
rH: 0.05
kIL: 0.1
kCT: 0.01
s: 0.01
K: 1000

Tumour Microenvironment Simulation Results

- CTL Initial: 0.1
- Th Initial: 0.05
- IL-2 Initial: 0.1
- Tumour Initial: 0.01
- ISF Initial:
- Simulation Time:
- Time Step:

Figure 1

Based on the Figure 1, it can be seen that the the model parameters are set in default when users didn’t edit any parameters’ value. After submitting the default value, the generated graph shown that the population of tumour cells decrease sharply from 1000 to 400 when time approaches 2 and then, it remains constantly. For the cancer cells, CTL, it can be observed the blue line drop slightly to 0 in population and stayed constantly from 3 to 100 in time. Conversely, healthy cells, interleukins and immune supperession factor are in increasing trend. When time is 100, interleukins reached the highest population which is 3000 and the healthy cells recorded with 1500 in population.



Model Parameters

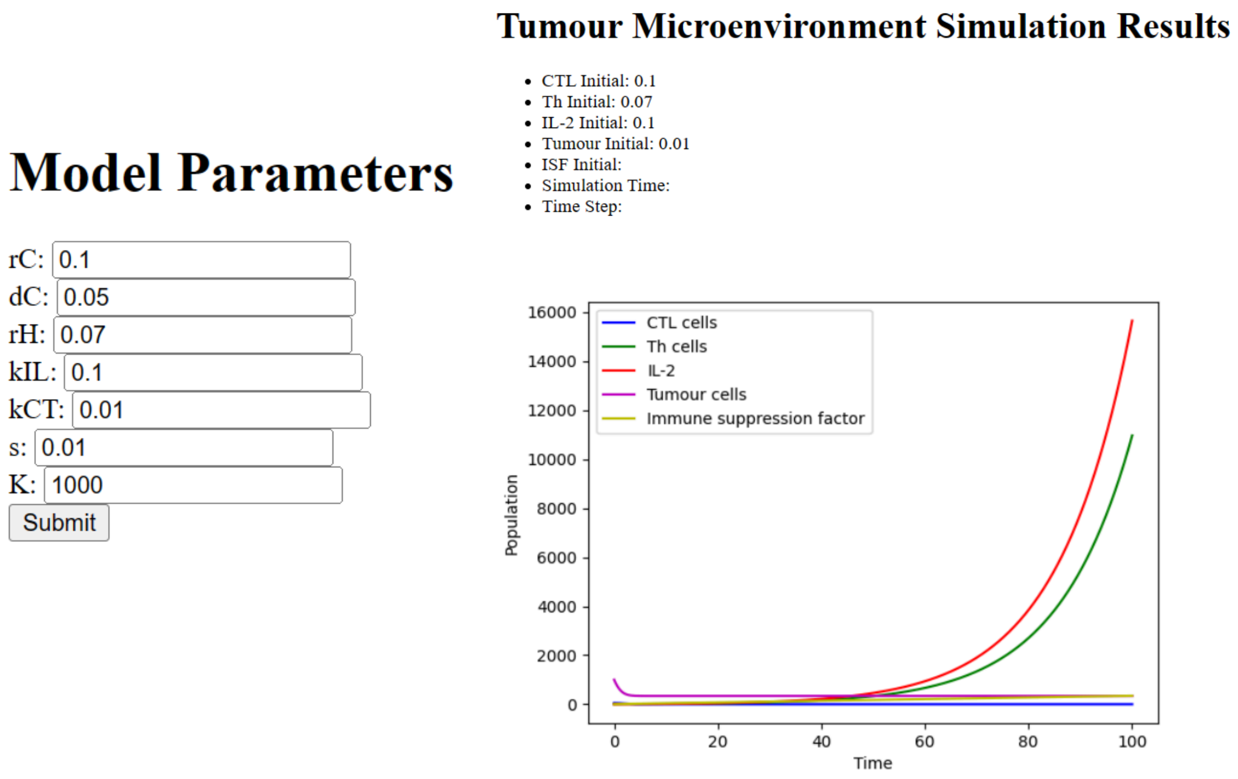
rC: 0.1
dC: 0.05
rH: 0.05
kIL: 0.1
kCT: 0.10
s: 0.01
K: 1000

Tumour Microenvironment Simulation Results

- CTL Initial: 0.1
- Th Initial: 0.05
- IL-2 Initial: 0.1
- Tumour Initial: 0.1
- ISF Initial:
- Simulation Time:
- Time Step:

Figure 2

According to Figure 2, the kCT parameter which indicates the death rate of tumor cells due to treatment is changed to 0.10. The result graph demonstrated the population of tumor cells is slumped to 0 when time is 1 and remained same until 100 in time. For the population of interleukins and heathy cells, they also in an upward trend as same as the result from figure 1. In comparison, the immune suppression factor's population displayed as 0 throughout the simulation time while the population of cancer cells, CTL decreased mildly from 50 to 0 when time is 5 and remain constantly to the end of simulation. The above graph outcome is due to the reason that when the death rate of tumor cells due to treatment was set in 0.1 which considered as higher rate compare to 0.01. When the tumor cells' death rate is high, the population of cancer cells was definitely less and the immune suppression factor recorded as 0 was affected by the population of tumor cells which responsible to produce immune suppression response to degrade the cancer cells.


Tumour Microenvironment Simulation Results

- CTL Initial: 0.1
- Th Initial: 0.07
- IL-2 Initial: 0.1
- Tumour Initial: 0.01
- ISF Initial:
- Simulation Time:
- Time Step:

Model Parameters

rC: 0.1
dC: 0.05
rH: 0.07
kIL: 0.1
kCT: 0.01
s: 0.01
K: 1000

Figure 3

By referring to Figure 3, the rH parameter was modified from 0.05 to 0.07 which means that the growth rate of heathy cells increased. The plotted graph illustrated the population of heathy cells and interleukins rose significantly to nearly 16000 and 11000 respectively. Meanwhile the lines represented the population of tumour cells, immune suppression factor and cancer cells remain same throughout simulation time compared to figure 1 but only the population indicators became larger and caused the changes of these 3 lines less distinct.

EXPERIMENTATION THAT CAN BE CARRIED OUT USING SIMULATION MODEL

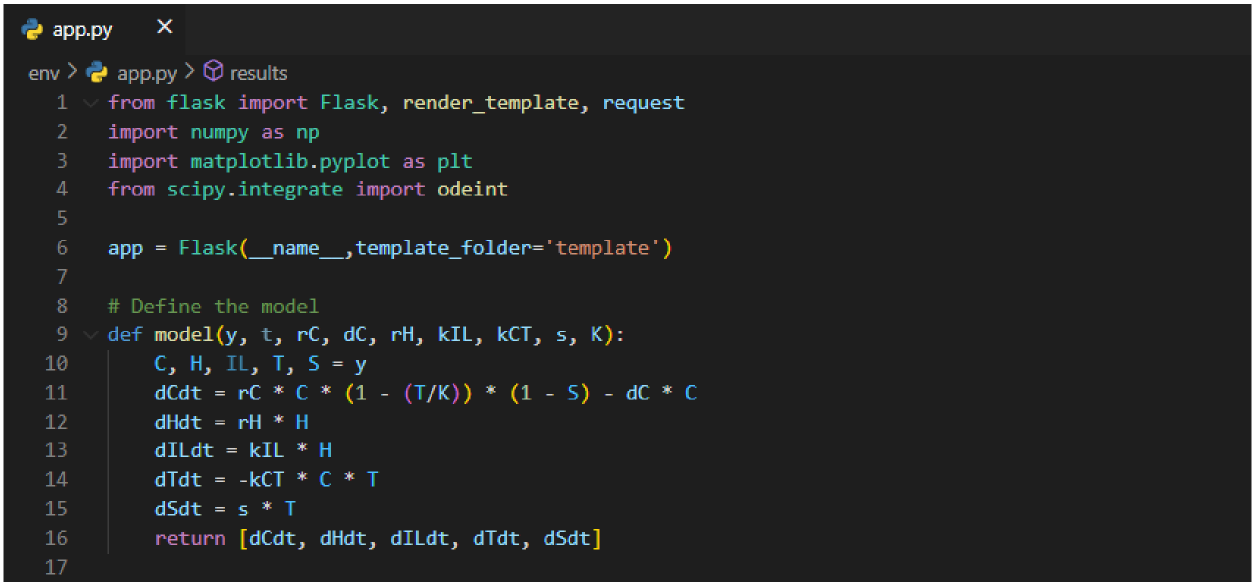
1. Disease Classification
2. Disease Prediction
3. Disease Identification
4. Disease Model Optimization
5. Drug Discovery and Development

REFERENCES

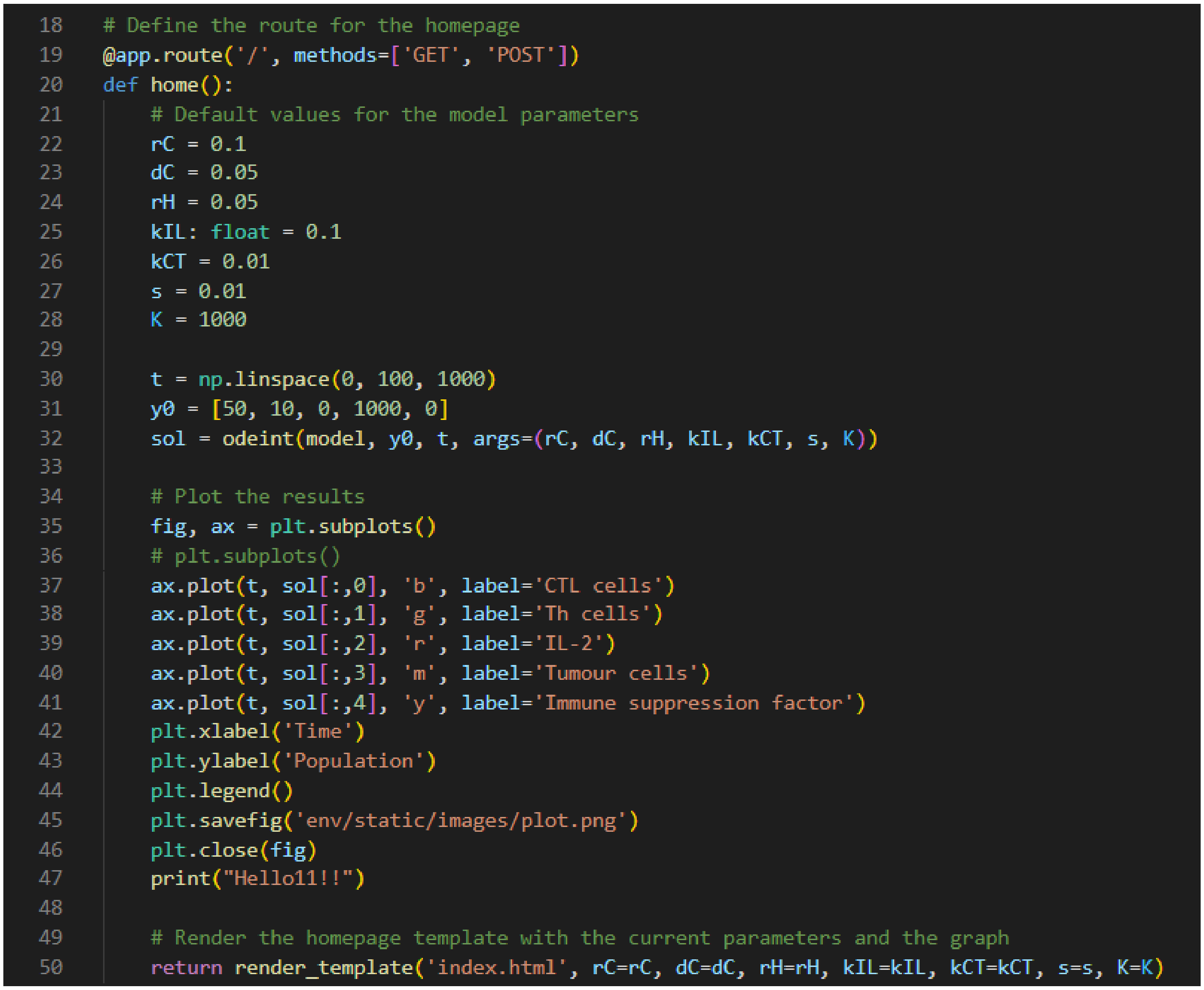
1. What is simulation? what does it mean? (definition and examples) (no date) TWI. Available at: https://www.twi-global.com/technical-knowledge/faqs/faq-what-is-simulation (Accessed: 23 April 2024).
2. Li, F. et al. (2020) ‘Review of real-time simulation of Power Electronics’, Journal of Modern Power Systems and Clean Energy, 8(4), pp. 796–808. doi:10.35833/mpce.2018.000560.
3. Gibson, R.R. (2002) ‘Planning and plant layout’, Plant Engineer’s Reference Book [Preprint]. doi:10.1016/b978-075064452-5/50060-2.

APPENDIX

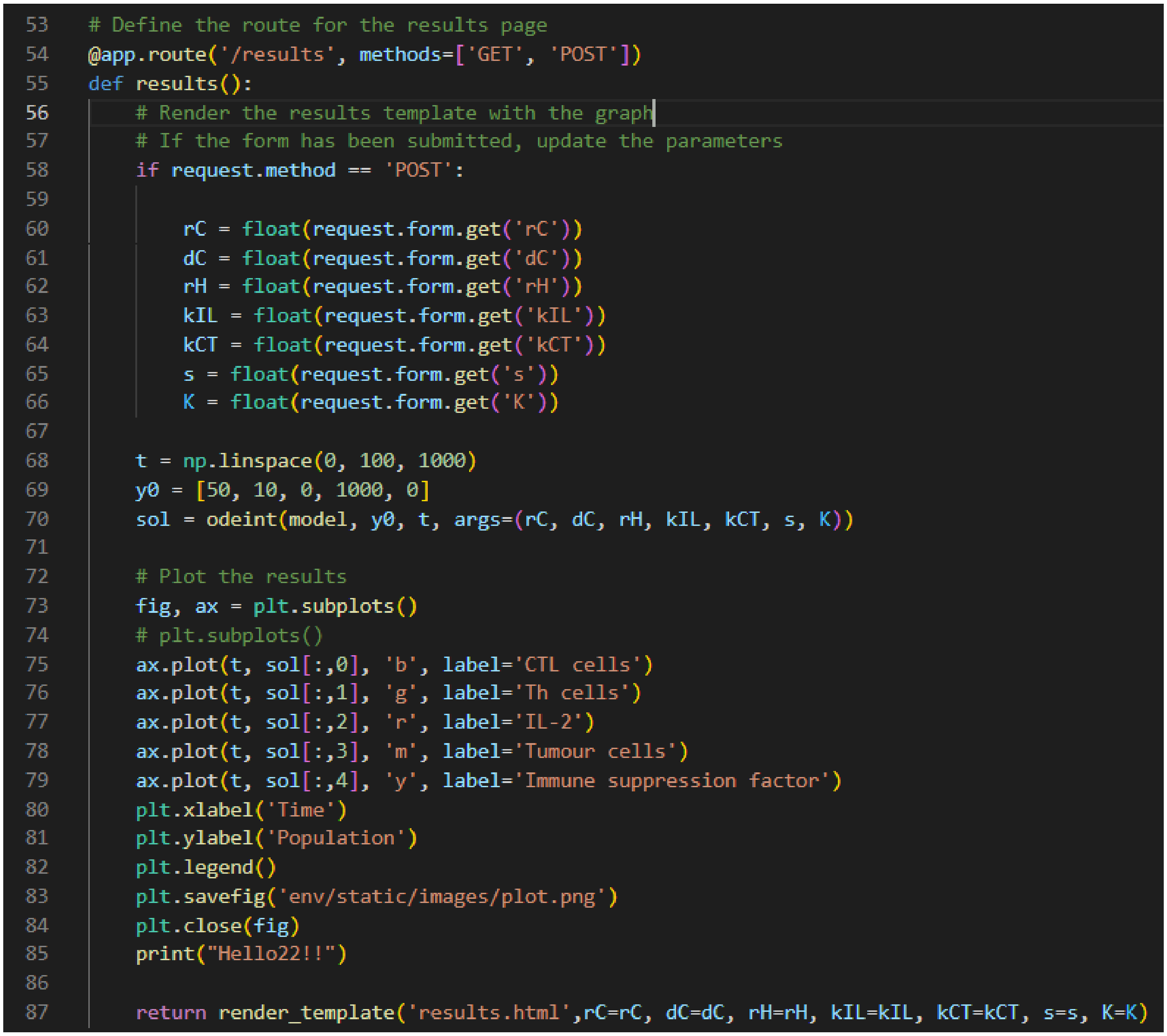
app.py:



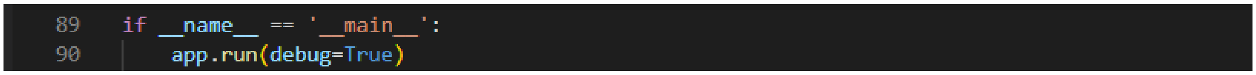
```python
from flask import Flask, render_template, request
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import odeint

app = Flask(__name__,template_folder='template')

# Define the model
def model(y, t, rC, dC, rH, kIL, kCT, s, K):
    C, H, IL, T, S = y
    dCdt = rC * C * (1 - (T/K)) * (1 - S) - dC * C
    dHdt = rH * H
    dILdt = kIL * H
    dTdt = -kCT * C * T
    dSdt = s * T
    return [dCdt, dHdt, dILdt, dTdt, dSdt]

```



```python
# Define the route for the homepage
@app.route('/', methods=['GET', 'POST'])
def home():
    # Default values for the model parameters
    rC = 0.1
    dC = 0.05
    rH = 0.05
    kIL: float = 0.1
    kCT = 0.01
    s = 0.01
    K = 1000

    t = np.linspace(0, 100, 1000)
    y0 = [50, 10, 0, 1000, 0]
    sol = odeint(model, y0, t, args=(rC, dC, rH, kIL, kCT, s, K))

    # Plot the results
    fig, ax = plt.subplots()
    # plt.subplots()
    ax.plot(t, sol[:,0], 'b', label='CTL cells')
    ax.plot(t, sol[:,1], 'g', label='Th cells')
    ax.plot(t, sol[:,2], 'r', label='IL-2')
    ax.plot(t, sol[:,3], 'm', label='Tumour cells')
    ax.plot(t, sol[:,4], 'y', label='Immune suppression factor')
    plt.xlabel('Time')
    plt.ylabel('Population')
    plt.legend()
    plt.savefig('env/static/images/plot.png')
    plt.close(fig)
    print("Hello11!!")

    # Render the homepage template with the current parameters and the graph
    return render_template('index.html', rC=rC, dC=dC, rH=rH, kIL=kIL, kCT=kCT, s=s, K=K)
```

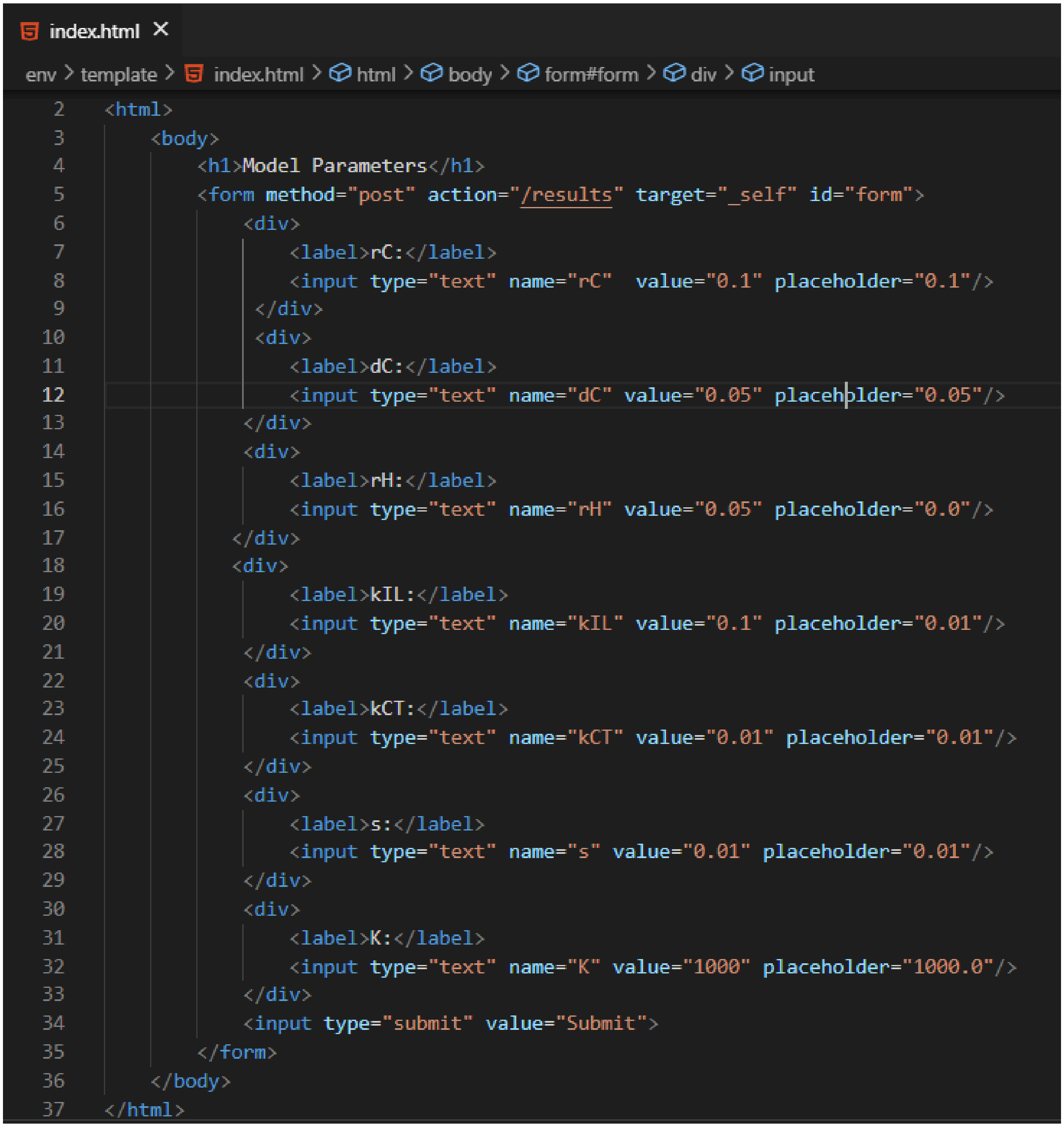


```python
# Define the route for the results page
@app.route('/results', methods=['GET', 'POST'])
def results():
    # Render the results template with the graph
    # If the form has been submitted, update the parameters
    if request.method == 'POST':

        rC = float(request.form.get('rC'))
        dC = float(request.form.get('dC'))
        rH = float(request.form.get('rH'))
        kIL = float(request.form.get('kIL'))
        kCT = float(request.form.get('kCT'))
        s = float(request.form.get('s'))
        K = float(request.form.get('K'))

    t = np.linspace(0, 100, 1000)
    y0 = [50, 10, 0, 1000, 0]
    sol = odeint(model, y0, t, args=(rC, dC, rH, kIL, kCT, s, K))

    # Plot the results
    fig, ax = plt.subplots()
    # plt.subplots()
    ax.plot(t, sol[:,0], 'b', label='CTL cells')
    ax.plot(t, sol[:,1], 'g', label='Th cells')
    ax.plot(t, sol[:,2], 'r', label='IL-2')
    ax.plot(t, sol[:,3], 'm', label='Tumour cells')
    ax.plot(t, sol[:,4], 'y', label='Immune suppression factor')
    plt.xlabel('Time')
    plt.ylabel('Population')
    plt.legend()
    plt.savefig('env/static/images/plot.png')
    plt.close(fig)
    print("Hello22!!")

    return render_template('results.html',rC=rC, dC=dC, rH=rH, kIL=kIL, kCT=kCT, s=s, K=K)
```



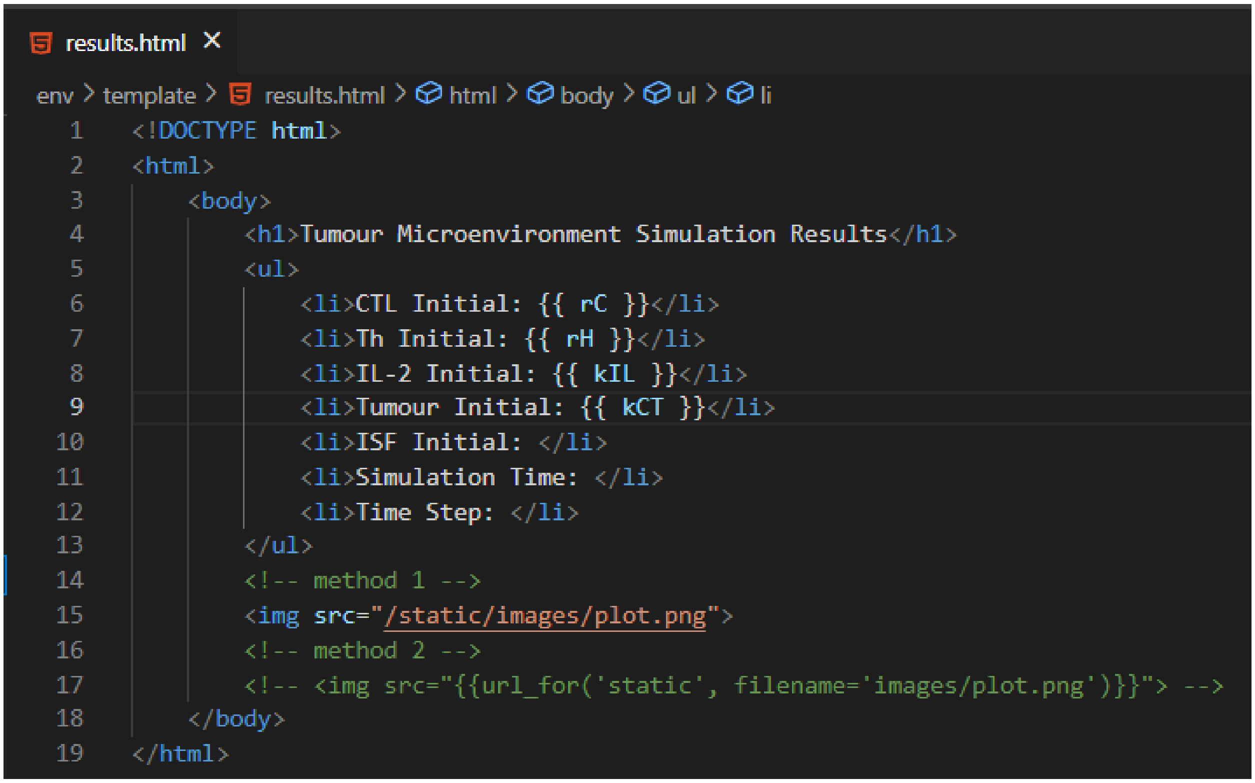
```python
if __name__ == '__main__':
    app.run(debug=True)
```

index.html:


index.html ×
env > template > index.html > html > body > form#form > div > input

```html
<html>
    <body>
        <h1>Model Parameters</h1>
        <form method="post" action="/results" target="_self" id="form">
            <div>
                <label>rC:</label>
                <input type="text" name="rC"  value="0.1" placeholder="0.1"/>
             </div>
             <div>
                <label>dC:</label>
                <input type="text" name="dC" value="0.05" placeholder="0.05"/>
            </div>
            <div>
                <label>rH:</label>
                <input type="text" name="rH" value="0.05" placeholder="0.0"/>
           </div>
           <div>
                <label>kIL:</label>
                <input type="text" name="kIL" value="0.1" placeholder="0.01"/>
            </div>
            <div>
                <label>kCT:</label>
                <input type="text" name="kCT" value="0.01" placeholder="0.01"/>
            </div>
            <div>
                <label>s:</label>
                <input type="text" name="s" value="0.01" placeholder="0.01"/>
            </div>
            <div>
                <label>K:</label>
                <input type="text" name="K" value="1000" placeholder="1000.0"/>
            </div>
            <input type="submit" value="Submit">
        </form>
    </body>
</html>
```

results.html:



```html
<!DOCTYPE html>
<html>
    <body>
        <h1>Tumour Microenvironment Simulation Results</h1>
        <ul>
            <li>CTL Initial: {{ rC }}</li>
            <li>Th Initial: {{ rH }}</li>
            <li>IL-2 Initial: {{ kIL }}</li>
            <li>Tumour Initial: {{ kCT }}</li>
            <li>ISF Initial: </li>
            <li>Simulation Time: </li>
            <li>Time Step: </li>
        </ul>
        <!-- method 1 -->
        <img src="/static/images/plot.png">
        <!-- method 2 -->
        <!-- <img src="{{url_for('static', filename='images/plot.png')}}"> -->
    </body>
</html>
```