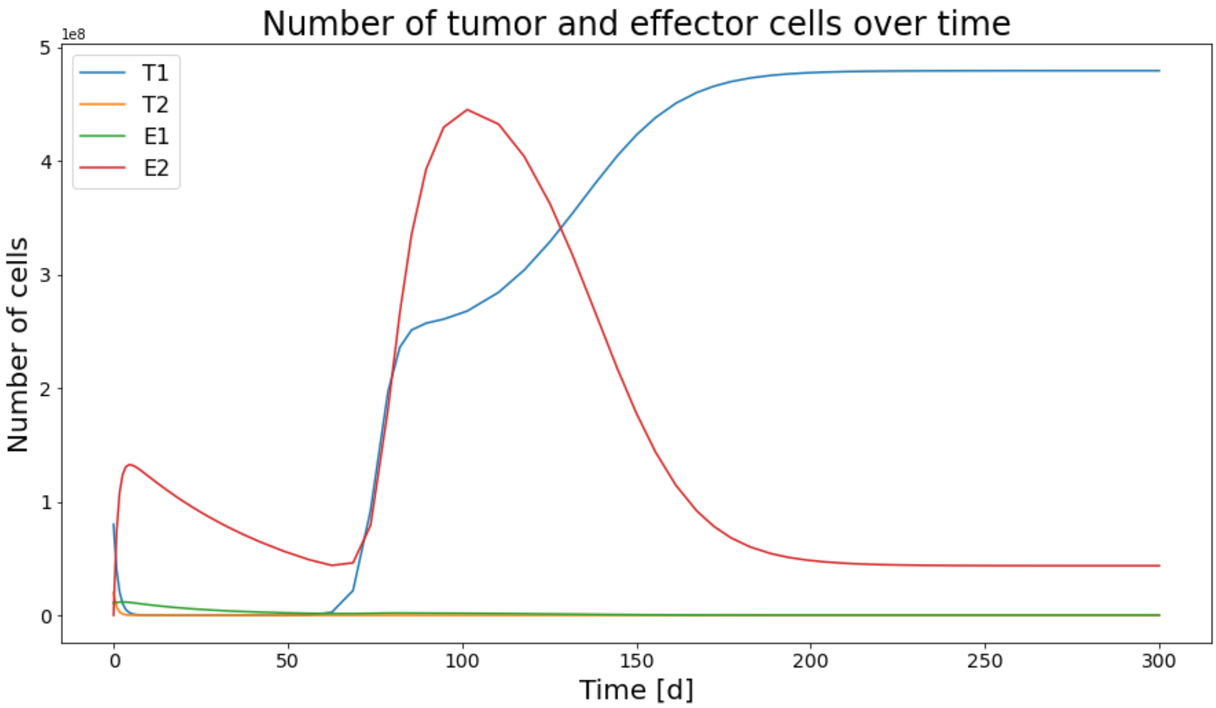
Imports

In [7]:
```
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import solve_ivp
from SALib.sample import saltelli
from SALib.analyze import sobol
import numpy as np
```

Create ODE system and plot the number of cells over time for T_1, T_2, E_1, and E_2.

In [5]:
```
def ode_system(t, state, g1,g2,a11,a12,a21,p1,d1,d2,e1,e2,r1,r2,r3,s1,s2,i12,i21,K1,
    T1,T2,E1,E2 = state

    dT1dt = g1*T1*(1-T1/K1) - a11*E1*T1 - a12*E2*T1 - i12*T1*T2
    dT2dt = g2*T2*(1-T2/K2) - a21*E1*T2 - i21*T1*T2
    dE1dt = p1 - d1*E1 - e1*(T1+T2)*E1 + ((r1*(T1+T2))/(s1+T1+T2))*E1
    dE2dt = -d2*E2 - e2*T1*E2 + ((r2*T1)/(s2+T1))*E2 + r3*E1*(T1+T2)
    return [dT1dt, dT2dt, dE1dt, dE2dt]

g1 = 0.514
g2 = 0.35 * g1
a11 = 1.1e-7
a12 = 1.1e-10
a21 = a11
p1 = 1.3e4
d1 = 4.12e-2
d2 = 2.0e-2
e1 = 3.42e-10
e2 = e1
r1 = 1.24e-1
r2 = 1.24e-3
r3 = 1.1e-7
s1 = 2.02e7
s2 = s1
i12 = 1.1e-9
i21 = 1.5*i12
K1 = 5e8
K2 = K1

p = (g1,g2,a11,a12,a21,p1,d1,d2,e1,e2,r1,r2,r3,s1,s2,i12,i21,K1,K2)
initial = [8.0e7,2.0e7,1.1e7,0]

t_span = (0.0, 300.0)
t = np.arange(0.0, 300.0, 0.01)

result_solve_ivp = solve_ivp(ode_system, t_span, initial, args=p)

labels = ['T1', 'T2', 'E1', 'E2']
fig = plt.figure(figsize=(15,8))
for i in range(result_solve_ivp.y.shape[0]):
    plt.plot(result_solve_ivp.t, result_solve_ivp.y[i], label=labels[i])

plt.xlabel('Time [d]', fontsize=20) # the horizontal axis represents the time
plt.ylabel('Number of cells', fontsize=20)
plt.xticks(size=14)
```

```
plt.yticks(size=14)
plt.title('Number of tumor and effector cells over time', fontsize = 24)
plt.legend(fontsize=16) # show how the colors correspond to the components of X
fig.savefig('Output_ODE_test.jpg')
plt.show()
```

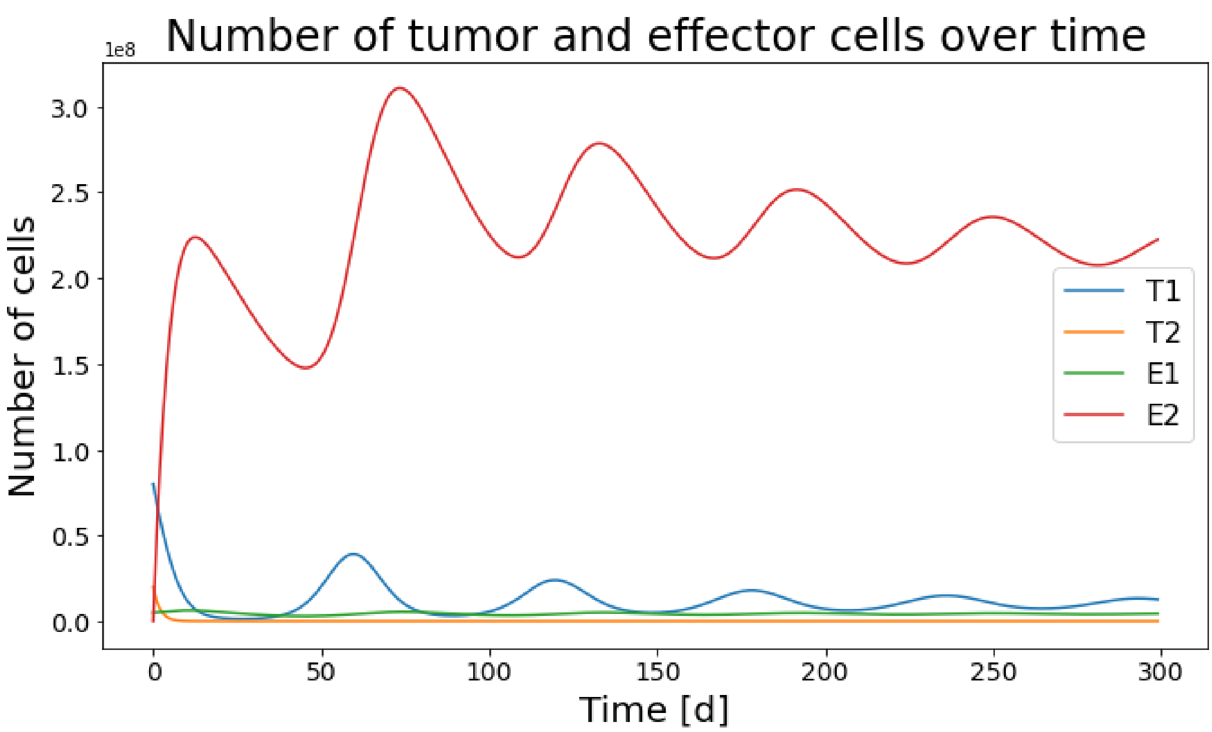


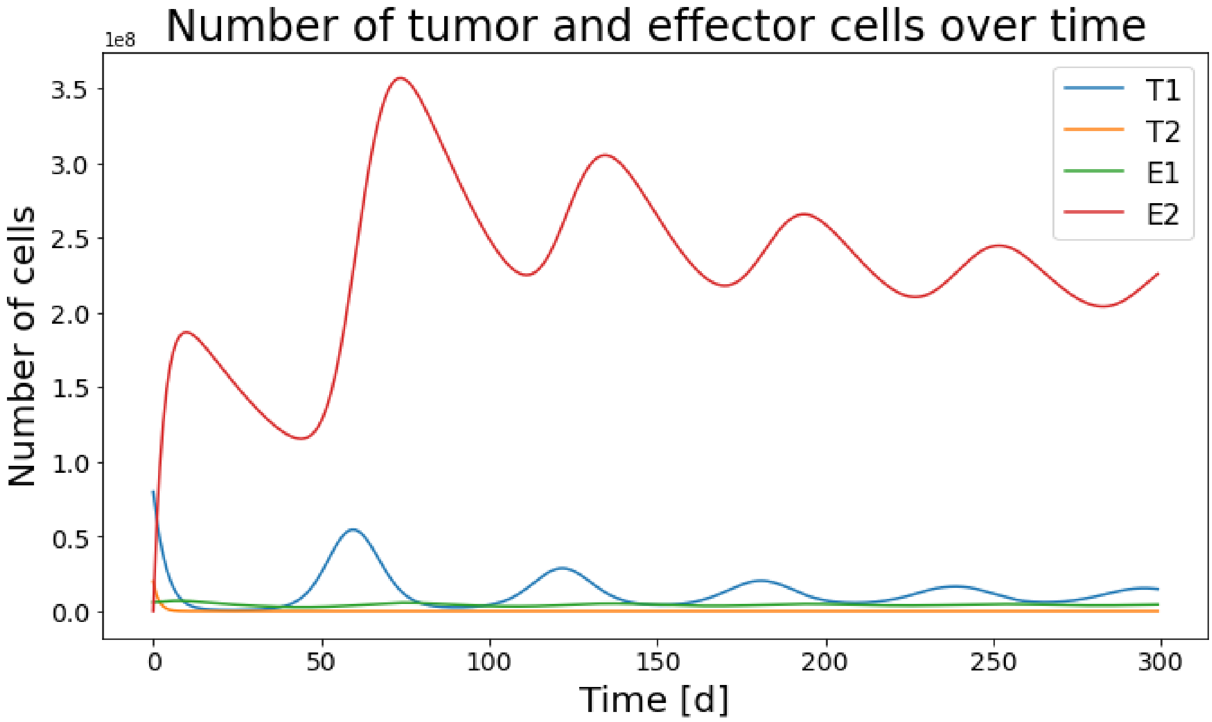
Make the same plots for different E_1 initial values and extract T_1 values for the time t = 250.

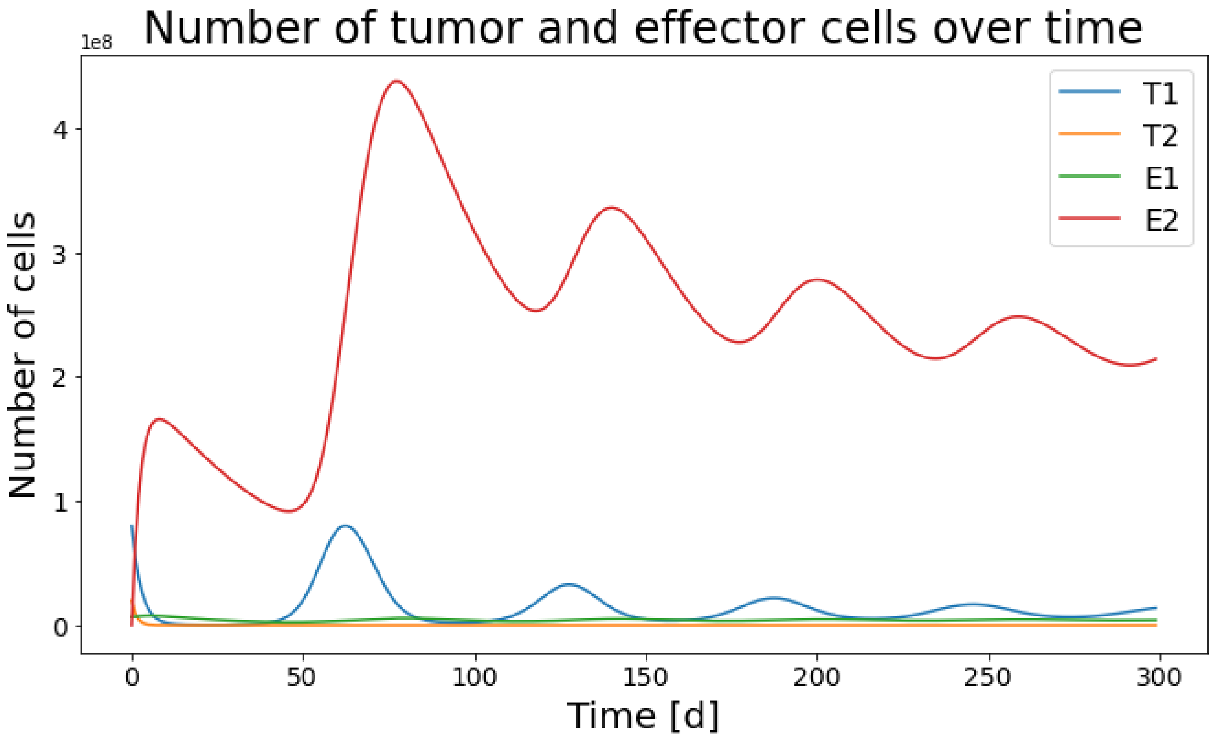
In [48]:

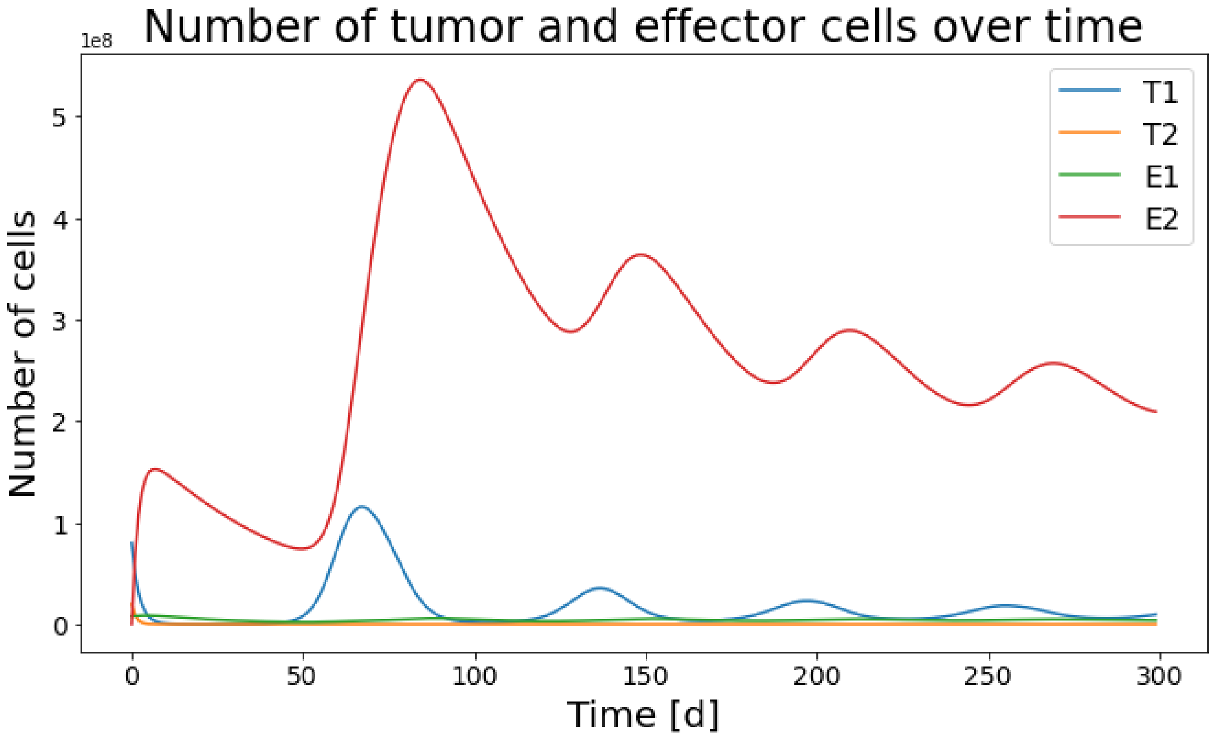
```
def ode_system(t, state, g1,g2,a11,a12,a21,p1,d1,d2,e1,e2,r1,r2,r3,s1,s2,i12,i21,K1,
    T1,T2,E1,E2 = state

    dT1dt = g1*T1*(1-T1/K1) - a11*E1*T1 - a12*E2*T1 - i12*T1*T2
    dT2dt = g2*T2*(1-T2/K2) - a21*E1*T2 - i21*T1*T2
    dE1dt = p1 - d1*E1 - e1*(T1+T2)*E1 + ((r1*(T1+T2))/(s1+T1+T2))*E1
    dE2dt = -d2*E2 - e2*T1*E2 + ((r2*T1)/(s2+T1))*E2 + r3*E1*(T1+T2)
    return [dT1dt, dT2dt, dE1dt, dE2dt]

g1 = 0.514
g2 = 0.35 * g1
a11 = 1.1e-7
a12 = 1.1e-10
a21 = a11
p1 = 1.3e4
d1 = 4.12e-2
d2 = 2.0e-2
e1 = 3.42e-10
e2 = e1
r1 = 1.24e-1
r2 = 1.24e-3
r3 = 1.1e-7
s1 = 2.02e7
s2 = s1
i12 = 1.1e-9
i21 = 1.5*i12
```

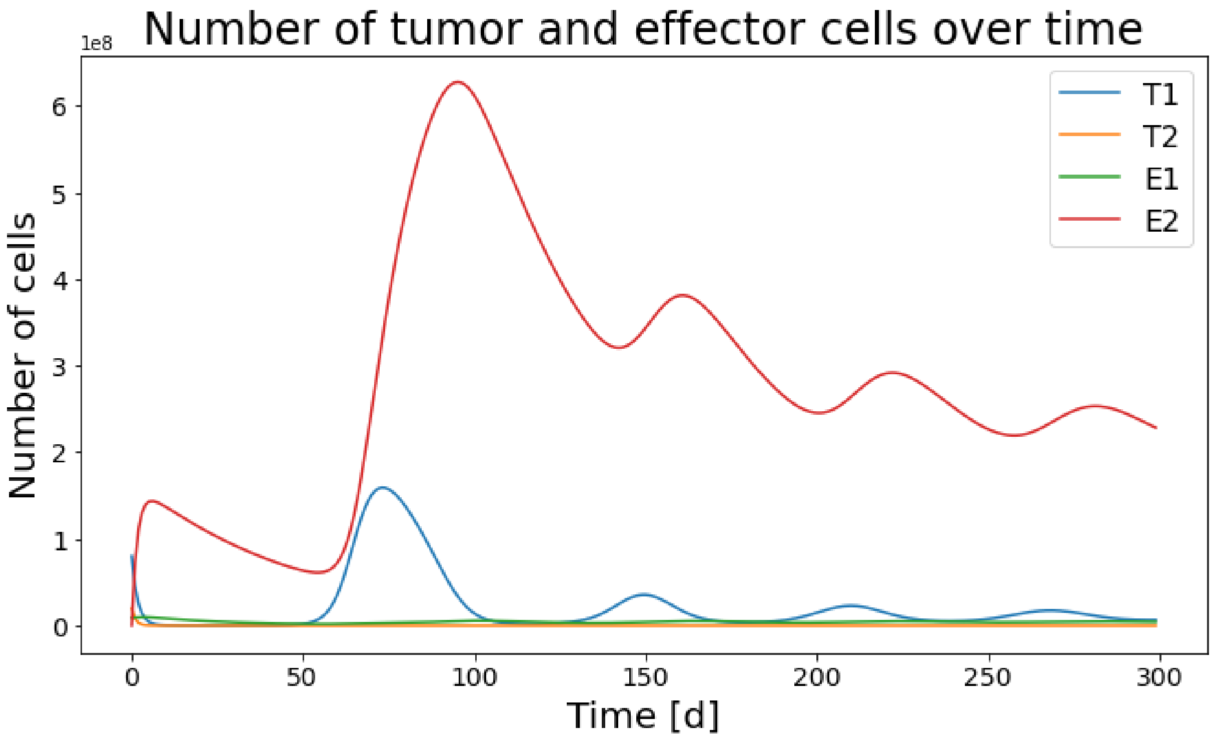
```
K1 = 5e8
K2 = K1

values_at_t250 = list()

p = (g1,g2,a11,a12,a21,p1,d1,d2,e1,e2,r1,r2,r3,s1,s2,i12,i21,K1,K2)
E1_list = [0.5e7, 0.6e7, 0.7e7, 0.8e7, 0.9e7, 1e7, 1.1e7, 1.2e7, 1.3e7, 1.4e7, 1.5e7
for i in range(11):
    E1 = E1_list[i]
    initial = [8.0e7,2.0e7,E1,0]

    t_span = (0, 300.0)
    t = np.arange(0, 300.0, 1)

    result_solve_ivp = solve_ivp(ode_system, t_span, initial, args=p, t_eval=t)

    labels = ['T1', 'T2', 'E1', 'E2']
    fig = plt.figure(figsize=(11,6))
    for i in range(result_solve_ivp.y.shape[0]):
        plt.plot(result_solve_ivp.t, result_solve_ivp.y[i], label=labels[i])
        values_at_t250.append(result_solve_ivp.y[i][249])

    plt.xlabel('Time [d]', fontsize=20) # the horizontal axis represents the time
    plt.ylabel('Number of cells', fontsize=20)
    plt.xticks(size=14)
    plt.yticks(size=14)
    plt.title('Number of tumor and effector cells over time', fontsize = 24)
    plt.legend(fontsize=16) # show how the colors correspond to the components of X
    fig.savefig('Output_ODE_test.jpg')
    plt.show()
```

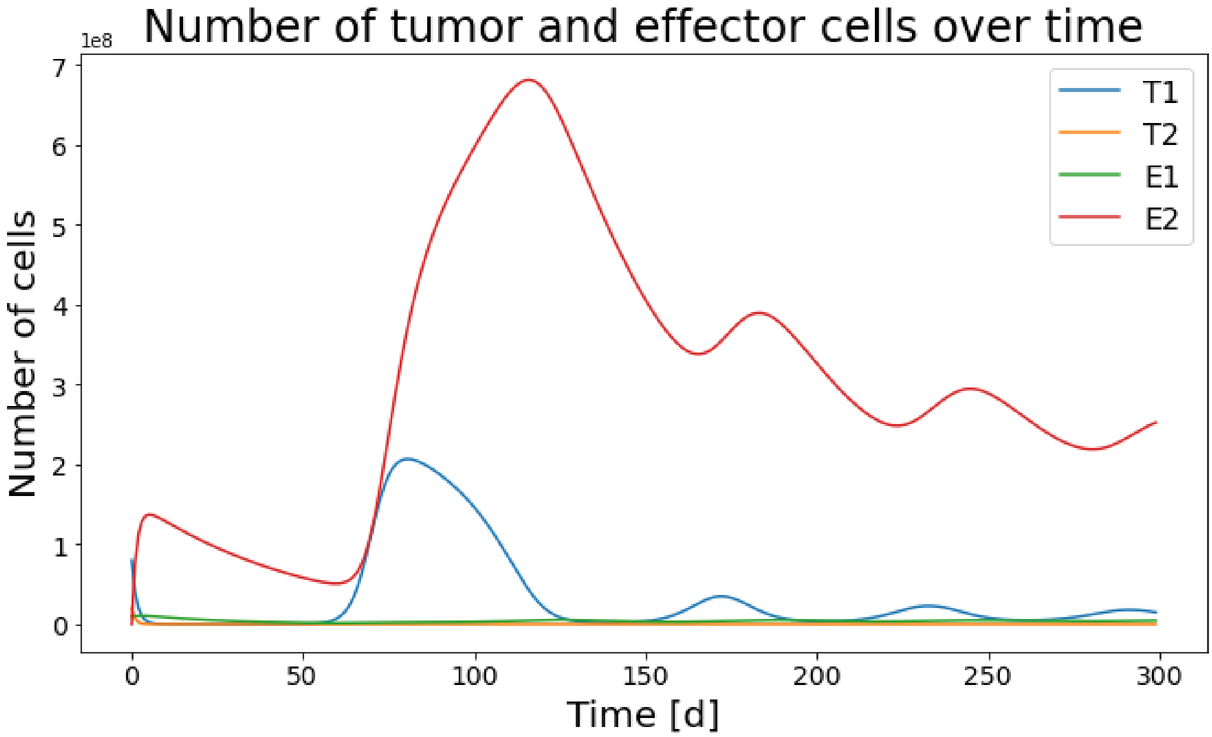


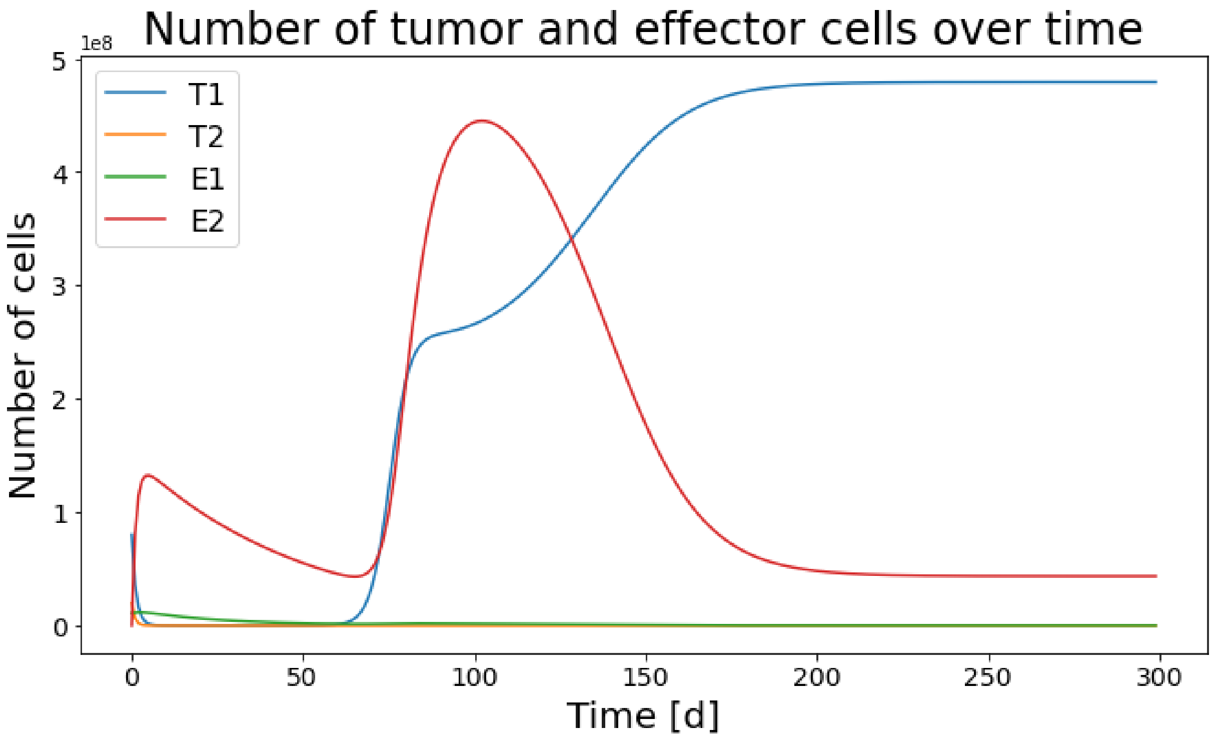

Number of tumor and effector cells over time
1e8
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
Number of cells
0
50
100
150
200
250
300
Time [d]
T1
T2
E1
E2

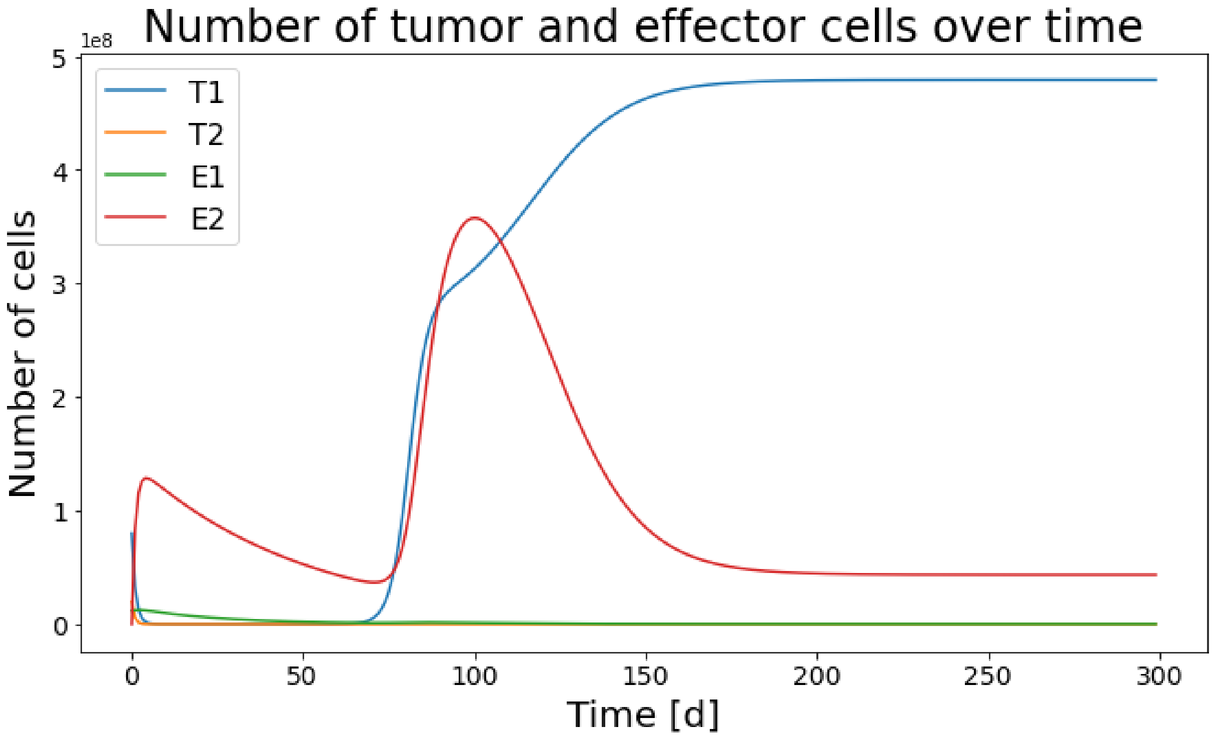

Number of tumor and effector cells over time
1e8
4
3
2
1
0
Number of cells
0
50
100
150
200
250
300
Time [d]
T1
T2
E1
E2

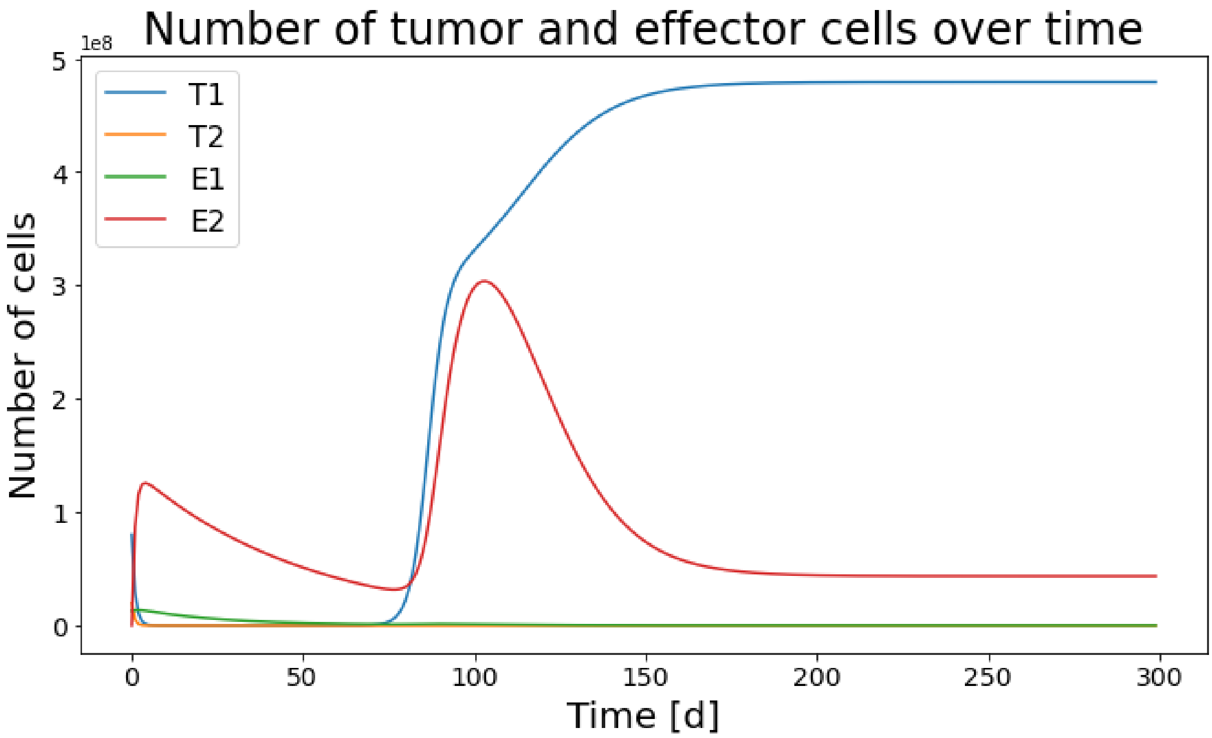

Number of tumor and effector cells over time
1e8
5
4
3
2
1
0
Number of cells
0 50 100 150 200 250 300
Time [d]
T1
T2
E1
E2

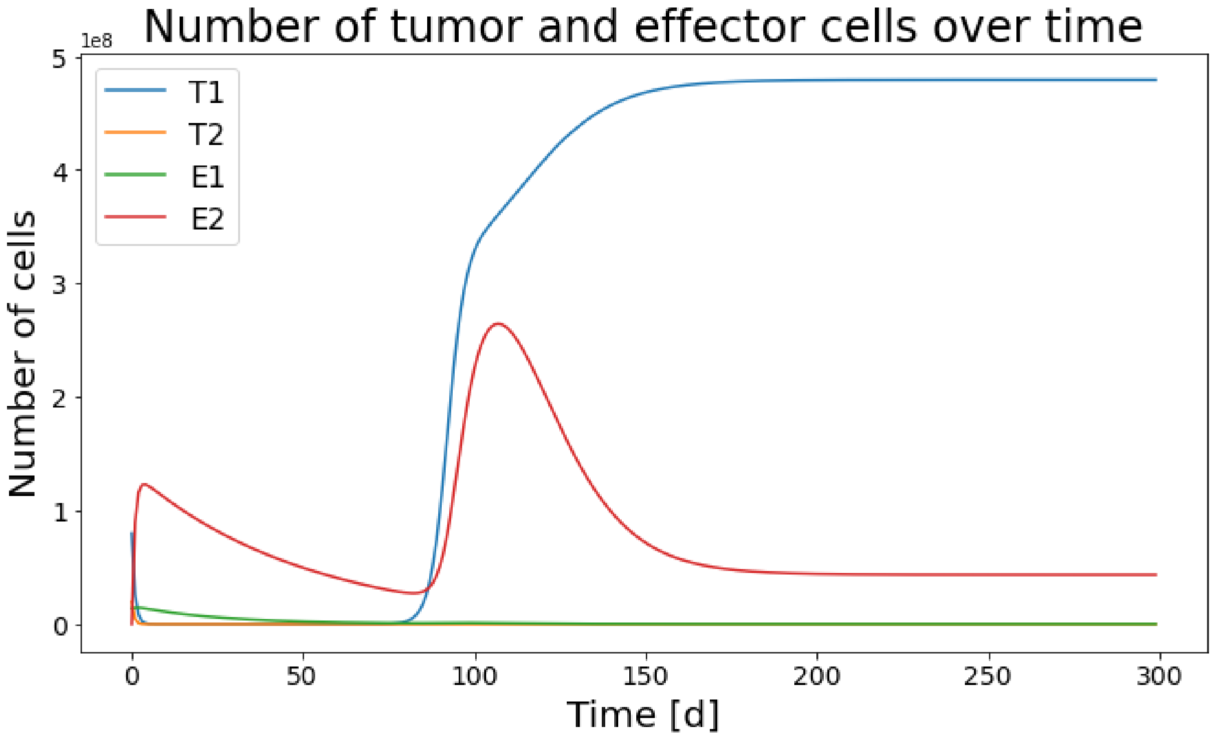

Number of tumor and effector cells over time
1e8
6
5
4
3
2
1
0
Number of cells
0 50 100 150 200 250 300
Time [d]
T1
T2
E1
E2

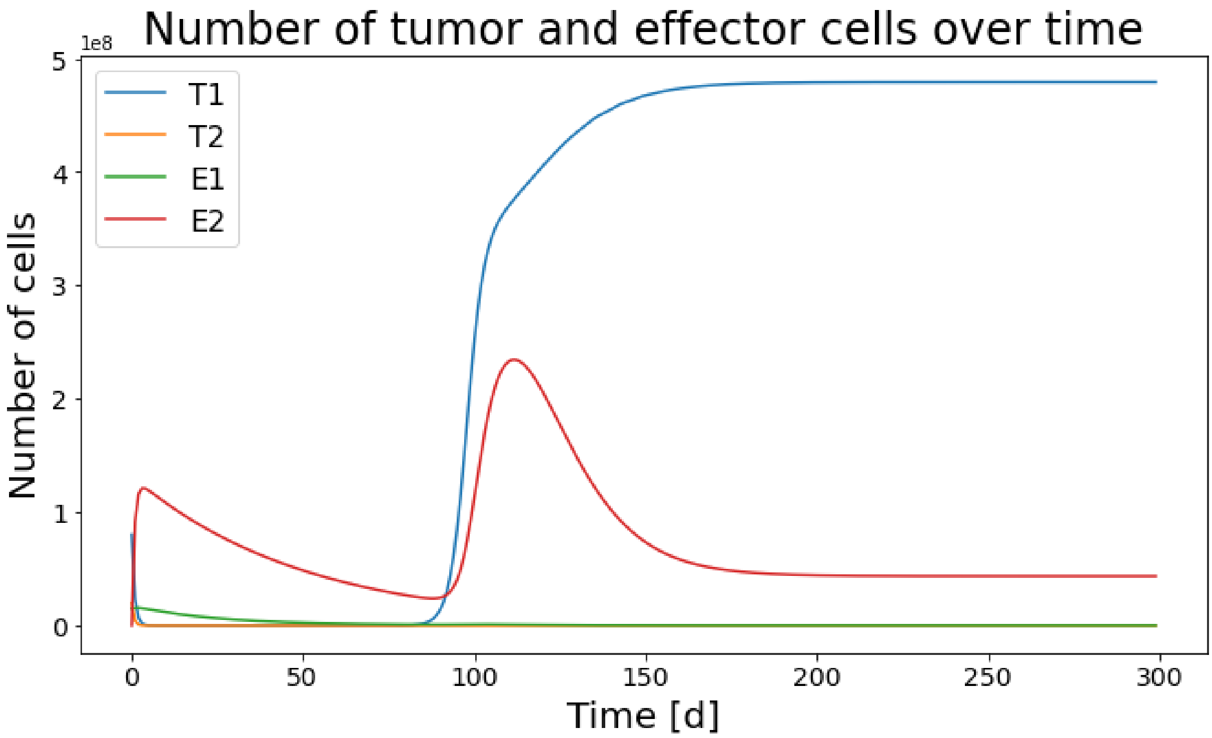

Number of tumor and effector cells over time
1e8
7
6
5
4
3
2
1
0
Number of cells
0
50
100
150
200
250
300
Time [d]
T1
T2
E1
E2


Number of tumor and effector cells over time
1e8
5
4
3
2
1
0
Number of cells
0
50
100
150
200
250
300
Time [d]
T1
T2
E1
E2


Number of tumor and effector cells over time
1e8
5
4
3
2
1
0
Number of cells
T1
T2
E1
E2
0
50
100
150
200
250
300
Time [d]


Number of tumor and effector cells over time
1e8
5
4
3
2
1
0
Number of cells
T1
T2
E1
E2
0
50
100
150
200
250
300
Time [d]





In [49]:

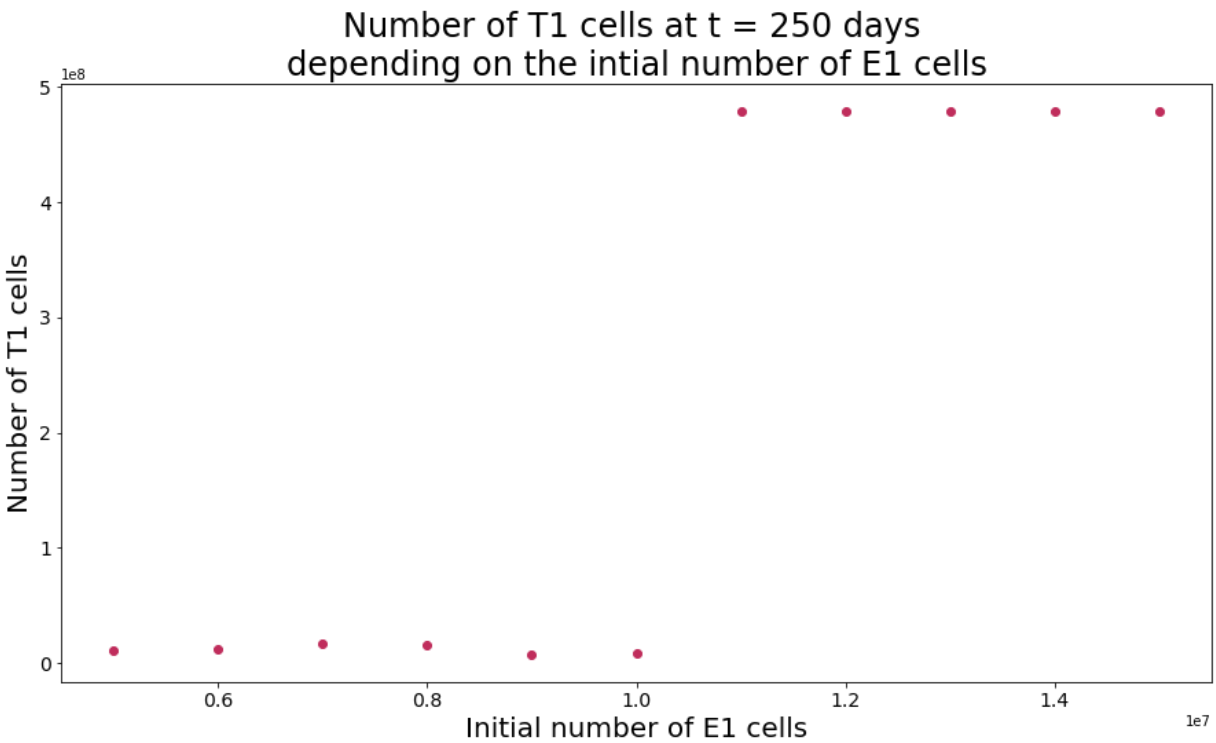
```
values_at_t250 = [values_at_t250[i] for i in range(0,len(values_at_t250),4)]
```

Plot the number of tumor cells against the number of initial E_1 cells.

In [55]:

```
fig = plt.figure(figsize=(15,8))
plt.scatter(E1_list, values_at_t250, color='#c02d5c')

plt.xlabel('Initial number of E1 cells', fontsize=20) # the horizontal axis represen
plt.ylabel('Number of T1 cells', fontsize=20)
plt.xticks(size=14)
plt.yticks(size=14)
```

```
plt.title('Number of T1 cells at t = 250 days \ndepending on the intial number of E1
# show how the colors correspond to the components of X
fig.savefig('Sensitivity_E1.jpg')
plt.show()
```



Extract values for the number of tumor cells for two different initial E_1 values over time and plot phase diagram and number of tumor cells against time.

In [79]:

```
def ode_system(t, state, g1,g2,a11,a12,a21,p1,d1,d2,e1,e2,r1,r2,r3,s1,s2,i12,i21,K1,
    T1,T2,E1,E2 = state

    dT1dt = g1*T1*(1-T1/K1) - a11*E1*T1 - a12*E2*T1 - i12*T1*T2
    dT2dt = g2*T2*(1-T2/K2) - a21*E1*T2 - i21*T1*T2
    dE1dt = p1 - d1*E1 - e1*(T1+T2)*E1 + ((r1*(T1+T2))/(s1+T1+T2))*E1
    dE2dt = -d2*E2 - e2*T1*E2 + ((r2*T1)/(s2+T1))*E2 + r3*E1*(T1+T2)
    return [dT1dt, dT2dt, dE1dt, dE2dt]

g1 = 0.514
g2 = 0.35 * g1
a11 = 1.1e-7
a12 = 1.1e-10
a21 = a11
p1 = 1.3e4
d1 = 4.12e-2
d2 = 2.0e-2
e1 = 3.42e-10
e2 = e1
r1 = 1.24e-1
r2 = 1.24e-3
r3 = 1.1e-7
s1 = 2.02e7
s2 = s1
i12 = 1.1e-9
```

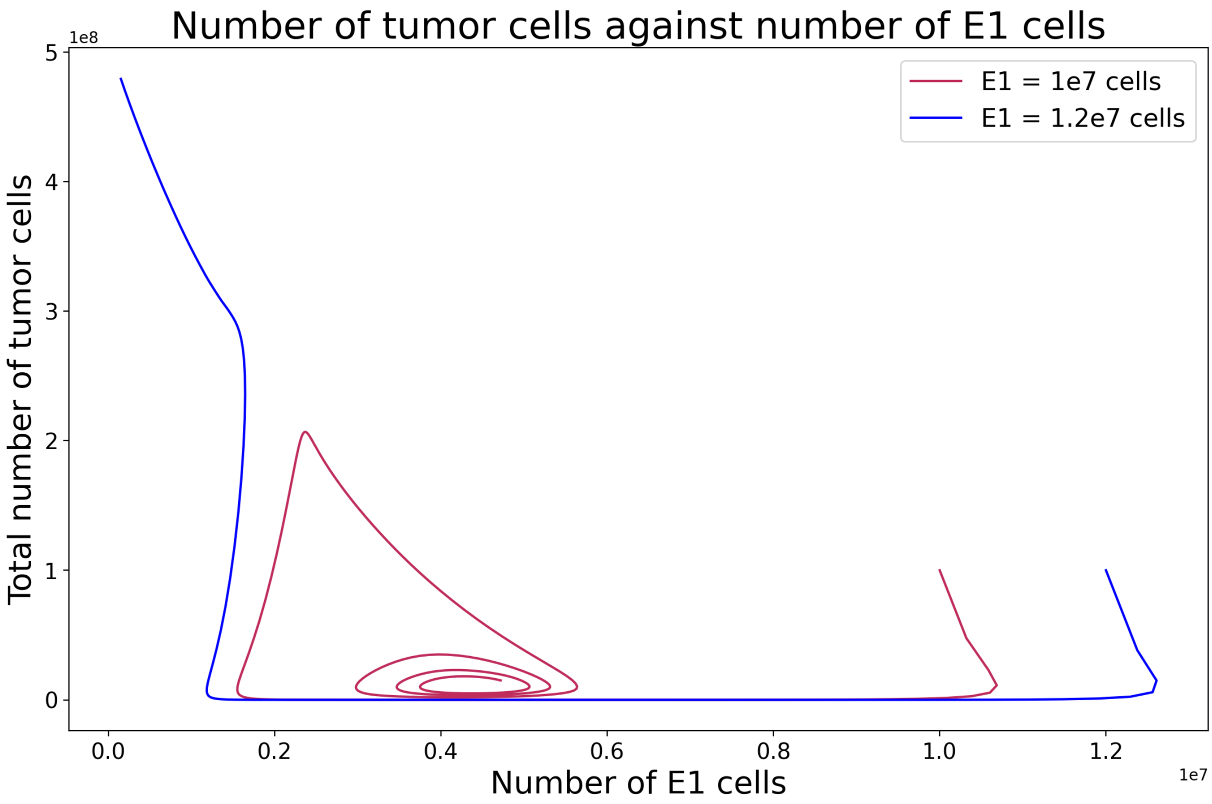
```
i21 = 1.5*i12
K1 = 5e8
K2 = K1

values = dict()

p = (g1,g2,a11,a12,a21,p1,d1,d2,e1,e2,r1,r2,r3,s1,s2,i12,i21,K1,K2)
E1_list = [1e7,1.2e7]
for i in range(len(E1_list)):
    E1 = E1_list[i]
    initial = [8.0e7,2.0e7,E1,0]

    t_span = (0, 300.0)
    t = np.arange(0, 300.0, 1)

    result_solve_ivp = solve_ivp(ode_system, t_span, initial, args=p, t_eval=t)

    labels = ['T1', 'T2', 'E1', 'E2']
    results = list()
    for i in range(result_solve_ivp.y.shape[0]):
        results.append(result_solve_ivp.y[i])
    values[E1] = results
```

In [67]:

```
#print(values[10000000][0])
tumor_cells = list()
E1_cells = list()
for i in range(len(values[10000000][0])):
    val = values[10000000][0][i] + values[10000000][1][i]
    tumor_cells.append(val)
    E1_cells.append(values[10000000][2][i])

tumor_cells_2 = list()
E1_cells_2 = list()
for i in range(len(values[12000000][0])):
    val = values[12000000][0][i] + values[12000000][1][i]
    tumor_cells_2.append(val)
    E1_cells_2.append(values[12000000][2][i])
```

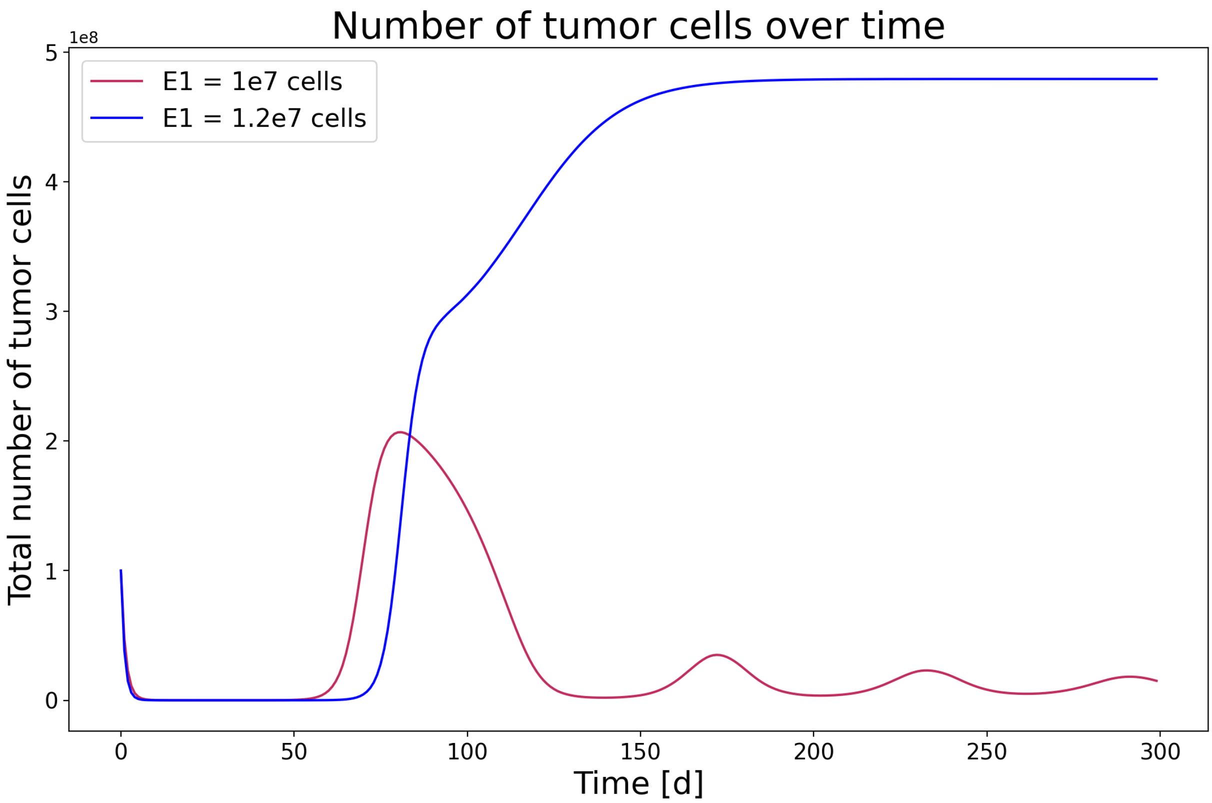
In [90]:

```
fig = plt.figure(figsize=(13,8), dpi=200)
plt.plot(E1_cells, tumor_cells, color='#c02d5c')
plt.plot(E1_cells_2, tumor_cells_2, color='blue')

labels=['E1 = 1e7 cells', 'E1 = 1.2e7 cells']
plt.xlabel('Number of E1 cells', fontsize=20) # the horizontal axis represents the t
plt.ylabel('Total number of tumor cells', fontsize=20)
plt.xticks(size=14)
plt.yticks(size=14)
plt.title('Number of tumor cells against number of E1 cells', fontsize = 24)
# show how the colors correspond to the components of X
plt.legend(labels, fontsize = 16)
fig.savefig('Phase_diagram.jpg')
plt.show()
```



In [89]:

```
fig = plt.figure(figsize=(13,8), dpi=200)
plt.plot(t, tumor_cells, color='#c02d5c')
plt.plot(t, tumor_cells_2, color='blue')

plt.xlabel('Time [d]', fontsize=20) # the horizontal axis represents the time
plt.ylabel('Total number of tumor cells', fontsize=20)
plt.xticks(size=14)
plt.yticks(size=14)
plt.title('Number of tumor cells over time', fontsize = 24)
labels=['E1 = 1e7 cells', 'E1 = 1.2e7 cells']
plt.legend(labels, fontsize = 16)
fig.savefig('Tumor_dynamics.jpg')
plt.show()
```



Sensitivity analysis with Sobol's method using the SALib package

In [102…

```
def ode_system(t, state,g1,g2,a11,a12,a21,p1,d1,d2,e1,e2,r1,r2,r3,s1,s2,i12,i21,K1,K
    T1,T2,E1,E2 = state

    dT1dt = g1*T1*(1-T1/K1) - a11*E1*T1 - a12*E2*T1 - i12*T1*T2
    dT2dt = g2*T2*(1-T2/K2) - a21*E1*T2 - i21*T1*T2
    dE1dt = p1 - d1*E1 - e1*(T1+T2)*E1 + ((r1*(T1+T2))/(s1+T1+T2))*E1
    dE2dt = -d2*E2 - e2*T1*E2 + ((r2*T1)/(s2+T1))*E2 + r3*E1*(T1+T2)
    return [dT1dt, dT2dt, dE1dt, dE2dt]

g10 = 0.514
g20 = 0.35 * g10
a110 = 1.1e-7
a120 = 1.1e-10
a210 = a110
p10 = 1.3e4
d10 = 4.12e-2
d20 = 2.0e-2
e10 = 3.42e-10
e20 = e10
r10 = 1.24e-1
r20 = 1.24e-3
r30 = 1.1e-7
s10 = 2.02e7
s20 = s1
i120 = 1.1e-9
i210 = 1.5*i12
K10 = 5e8
K20 = K10
T10 = 8.0e7
T20 = 2.0e7
E10 = 1.1e7
E20 = 0
```

```
t_span = (0.0, 300.0)
t = np.arange(0.0, 300.0, 1)

result_solve_ivp = solve_ivp(ode_system, t_span, initial, args=p)


#definition of the problem settings
#the parameters as and the initial parameters for T1, T2, E1, and E2 are varied
problem = {
    'num_vars': 23,
    'names': ['g1','g2','a11','a12','a21','p1','d1','d2','e1','e2','r1','r2','r3','s
             'T1','T2','E1','E2'],
    'bounds': [[0.8*g10,1.2*g10],[0.8*g20,1.2*g20],[0.8*a110,1.2*a110],[0.8*a120,1.2
               [0.8*p10,1.2*p10],[0.8*d10,1.2*d10],[0.8*d20,1.2*d20],[0.8*e10,1.2*e1
               [0.8*r10,1.2*r10],[0.8*r20,1.2*r20],[0.8*r30,1.2*r30],[0.8*s10,1.2*s1
               [0.8*i120,1.2*i120],[0.8*i210,1.2*i210],[0.8*K10,1.2*K10],[0.8*K20,1.
               [0.8*T20,1.2*T20],[0.8*E10,1.2*E10],[0,E10]]
}

#sampling with n = 10, leading to 480 samples
param_values = saltelli.sample(problem, 10)

#evaluation of the model
for i in range(len(param_values)):
    vals = param_values[i][:19]
    initial = param_values[i][19:]
    sol = solve_ivp(ode_system, t_span, initial, args=vals,t_eval=t)
    if i == 0:
        Y = sol.y
        #Y = sol.y.reshape((sol.y.shape[0]*sol.y.shape[1],1))
    else:
        #Y = np.append(Y, sol.y.reshape((sol.y.shape[0]*sol.y.shape[1],1)), axis=1)
        Y = np.append(Y, sol.y, axis=0)
print(Y.shape)
np.savetxt('test.out',Y, delimiter=',', fmt='%.18f')
```

```
(1920, 300)
```

In [103…

```
#reading of the output file
values = np.loadtxt("test.out",delimiter=',', dtype=float)
```

In [104…

```
#sobol analysis for the problem
S_indices = [sobol.analyze(problem, Y) for Y in values.T]
```

In [105…

```
S_indices[250]
```

Out[105…

```
{'S1': array([ 0.02716781,  0.03724329,  0.68603832, -0.0108566 ,  0.02716781,
         0.03735469,  0.66113939,  0.14234826,  0.02716781,  0.0356917 ,
         0.53884975, -0.02050896,  0.02716781,  0.03738773,  0.69411509,
         0.01289318,  0.02716781,  0.03705073,  0.66797168,  0.15937505,
         0.02716781,  0.03574569,  0.56959616]),
 'S1_conf': array([0.28246723, 0.28228893, 0.4001919 , 0.16771533, 0.28246723,
        0.28158532, 0.35682081, 0.14606378, 0.28246723, 0.28248444,
        0.4040731 , 0.08436769, 0.28246723, 0.28221236, 0.39757994,
        0.02322156, 0.28246723, 0.2823092 , 0.39876685, 0.24148875,
        0.28246723, 0.28282834, 0.4076909 ]),
 'ST': array([1.39339353, 1.39152681, 1.66569817, 0.15472811, 1.39339353,
        1.38977886, 1.57912166, 0.22341332, 1.39339353, 1.38877079,
        1.5528489 , 0.19359589, 1.39339353, 1.38959036, 1.62418098,
```

```
        0.00244128, 1.39339353, 1.39150177, 1.66896435, 0.17585036,
        1.39339353, 1.39076777, 1.61627606]),
 'ST_conf': array([0.35224733, 0.35234667, 0.17488881, 0.20148442, 0.35224733,
        0.35156502, 0.21090305, 0.20420268, 0.35224733, 0.35134082,
        0.17890132, 0.22677189, 0.35224733, 0.35211269, 0.17171705,
        0.00275712, 0.35224733, 0.35231695, 0.17232569, 0.23716687,
        0.35224733, 0.35227364, 0.18552178]),
 'S2': array([[        nan, -0.29269115, -0.62840731,  1.73251232, -0.28276076,
         -0.29099692, -0.53218296,  1.85318684, -0.28276076, -0.29125726,
         -0.5837529 ,  1.81497555, -0.28276076, -0.2920058 , -0.6501755 ,
          1.88168104, -0.28276076, -0.29267954, -0.62239189,  1.57386473,
         -0.28276076, -0.29008439, -0.49668357],
        [        nan,         nan, -0.73765196, -0.12765428,  0.43074142,
          0.41321208, -0.71184934, -0.29255893,  0.43074142,  0.41489648,
         -0.55630051, -0.10555761,  0.43074142,  0.41283557, -0.74655724,
         -0.15256801,  0.43074142,  0.4136288 , -0.72395238, -0.31555408,
          0.43074142,  0.41497791, -0.5943315 ],
        [        nan,         nan,         nan, -0.77952017, -0.22448088,
         -0.24172146, -1.34993836, -0.94562402, -0.22448088, -0.24003312,
         -1.19473304, -0.75803021, -0.22448088, -0.24208435, -1.38418485,
         -0.8056642 , -0.22448088, -0.24129404, -1.36148638, -0.966436  ,
         -0.22448088, -0.23996292, -1.23336682],
        [        nan,         nan,         nan,         nan,  0.06200287,
          0.06110934, -0.0268389 , -0.3244523 ,  0.06200287,  0.06318648,
          0.11722745, -0.15993682,  0.06200287,  0.0619492 , -0.00481126,
         -0.18559052,  0.06200287,  0.06207987,  0.01705616, -0.19031917,
          0.06200287,  0.06216235,  0.02666293],
        [        nan,         nan,         nan,         nan,         nan,
         -0.1166808 , -0.91595362,  1.38712838, -0.10130993, -0.11157578,
         -0.64917098,  1.77642132, -0.10130993, -0.11635792, -0.91831104,
          1.73122132, -0.10130993, -0.1157812 , -0.89478769,  1.52448669,
         -0.10130993, -0.11433324, -0.77811498],
        [        nan,         nan,         nan,         nan,         nan,
                nan, -0.71196073, -0.29267032,  0.43063003,  0.41478509,
         -0.55641191, -0.10566901,  0.43063003,  0.41272418, -0.74666864,
         -0.1526794 ,  0.43063003,  0.41351741, -0.72406377, -0.31566547,
          0.43063003,  0.41486651, -0.5944429 ],
        [        nan,         nan,         nan,         nan,         nan,
                nan,         nan, -0.91698207, -0.20086501, -0.21640594,
         -1.16937271, -0.73254858, -0.20086501, -0.21842927, -1.35747541,
         -0.77917764, -0.20086501, -0.21764968, -1.33475039, -0.94042323,
         -0.20086501, -0.21631282, -1.20673897],
        [        nan,         nan,         nan,         nan,         nan,
                nan,         nan,         nan, -0.18670675, -0.18452413,
         -0.00919386, -0.21718618, -0.18670675, -0.18417624, -0.02774291,
         -0.21000936, -0.18670675, -0.18448266, -0.00360749, -0.24871895,
         -0.18670675, -0.1839803 ,  0.00479224],
        [        nan,         nan,         nan,         nan,         nan,
                nan,         nan,         nan,         nan, -0.15765541,
         -0.69441079,  1.94188575, -0.14659676, -0.16135226, -0.90041902,
          2.01388757, -0.14659676, -0.16177779, -0.90190716,  1.806441  ,
         -0.14659676, -0.16041939, -0.8143942 ],
        [        nan,         nan,         nan,         nan,         nan,
                nan,         nan,         nan,         nan,         nan,
         -0.55474892, -0.10400602,  0.43229302,  0.41438717, -0.74500565,
         -0.15101642,  0.43229302,  0.4151804 , -0.72240079, -0.31400249,
          0.43229302,  0.4165295 , -0.59277991],
        [        nan,         nan,         nan,         nan,         nan,
                nan,         nan,         nan,         nan,         nan,
                nan, -0.61186228, -0.07841031, -0.09596791, -1.23492333,
         -0.65910197, -0.07841031, -0.09518436, -1.21203842, -0.82077422,
         -0.07841031, -0.09384169, -1.08363424],
        [        nan,         nan,         nan,         nan,         nan,
                nan,         nan,         nan,         nan,         nan,
```

```
           nan,          nan, -0.0299372 , -0.02562967,  0.23250446,
       -0.15165714, -0.0299372 , -0.02688073,  0.24107378, -0.30696319,
       -0.0299372 , -0.02510079,  0.30432497],
      [        nan,          nan,          nan,          nan,          nan,
               nan,          nan,          nan,          nan,          nan,
               nan,          nan,          nan, -0.10230305, -0.94007119,
        1.83732528, -0.08721445, -0.10261698, -0.9345143 ,  1.59713303,
       -0.08721445, -0.10131437, -0.82794885],
      [        nan,          nan,          nan,          nan,          nan,
               nan,          nan,          nan,          nan,          nan,
               nan,          nan,          nan,          nan, -0.74670168,
       -0.15271245,  0.43059699,  0.41348437, -0.72409682, -0.31569851,
        0.43059699,  0.41483347, -0.59447594],
      [        nan,          nan,          nan,          nan,          nan,
               nan,          nan,          nan,          nan,          nan,
               nan,          nan,          nan,          nan,          nan,
       -0.81270016, -0.23381206, -0.25059986, -1.36773947, -0.97368948,
       -0.23381206, -0.24926054, -1.23961968],
      [        nan,          nan,          nan,          nan,          nan,
               nan,          nan,          nan,          nan,          nan,
               nan,          nan,          nan,          nan,          nan,
               nan, -0.07213545, -0.06932463,  0.18763281, -0.16419377,
       -0.07213545, -0.06863254,  0.1942887 ],
      [        nan,          nan,          nan,          nan,          nan,
               nan,          nan,          nan,          nan,          nan,
               nan,          nan,          nan,          nan,          nan,
               nan,          nan, -0.12658215, -0.70556323,  1.27336095,
       -0.11440803, -0.12433407, -0.58005242],
      [        nan,          nan,          nan,          nan,          nan,
               nan,          nan,          nan,          nan,          nan,
               nan,          nan,          nan,          nan,          nan,
               nan,          nan,          nan, -0.72375981, -0.31536151,
        0.43093399,  0.41517047, -0.59413893],
      [        nan,          nan,          nan,          nan,          nan,
               nan,          nan,          nan,          nan,          nan,
               nan,          nan,          nan,          nan,          nan,
               nan,          nan,          nan,          nan, -0.94518445,
       -0.20756667, -0.22304526, -1.21537462],
      [        nan,          nan,          nan,          nan,          nan,
               nan,          nan,          nan,          nan,          nan,
               nan,          nan,          nan,          nan,          nan,
               nan,          nan,          nan,          nan,          nan,
       -0.19879749, -0.19644048, -0.05797585],
      [        nan,          nan,          nan,          nan,          nan,
               nan,          nan,          nan,          nan,          nan,
               nan,          nan,          nan,          nan,          nan,
               nan,          nan,          nan,          nan,          nan,
               nan, -0.19633713, -0.46455633],
      [        nan,          nan,          nan,          nan,          nan,
               nan,          nan,          nan,          nan,          nan,
               nan,          nan,          nan,          nan,          nan,
               nan,          nan,          nan,          nan,          nan,
               nan,          nan, -0.5928339 ],
      [        nan,          nan,          nan,          nan,          nan,
               nan,          nan,          nan,          nan,          nan,
               nan,          nan,          nan,          nan,          nan,
               nan,          nan,          nan,          nan,          nan,
               nan,          nan,          nan]]),
 'S2_conf': array([[       nan, 0.42916972, 0.45924884, 0.41674666, 0.43393348,
       0.42966459, 0.51383243, 0.4535609 , 0.43393348, 0.4285259 ,
       0.44386736, 0.43699371, 0.43393348, 0.42881309, 0.45892423,
       0.41757212, 0.43393348, 0.42913387, 0.46194014, 0.5512668 ,
       0.43393348, 0.43012956, 0.48956408],
      [       nan,        nan, 0.44872085, 0.31559244, 0.36818506,
```

```
 0.36464854, 0.38877254, 0.27209297, 0.36818506, 0.36590959,
 0.45431234, 0.27164112, 0.36818506, 0.36499717, 0.44474169,
 0.25355461, 0.36818506, 0.36528208, 0.44348097, 0.35660378,
 0.36818506, 0.3660778 , 0.47172987],
[       nan,        nan,        nan, 0.37891635, 0.33377765,
 0.3368736 , 0.65007168, 0.34544207, 0.33377765, 0.33817455,
 0.69296018, 0.23164247, 0.33377765, 0.33812904, 0.74365499,
 0.23208586, 0.33377765, 0.33826953, 0.73941238, 0.42759587,
 0.33377765, 0.3382123 , 0.72184265],
[       nan,        nan,        nan,        nan, 0.21222309,
 0.20922811, 0.17805377, 0.30759007, 0.21222309, 0.21116797,
 0.23650432, 0.27357799, 0.21222309, 0.21084632, 0.26515531,
 0.2458667 , 0.21222309, 0.21081031, 0.26821107, 0.26047204,
 0.21222309, 0.21079295, 0.25880638],
[       nan,        nan,        nan,        nan,        nan,
 0.44554772, 0.36930809, 0.52031692, 0.45003887, 0.44588416,
 0.47762152, 0.4086719 , 0.45003887, 0.4461979 , 0.44540069,
 0.42156167, 0.45003887, 0.44638496, 0.44787585, 0.55589221,
 0.45003887, 0.44721399, 0.48855196],
[       nan,        nan,        nan,        nan,        nan,
        nan, 0.38848231, 0.27227771, 0.36780643, 0.36551828,
 0.45346677, 0.27206253, 0.36780643, 0.36460234, 0.44365566,
 0.25395504, 0.36780643, 0.36488875, 0.44243688, 0.35683212,
 0.36780643, 0.36568564, 0.47080067],
[       nan,        nan,        nan,        nan,        nan,
        nan,        nan, 0.29589765, 0.31033298, 0.31409922,
 0.64668984, 0.19642437, 0.31033298, 0.31378965, 0.67225014,
 0.18677532, 0.31033298, 0.31404883, 0.67058068, 0.41837582,
 0.31033298, 0.31378027, 0.65006726],
[       nan,        nan,        nan,        nan,        nan,
        nan,        nan,        nan, 0.15797139, 0.15667642,
 0.21789416, 0.25060427, 0.15797139, 0.15677644, 0.23029168,
 0.22361951, 0.15797139, 0.15665293, 0.24223444, 0.22567333,
 0.15797139, 0.15655391, 0.2349322 ],
[       nan,        nan,        nan,        nan,        nan,
        nan,        nan,        nan,        nan, 0.41637449,
 0.44177361, 0.37183362, 0.42136634, 0.41712009, 0.42934291,
 0.35588903, 0.42136634, 0.41694131, 0.42174217, 0.48788766,
 0.42136634, 0.41769852, 0.45109187],
[       nan,        nan,        nan,        nan,        nan,
        nan,        nan,        nan,        nan,        nan,
 0.45451445, 0.27196846, 0.36855068, 0.36535956, 0.44449151,
 0.2538583 , 0.36855068, 0.36564453, 0.44323947, 0.35683692,
 0.36855068, 0.36643895, 0.47151136],
[       nan,        nan,        nan,        nan,        nan,
        nan,        nan,        nan,        nan,        nan,
        nan, 0.18634479, 0.36696821, 0.37035775, 0.71147782,
 0.18099725, 0.36696821, 0.37051529, 0.70832868, 0.41555055,
 0.36696821, 0.37028966, 0.68848578],
[       nan,        nan,        nan,        nan,        nan,
        nan,        nan,        nan,        nan,        nan,
        nan,        nan, 0.17005279, 0.16546759, 0.26154417,
 0.29491897, 0.17005279, 0.16819997, 0.25700887, 0.28480107,
 0.17005279, 0.16584558, 0.29525341],
[       nan,        nan,        nan,        nan,        nan,
        nan,        nan,        nan,        nan,        nan,
        nan,        nan,        nan, 0.43743605, 0.39743433,
 0.43434667, 0.44162763, 0.43731429, 0.38999396, 0.54711738,
 0.44162763, 0.43807284, 0.42535415],
[       nan,        nan,        nan,        nan,        nan,
        nan,        nan,        nan,        nan,        nan,
        nan,        nan,        nan,        nan, 0.4446206 ,
 0.25350497, 0.36815796, 0.36525424, 0.44335558, 0.35662609,
 0.36815796, 0.36604944, 0.47158929],
```

```
       [       nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
        0.20819205, 0.33367976, 0.33823435, 0.73519122, 0.43336702,
        0.33367976, 0.33795604, 0.71107675],
       [       nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan, 0.11139448, 0.10906296, 0.21771875, 0.26524063,
        0.11139448, 0.10869087, 0.2040408 ],
       [       nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan, 0.40324497, 0.4074585 , 0.59028371,
        0.40759624, 0.40386925, 0.46370135],
       [       nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan, 0.44338316, 0.35667307,
        0.36825447, 0.36614602, 0.47164086],
       [       nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan, 0.43753891,
        0.33623126, 0.34039844, 0.7100856 ],
       [       nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
        0.23738332, 0.23651913, 0.34984603],
       [       nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan, 0.46557448, 0.56365187],
       [       nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan, 0.47237338],
       [       nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan,        nan,        nan,
               nan,        nan,        nan]])}
```

The Sobol indices at the time t = 250 show the influences of the input variables and the initial values for T_1, T_2, E_1, and E_2 on the values of the ODEs at that time. The 'ST' indices show the total influence of each input variable, taking first-order and higher-order interactions into account. The highest influence on the output values seem to have the third variable a_{11}, which stands for the elimination of tumor cells of type 1 by effector cells type 1, the 15th variable s_2, the carrying capacity of tumor cell type 2, and the initial number of E_2 cells which all have values above 1.6. The smallest influences have the elimination rate of T_1 cells by E_2 cells and the influence that T_2 cells have on T_1 cells (i_{12}).

```
In [ ]:
```