Министерство образования и науки Российской Федерации
Федеральное государственное автономное образовательное учреждение высшего образования
«КАЗАНСКИЙ (ПРИВОЛЖСКИЙ) ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
ИНСТИТУТ ВЫЧИСЛИТЕЛЬНОЙ МАТЕМАТИКИ И ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ
КАФЕДРА ВЫЧИСЛИТЕЛЬНОЙ МАТЕМАТИКИ

Специальность: «01.03.02 Прикладная математика и информатика»

Курсовая работа по курсу «Численные методы» на тему: «Математическая модель роста опухоли»

Выполнил:
Студент группы 09-813
Махмутов Ринат -

Проверил:
Макаров М.В.

КАЗАНЬ – 2021 г.

Содержание

Постановка задачи

Клетки опухоли содержат особые вещества (антигены), которые вызывают резкую иммунную реакцию у больного. Эта реакция состоит в том, что производятся клетки — лимфоциты, которые атакуют и уничтожают клетки опухоли. Модель оперирует следующими переменными (имеется в виду плотность названной популяции клеток):

- L — свободные лимфоциты на поверхности опухоли;
- C — опухолевые клетки внутри опухоли и на ее поверхности;
- CS — опухолевые клетки на поверхности опухоли;
- CN — опухолевые клетки на поверхности опухоли, не связанные лимфоцитами;
- CF — опухолевые клетки внутри и на поверхности опухоли, не связанные лимфоцитами;

Ясно, что $C = C_F - C_N + C_S$. Предполагается, что опухоль всегда имеет форму шара, так что$C_S = K_1 C^{\frac{2}{3}}$, где K_1— постоянная, и что взаимодействие опухолевых клеток с лимфоцитами происходит только на поверхности опухоли. Будем считать, что между количеством свободных и связанных лимфоцитами клеток опухоли выполняется соотношение $C_S - C_N = K_2 C_N L$ (правдоподобно ли это предположение?). Из приведенных соотношений следует, что $C_F = C - \frac{K_1 K_2 L C^{\frac{2}{3}}}{(1 + K_2 L)}, C_N = \frac{K_1 C^{\frac{2}{3}}}{(1 + K_2 L)}$, т.е. переменные L и C можно взять за основные переменные модели, которая сводится к системе уравнений

$$L' = (-\lambda_1 + \alpha_1 C_N (1 - \frac{L}{L_M}))L,$$

$$C' = \lambda_2 C_F - \alpha_2 C_N L\ .$$

Здесь λ_1 характеризует уровень естественной смертности лимфоцитов, следующее слагаемое — их стимуляцию: когда L мало, стимуляция свободных лимфоцитов возрастает линейно с ростом C_N и что существует максимальный размер популяцииL_M , при котором стимуляция обращается в нуль. Первое слагаемое во втором уравнении

описывает рост опухоли, не подвергающейся атакам лимфоцитов, а второй член учитывает взаимодействие свободных лимфоцитов с опухолевыми клетками на поверхности опухоли. Подставляя в уравнения значения C_N и C_F, можно переписать их в виде

$$x' = (-\lambda_1 + \frac{\beta_1 y^{\frac{2}{3}}(1 - x/c)}{(1 + x)})x,$$
$$y' = \lambda_2 y - \frac{\beta_2 x y^{\frac{2}{3}}}{(1 + x)}, \qquad (1)$$

где$x = K_2 L$, $c = K_2 L_M$, $y = K_1 C$, а $\lambda_1, \lambda_2, \beta_1, \beta_2$ — положительные параметры. Так как x и y — размеры популяций, они должны быть неотрицательными, а x не может превышать c, поскольку L ограничено сверху величиной L_M. Уравнения (1) дополняются начальными условиями

$$x(0) = x_0, y(0) = y_0. \qquad (2)$$

Метод Рунге-Кутты

Напишите программу интегрирования задачи Коши для системы из n уравнений первого порядка вида

$$y' = f(t, y), y(0) = y_0, y(t) \in \mathfrak{R}^n,$$

на произвольном отрезке $[a, b]$, используя метод Рунге – Кутты 3-го порядка точности с постоянным шагом h:

$$\begin{aligned} k_1 &= f(t_n, y_n),\\ k_2 &= f(t_n + \tfrac{h}{2}, y_n + \tfrac{h}{2}k_1),\\ k_3 &= f(t_n + \tfrac{3}{4}h, y_n + \tfrac{3}{4}hk_2),\\ y_{n+1} &= y_n + h\,\frac{2k_1 + 3k_2 + 4k_3}{9}, \end{aligned} \qquad (3)$$

где t_i — точки сетки $y_i \equiv y(t_i)$, а h — шаг сетки

Тестовая задача.

Постановка тестовой задачи.

Для отладки работы метода Рунге-Кутты было предложено решить следующую задачу Коши для системы из двух уравнений первого порядка:

$$y_1{}' = \frac{-sin(t)}{(1+e^{2t})^{1/2}} + y_1(y_1^2 + y_2^2 - 1),$$

$$y_2{}' = \frac{cos(t)}{(1+e^{2t})^{1/2}} + y_2(y_1^2 + y_2^2 - 1),$$

на отрезке [0,5] с точным решением

$$y_1 = \frac{cos(t)}{(1+e^{2t})^{1/2}}, y_2 = \frac{sin(t)}{(1+e^{2t})^{1/2}}.$$

Откладка тестовой задачи

Проверим правильность работы метода, сравнивая результаты работы программы с точным решением. Пусть $n = 20$, тогда $h = \frac{5}{20} = 0.25$.

Таблица 1: Сравнение точного и приближенного решения для y_1

x	y_1	u_1	$\mid u_1 - y_1 \mid$
0.0	0.7071067811865475	0.7071067811865475	0.0
0.25	0.5953794618936874	0.5953418245595662	3.763733412120995e-05
0.5	0.45516110141834376	0.45511047641185903	5.062500648472801e-05
0.75	0.31255632087797713	0.3125138355922686	4.2485285708548925e-05
1.0	0.1865544767770587	0.18654356477166084	1.0912005397850644e-05
1.25	0.0868113628504781	0.08684723199450378	3.586914402568808e-05
1.5	0.015321775091512974	0.015404781515025492	8.300642351251851e-05
1.75	-0.03063607269527452	-0.030517177497581344	0.00011889519769317514
2.0	-0.05594936837135983	-0.0558105660106438З	0.0001388023607159991
2.25	-0.0659876972787596	-0.06584429085151261	0.0001434064272469865
2.5	-0.06567757115121717	-0.06554143591823065	0.00013613523298651453
2.75	-0.05908944527214152	-0.05896830204590676	0.0001211432262347642
3.0	-0.04933004883851279	-0.04922785005319204	0.00010219878532075066
3.25	-0.038599879051014176	-0.038517647076559636	8.223197445453995e-05
3.5	-0.028328916584278767	-0.028265657109657404	6.325947462136258e-05
3.75	-0.019338862321472965	-0.01929237200846521	4.649031300775486e-05
4.0	-0.012002391253746456	-0.011969892964138872	3.24982896075833e-05
4.25	-0.006383850589768047	-0.006362449054364513	2.1401535403533893e-05
4.5	-0.002354606728567357	-0.002341585323317062	1.302140525029490бe-05
4.75	0.00031829992425301936	0.00032531019096420044	7.0102667111810834e-06
5.0	0.0019083105769651	0.0019112573863128354	2.946809347735514e-06

$u_1 = \frac{cos(t)}{(1 + exp^{2t})^{\frac{1}{2}}}$ — точное решение. y_1 — приближенное решение, полученное методом Рунге-Кутты. $| u_1 - y_1 |$ — погрешность метода.

Приведём те же выкладки для y_2.

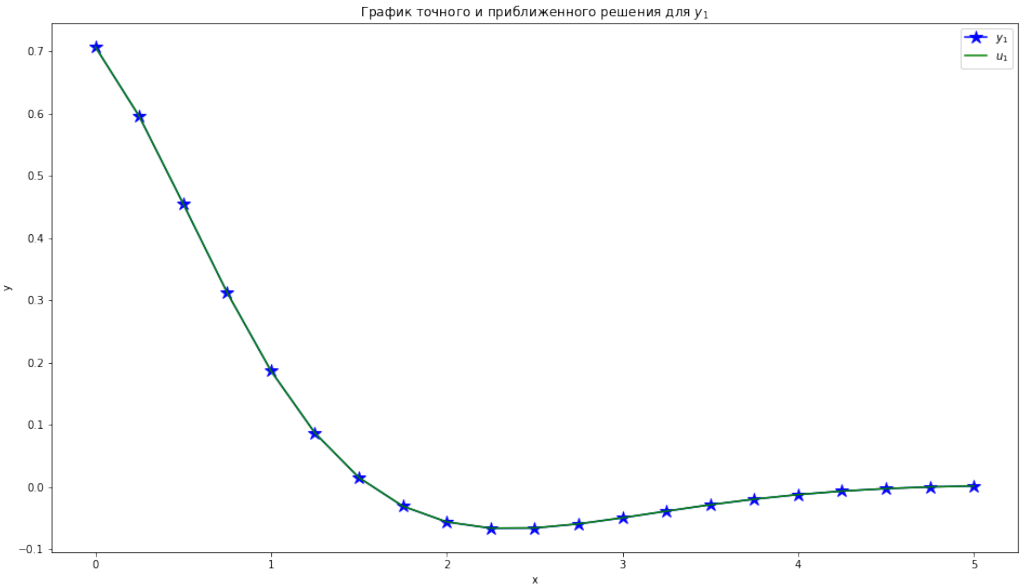
Таблица 2: Сравнение точного и приближенного решения для y_2

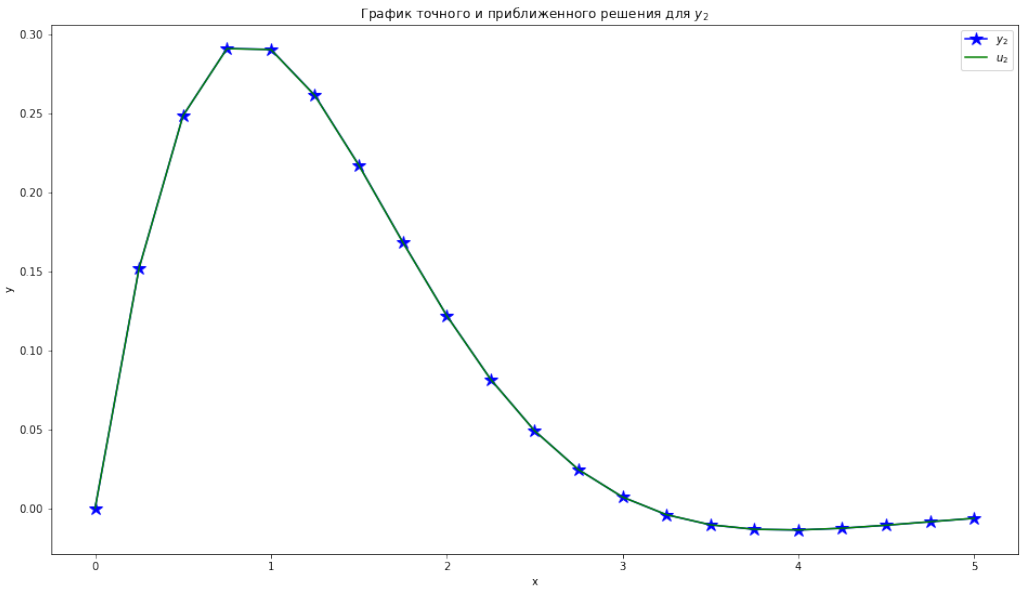
x	y_2	u_2	$\| u_2 - y_2 \|$
0.0	0.0	0.0	0.0
0.25	0.15201085401944445	0.15201572526627677	4.871246832321985e-06
0.5	0.2486491137803337	0.24862798641779227	2.1127362541423222e-05
0.75	0.29120626664580906	0.2911367829213013	6.948372450776841e-05
1.0	0.29064274363509335	0.29052438876004666	0.00011835487504668896
1.25	0.2615199953271708	0.2613727956695884	0.0001471996575824197
1.5	0.21737872460606933	0.21722929333780266	0.0001494312682666643
1.75	0.16859672474277368	0.1684664139493016	0.00013031079347208818
2.0	0.12204772348697429	0.12194831152444517	9.941196252911955e-05
2.25	0.08162196103691274	0.08155655701937517	6.540401753757363e-05
2.5	0.04899507337927282	0.0489609140239571	3.4159355315720175e-05
2.75	0.024357838423828583	0.024349067122147353	8.771301681229776e-06
3.0	0.007007512822476102	0.007017259848061428	9.747025585326446e-06
3.25	-0.00421372741378967	-0.004192030584643722	2.169682914594795e-05
3.5	-0.010615986931942687	-0.010587909262924352	2.807766901833514e-05
3.75	-0.013468276489264765	-0.013438118142455352	3.01583468094134e-05
4.0	-0.013888204903266263	-0.013858996821326455	2.9208081939808422e-05
4.25	-0.0127913863370581	-0.012765038977058496	2.634735999960426e-05
4.5	-0.010881189847052723	-0.01085870867828481	2.2481168767912382e-05
4.75	-0.008663538217372296	-0.008645253078439408	1.8285138932887593e-05
5.0	-0.006475257133575279	-0.006461034275230167	1.4222858345112367e-05

$u_2 = \frac{sin(t)}{(1 + exp^{2t})^{\frac{1}{2}}}$ — точное решение. y_2 — приближенное решение, полученное методом Рунге-Кутты. $| u_2 - y_2 |$ — погрешность метода.

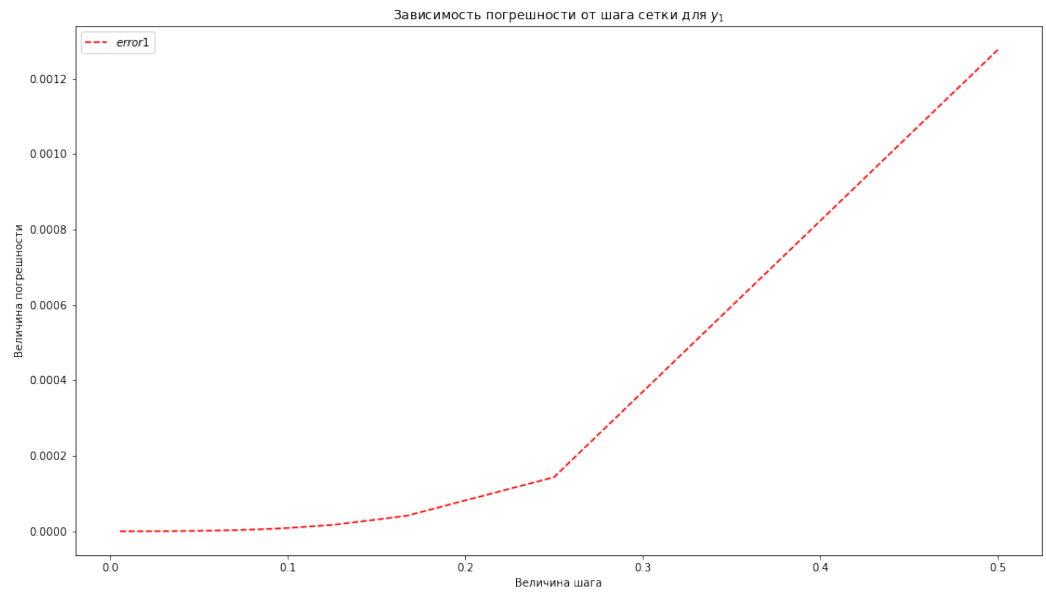
максимальная погрешность между u1 и y1 равна: 0.0001434064272469865
максимальная погрешность между u2 и y2 равна: 0.0001494312682666643

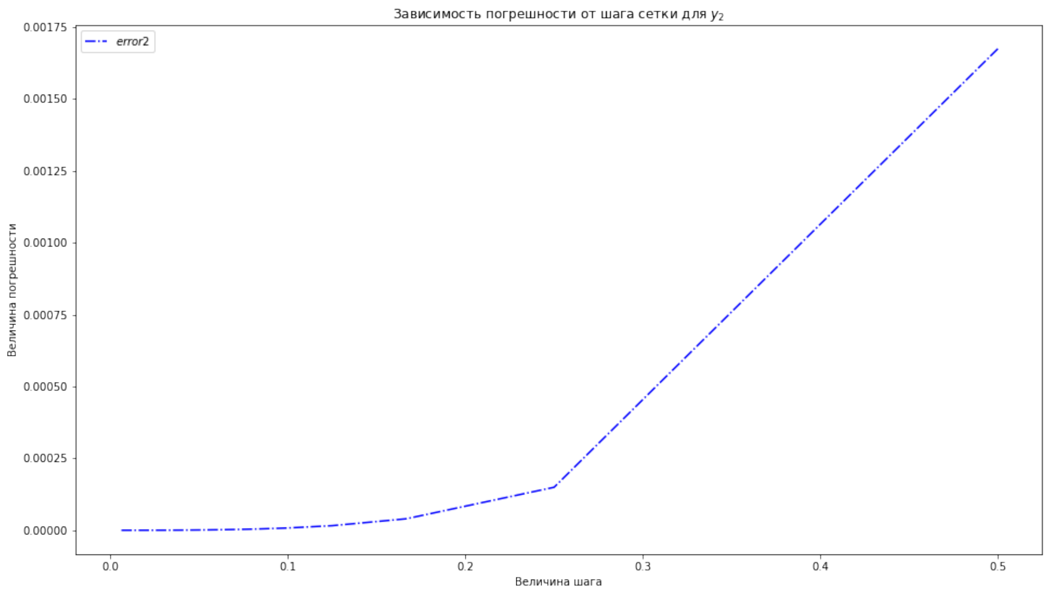
Графики





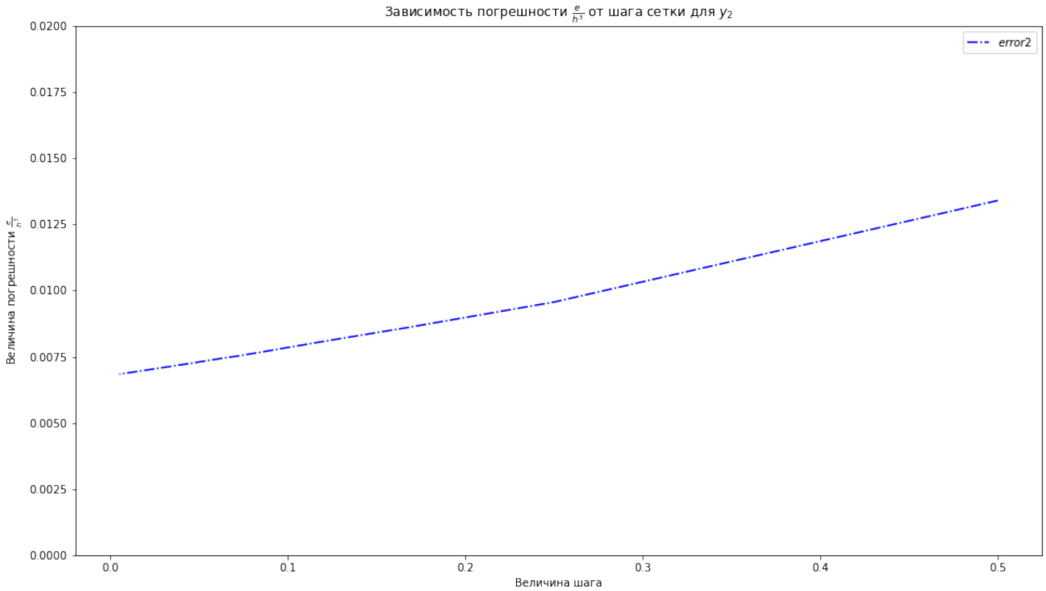
Построим графики зависимости погрешности от шага сетки. Для этого зададим массив шагов $H = \frac{5}{N_n}$, где $N_n = 10 : 10 : 1000$ и вычислим значение погрешности для каждой величины шага.

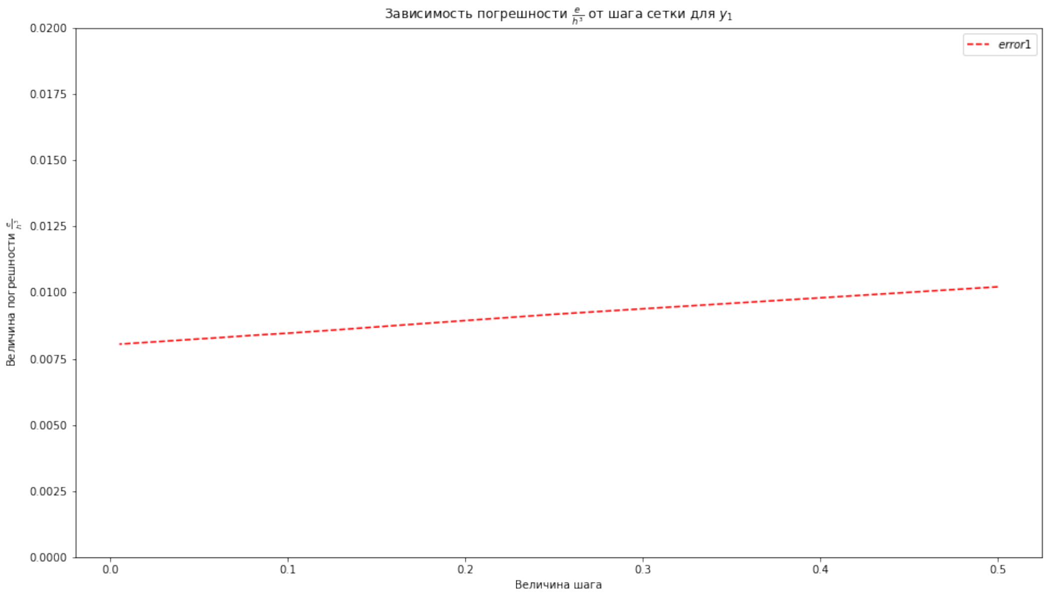




Графики показывают что, чем меньше шаг сетки, тем меньше погрешность решения.

Для проверки порядка точности метода отнормируем значения погрешности на H^3 и снова выведем аналогичные графики зависимости.





Вывод по тестовому заданию:
Оси ординат меняются незначительно, практически у нас константа. Получается, что метод Рунге-Кутты действительно имеет третий порядок точности.

Решение задачи о росте опухоли.

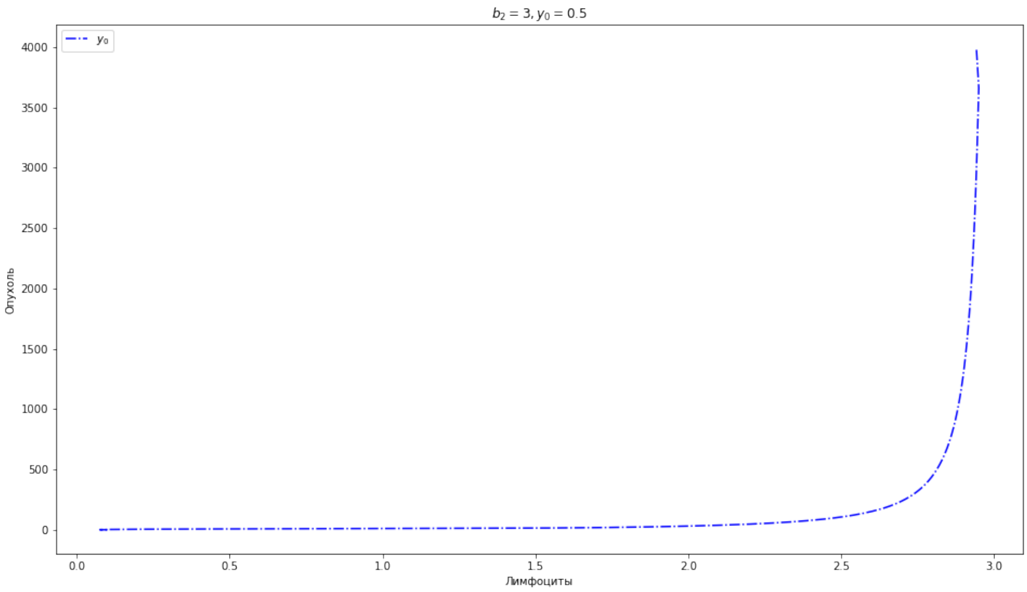
Решим задачу (1) и (2), где x — количество лимфоцитов, а y — размер опухоли. Для значений параметра $\beta_2 = 3; 3.48; 5$ при помощи разработанной процедуры рассчитаем динамику популяции при различных начальных значениях размера опухоли $y_0 \in [0.5,9]$. Построим графики наиболее характерных решений в координатах (x, y) и дадим их интерпретацию.
Параметры: $\lambda_1 = \lambda_2 = 1$,$\beta_1 = 1$,$c = 3$,$t \in [0,20]$.

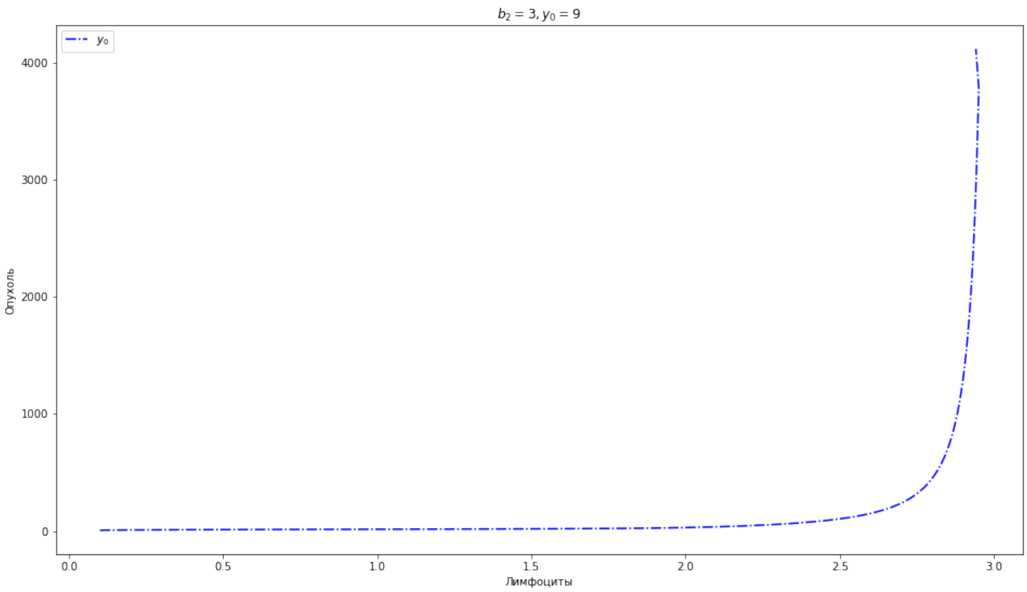
Значение c является предельным значением x, при котором стимуляция свободных лимфоцитов обращается в ноль. Коэффициент β_2 учитывается взаимодействие свободных лимфоцитов с опухолевыми клетками на поверхности опухоли.

Задача предполагает, что в начальной момент времени не происходит выработки лимфоцитов, потому что $x(0) = 0.1$ (достаточно маленькое величина, которая не повлияет на ход эксперимента).

Пусть $\beta_2 = 3$, то есть лимфоциты не слишком активно реагируют на опухоль.

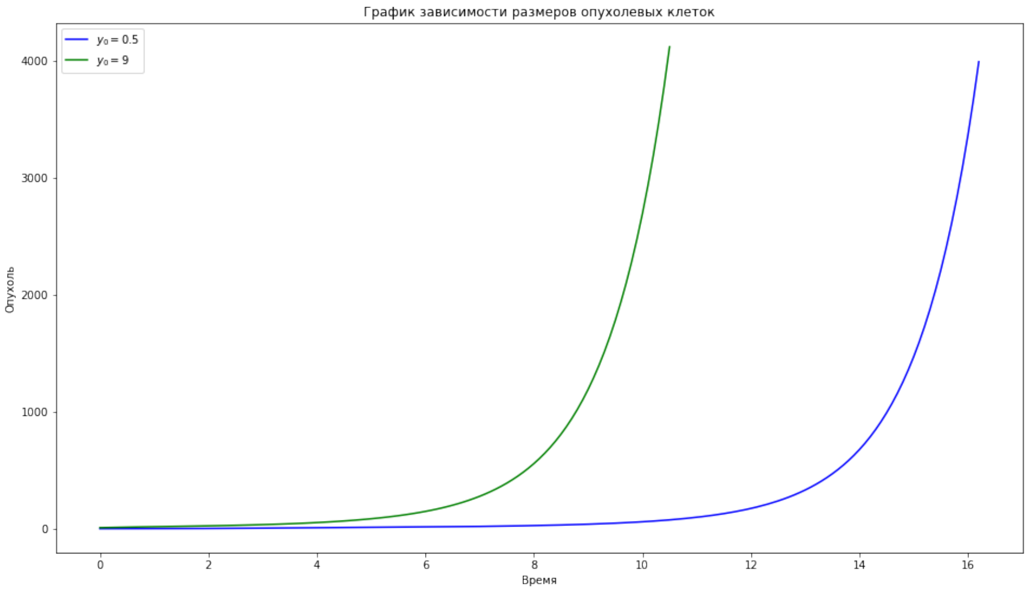
Рассмотрим графики при различных начальных значениях y_0.





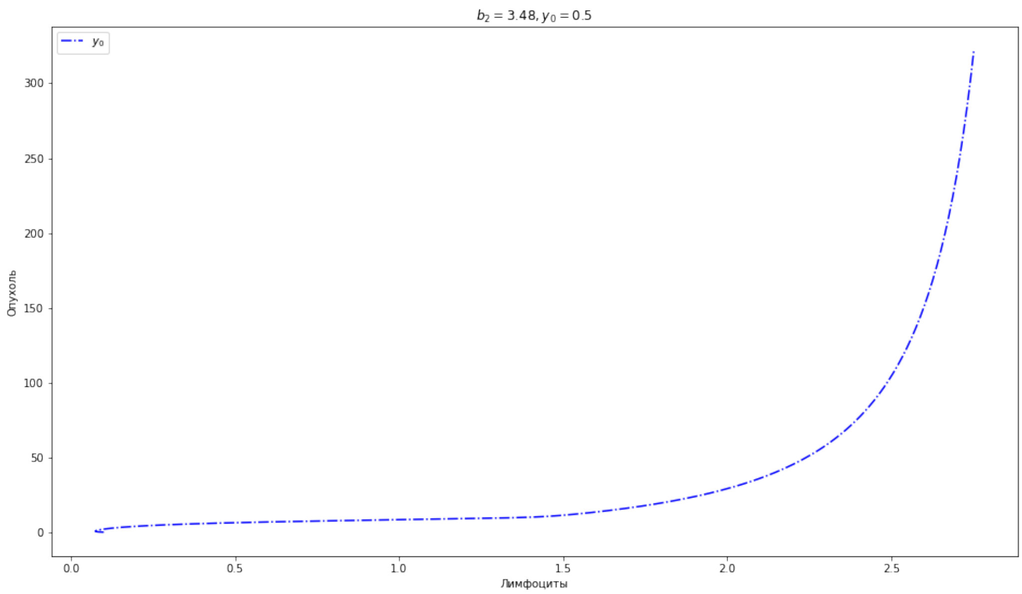
Наблюдаем, что независимо от того мал был начальный размер опухоли или велик — лимфоциты не справляются, их количество доходит до предельного значения, а опухоль продолжает расти.

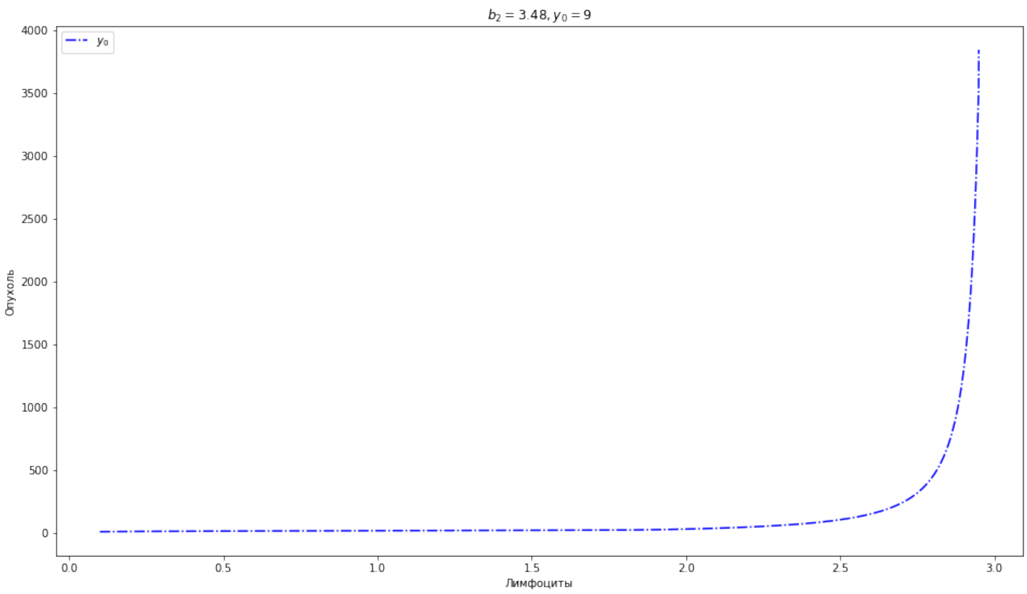
График зависимости роста опухоли от времени.



Видим, что чем больше y_0, тем быстрее растет опухоль.

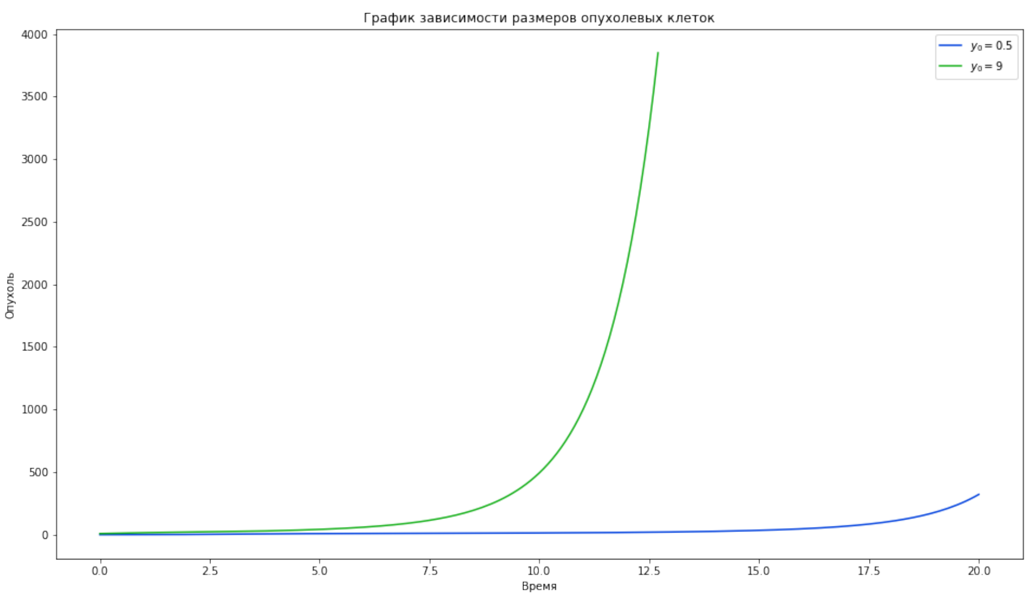
Проделаем тот же эксперимент в случае, когда $\beta_2 = 3.48$



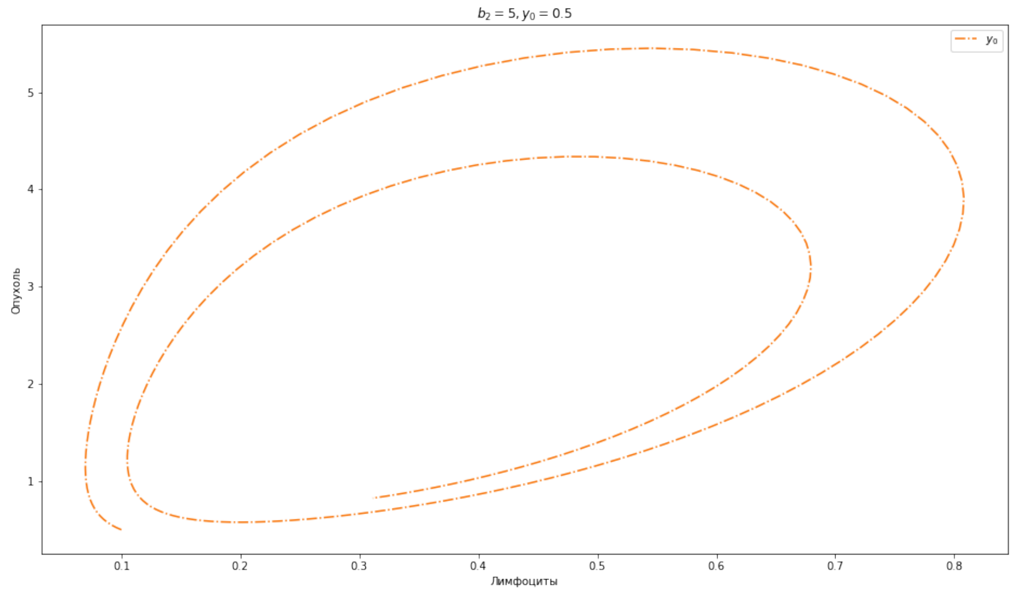


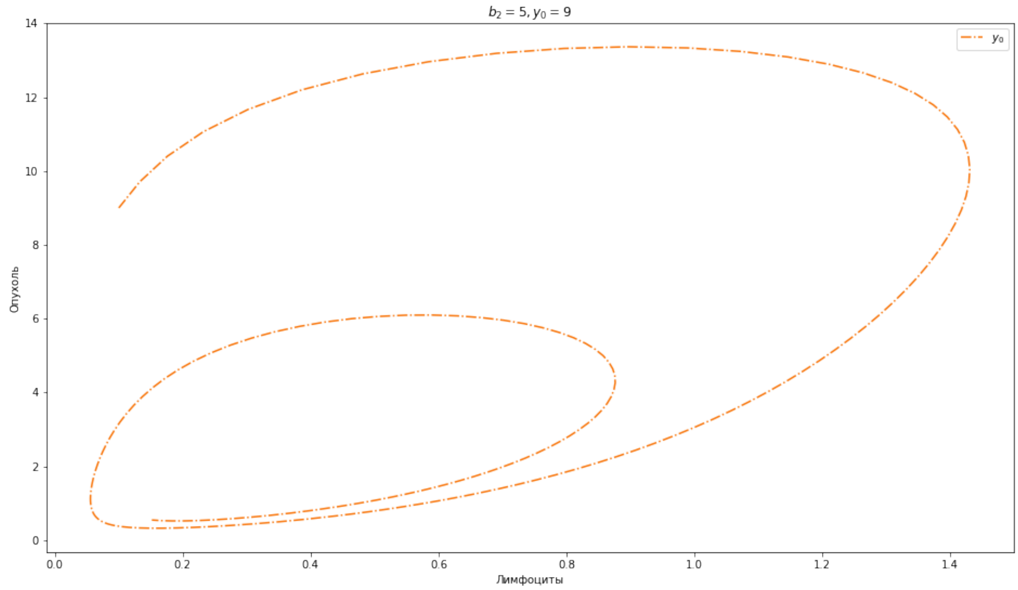
Вывод: лимфоциты по-прежнему не справляются, но отметим, что при $y_0 = 0.5$ конечный размер опухоли заметно меньше, чем при $y_0 = 9$.

Покажем, что при большой начальной величине опухоли, она растет быстрее, чем при маленькой, однако не быстро как при $\beta_2 = 3$, потому как здесь лимфоциты вырабатываются активнее.



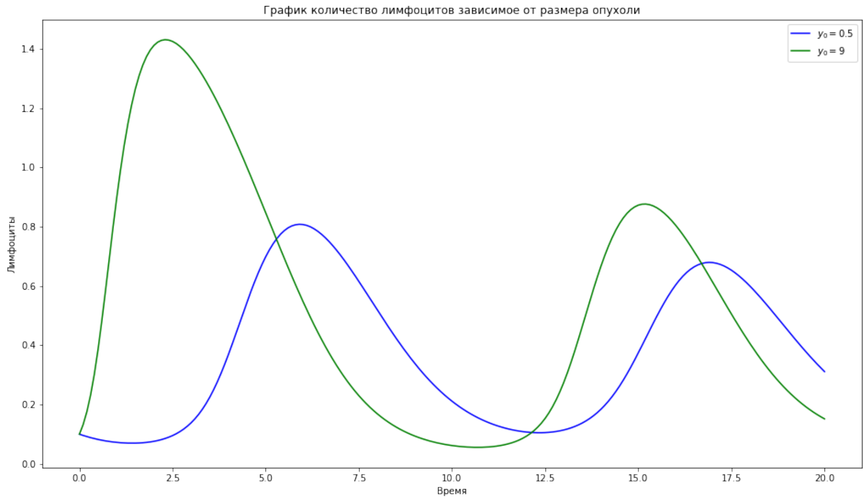
Рассмотрим задачу при $\beta_2 = 5$

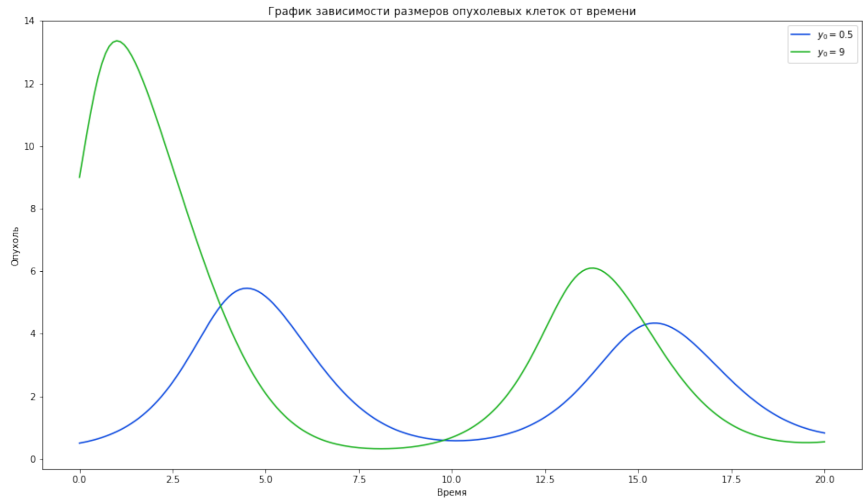


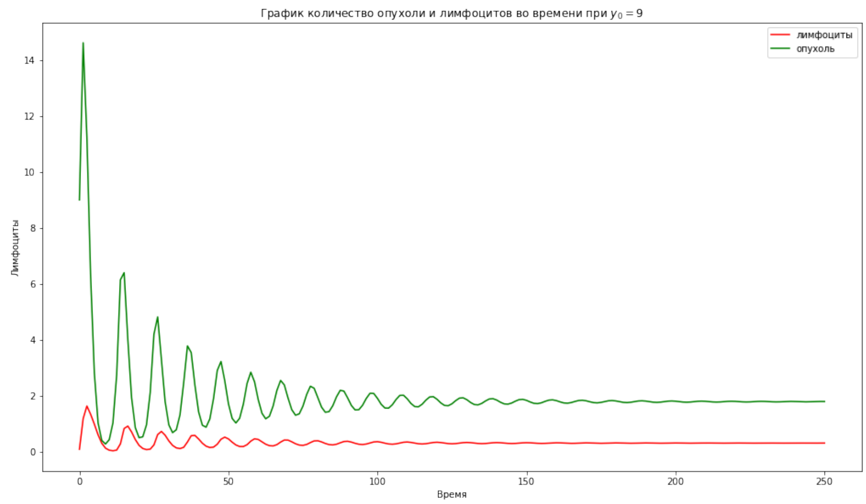


Графики демонстрируют, что функция уходит в предельный цикл, т.е. лимфоциты вырабатываются достаточно активно для того, чтобы сдерживать рост опухоли.

Рассмотрим графики изменения количество опухолевых клеток и лмфоцитов от времени.
Приведём график с теме же данными, но увеличим временной отрезок, что бы увидеть предельные значения.

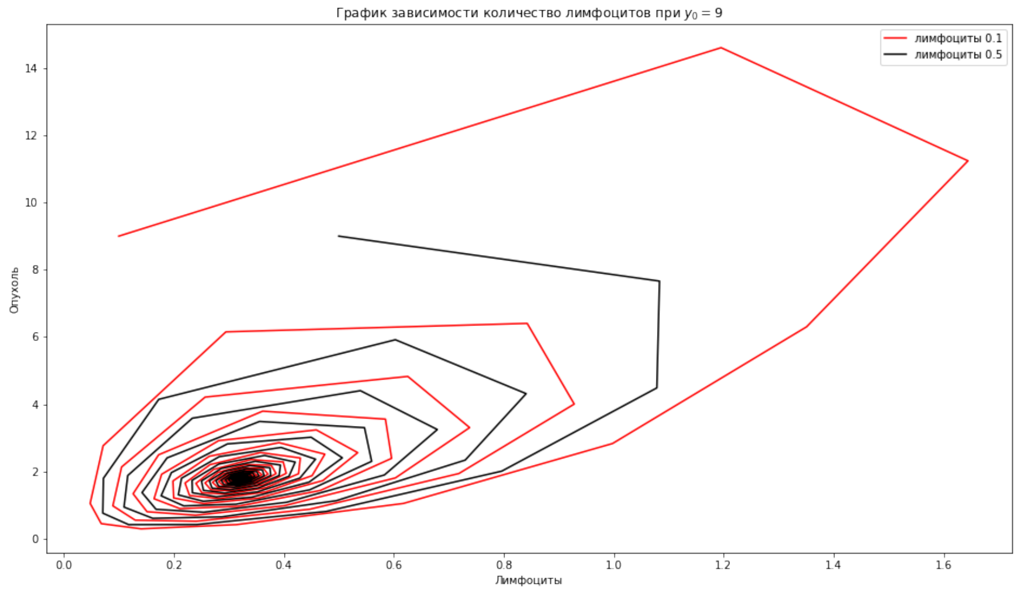






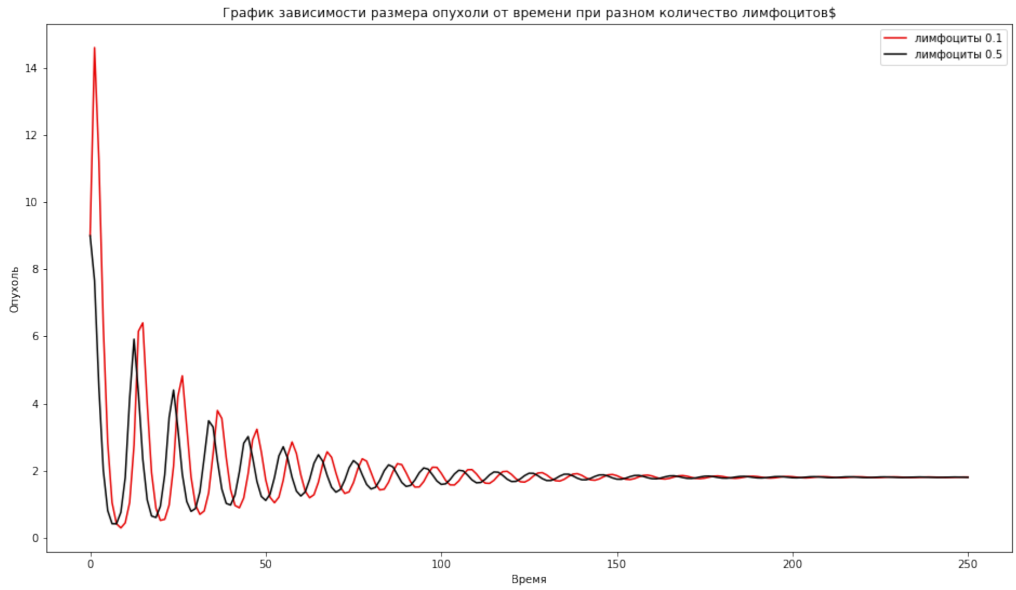
Вывод: увеличение опухолевых клеток порождает увеличение лимфоцитов, которые, в свою очередь, ведут к уменьшению опухоли, но в конечном итоге, количество опухолевых клеток не стремится к нулю, но и не возрастает.

Теперь предположим, что начальное значение лимфоцитов больше, чем в приведенных выше экспериментов



Функция уходит в предельный цикл заметно быстрее.

Приведем график зависимости размера опухоли от времени при разных начальных значениях количество лимфоцитов.



Большое начальное значение порождает более сильное колебания, т.к. опухолевые клетки вырабатываются в ответ на выброс лимфоцитов, как и наоборот. Видно, что в пределе опухоль стремится к одному значению, независимо от того, сколько лимфоцитов было до их взаимодействия.

Литература

1. Самарский А.А, Гулин А.В. Численные методы. — М.: Наука, 1989.
2. Бахвалов Н.С., Жидков Н.П., Кобельков Г.М. Численные методы. — М.: Наука, 1987.

Листинг.

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

# 1
def rk3(f, tspan, y0, N):
    '''метод Рунге-Кутта третьего порядка точности'''
    n = len(y0)
    T = np.zeros((N+1))
    Y = np.zeros((N+1, n))
    t = tspan[0]
    y = y0
    h = (tspan[1] - tspan[0])/float(N)
    T[0] = t
    Y[0, :] = np.transpose(y)
    i = 0
    while i < N:
        k1 = f(t, y)
        k2 = f(t + 0.5*h, y + 0.5*h*k1)
        k3 = f(t + 0.75*h, y + 0.75*h*k2)

        y = y + h*(2*k1 + 3*k2 + 4*k3)/9
        t += h

        T[i+1] = t
        Y[i+1, :] = np.transpose(y)

        i  += 1

        # условие основного задание(задача 4):
        # x не может превышать с
        if y[0] > 3:
            Y = Y[0:i, :]
            T = T[0:i]
            break

    return T, Y
```

```
# 2
def f_test(t, y):
    '''функция для правой части тестовой задачи'''
    dy = np.zeros(len(y))
    dy[0] = -np.sin(t) / ((1 + np.exp(2*t))**0.5) + y[0] * (y[0]**2 + y[1]**2 - 1)
    dy[1] = np.cos(t)/ ((1 + np.exp(2*t))**0.5) + y[1] * (y[0]**2 + y[1]**2 - 1)

    return dy

tspan = np.array([0, 5])
y0 = np.array([1/np.sqrt(2), 0])
N = 20

# решение
T, Y = rk3(f_test, tspan, y0, N)

# u1 и y1
u1 = np.array([ np.cos(t) / (1 + np.exp(2*t))**0.5 for t in T ])
lst_err1 = np.abs([u1[i] - Y[i, 0] for i in range(len(Y))])
max_err1 = np.max(lst_err1)

# u2 и y2
u2 = np.array([ np.sin(t) / (1+np.exp(2*t))**0.5 for t in T ] )
lst_err2 = np.abs([u2[i] - Y[i, 1] for i in range(len(Y))])
max_err2 = np.max(lst_err2)

# таблицы и графики
data1 = {f"$x$": T,
        f"$y_1$": Y[:, 0],
        f"$u_1$": u1,
        f"$$\mid u_1 - y_1\mid$$": lst_err1
       }
table1 = pd.DataFrame(data1)
table1.to_csv('table1.csv', index=False)

data2 = {f"$x$": T,
        f"$y_2$": Y[:, 1],
        f"$u_2$": u2,
        f"$$\mid u_2 - y_2\mid$$": lst_err2
       }
table2 = pd.DataFrame(dat2a)
table2.to_csv('table2.csv', index=False)
```

```
fig, ax = plt.subplots(figsize=(16,9))
ax.plot(T, Y[:, 0], color='blue', marker='*', markersize=12, label=f"$y_1$")
ax.plot(T, u1, color='green', label=f"$u_1$")
ax.set_xlabel('x')
ax.set_ylabel('y')
ax.set_title(f"График точного и приближенного решения для $y_1$")
ax.legend()
plt.savefig('1.png')
plt.show()

fig, ax = plt.subplots(figsize=(16,9))
ax.plot(T, Y[:, 1], color='blue', marker='*', markersize=12, label=f"$y_2$")
ax.plot(T, u2, color='green', label=f"$u_2$")
ax.set_xlabel('x')
ax.set_ylabel('y')
ax.set_title(f"График точного и приближенного решения для $y_2$")
ax.legend()
plt.savefig('2.png')
plt.show()

print(f"максимальная прогрешность между u1 и y1 равна:{max_err1}\n\
максимальная прогрешность между u2 и y2 равна:{max_err2}”)

# 3
arr_N = np.arange(10,1000,10)
# массив шагов
H = np.array([5/i for i in arr_N])
# массивы погрешности
arr_err1 = []
arr_err2 = []

for k in range(len(arr_N)):
    nk = arr_N[k]
    T, Y = rk3(f_test, tspan, y0, nk)
    # u1 и y1
    u1 = np.array([ np.cos(t) / (1 + np.exp(2*t))**0.5 for t in T ])
    lst_err1 = np.abs([u1[i] - Y[i, 0] for i in range(len(Y))])
    arr_err1 = np.append(arr_err1, np.max(lst_err1))
    # u2 и y2
    u2 = np.array([ np.sin(t) / (1+np.exp(2*t))**0.5 for t in T ] )
    lst_err2 = np.abs([u2[i] - Y[i, 1] for i in range(len(Y))])
    arr_err2 = np.append(arr_err2, np.max(lst_err2))
```

```
fig, ax = plt.subplots(figsize=(16,9))
ax.plot(H, arr_err1, color='red', linestyle='dashed', label=f"$error1$")
ax.set_xlabel('Величина шага')
ax.set_ylabel('Величина погрешности')
ax.set_title(f"Зависимость погрешности от шага сетки для $y_1$")
ax.legend()
plt.savefig('3.png')
plt.show()

fig, ax = plt.subplots(figsize=(16,9))
ax.plot(H, arr_err2, color='blue', linestyle='dashdot', label=f"$error2$")
ax.set_xlabel('Величина шага')
ax.set_ylabel('Величина погрешности')
ax.set_title(f"Зависимость погрешности от шага сетки для $y_2$")
ax.legend()
plt.savefig('4.png')
plt.show()

arr_err11 = arr_err1/(H**3)
arr_err22 = arr_err2/(H**3)

fig, ax = plt.subplots(figsize=(16,9))
ax.plot(H, arr_err11, color='red', linestyle='dashed', label=f"$error1$")
plt.ylim([0, 0.02])
ax.set_xlabel('Величина шага')
ax.set_ylabel(r"Величина погрешности $\frac{e}{h^3}$")
ax.set_title(r"Зависимость погрешности $\frac{e}{h^3}$ от шага сетки для
$y_1$")
ax.legend()
plt.savefig('5.png')
plt.show()

fig, ax = plt.subplots(figsize=(16,9))
ax.plot(H, arr_err22, color='blue', linestyle='dashdot', label=f"$error2$")
plt.ylim([0, 0.02])
ax.set_xlabel('Величина шага')
ax.set_ylabel(r"Величина погрешности $\frac{e}{h^3}$")
ax.set_title(r"Зависимость погрешности $\frac{e}{h^3}$ от шага сетки для
$y_2$")
ax.legend()
plt.savefig('6.png')
plt.show()
```

```
# 4
def f_main(t, u):
    '''функция для правой части главной задачи'''
    global lambda1, lambda2, beta1, beta2, c
    du = np.zeros(len(u))
    x = u[0]
    y = u[1]
    du[0] = (-lambda1 + beta1 * y**(2/3) * (1 - x/c) / (1+x)) * x
    du[1] = lambda2 * y - beta2 * x * y**(2/3) / (1+x)
    return du

tspan = [0, 20] # время
y0 = [0.1, 0.5] # первое число лимфоциты, второе - опухоль
N = 200         # шаг

global lambda1, lambda2, beta1, beta2, c
lambda1=1
lambda2=1
beta1=1
c=3

# условие
beta2 = 3   # уровень стемуляции лимфоцитов
y0[1] = 0.5 # количество опухолевых клеток

# эксперимент
T_05, Y_05 = rk3(f_main, tspan, y0, N)

# график
fig, ax = plt.subplots(figsize=(16,9))
ax.plot(Y_05[:, 0], Y_05[:, 1], color='blue', linestyle='dashdot', label=r"$y_0$" )
ax.set_xlabel('Лимфоциты')
ax.set_ylabel("Опухоль")
ax.set_title(f"$b_2={beta2}, y_0={y0[1]}$")
ax.legend()
plt.savefig('7.png')
plt.show()

# условие
y0[1] = 9 # количество опухолевых клеток
# эксперимент
T_9, Y_9 = rk3(f_main, tspan, y0, N)
# график
fig, ax = plt.subplots(figsize=(16,9))
ax.plot(Y_9[:, 0], Y_9[:, 1], color='blue', linestyle='dashdot', label=r"$y_0$")
```

```
ax.set_xlabel('Лимфоциты')
ax.set_ylabel("Опухоль")
ax.set_title(f"$b_2={beta2}, y_0={y0[1]}$")
ax.legend()
plt.savefig('8.png')
plt.show()

fig, ax = plt.subplots(figsize=(16,9))
ax.plot(T_05, Y_05[:, 1], color='blue', label=r"$y_0=0.5$")
ax.plot(T_9, Y_9[:, 1], color='green', label=r"$y_0=9$")
ax.set_xlabel('Время')
ax.set_ylabel('Опухоль')
ax.set_title(r"График зависимости размеров опухолевых клеток")
ax.legend()
plt.savefig('9.png')
plt.show()

# условие
beta2 = 3.48 # уровень стемуляции лимфоцитов
y0[1] = 0.5  # количество опухолевых клеток
# эксперимент
T_05, Y_05 = rk3(f_main, tspan, y0, N)
#график
fig, ax = plt.subplots(figsize=(16,9))
ax.plot(Y_05[:, 0], Y_05[:, 1], color='blue', linestyle='dashdot', label=r"$y_0$" )
ax.set_xlabel('Лимфоциты')
ax.set_ylabel("Опухоль")
ax.set_title(f"$b_2={beta2}, y_0={y0[1]}$")
ax.legend()
plt.savefig('10.png')
plt.show()

# условие
y0[1] = 9 # количество опухолевых клеток
# эксперимент
T_9, Y_9 = rk3(f_main, tspan, y0, N)
# график
fig, ax = plt.subplots(figsize=(16,9))
ax.plot(Y_9[:, 0], Y_9[:, 1], color='blue', linestyle='dashdot', label=r"$y_0$")
ax.set_xlabel('Лимфоциты')
ax.set_ylabel("Опухоль")
ax.set_title(f"$b_2={beta2}, y_0={y0[1]}$")
ax.legend()
plt.savefig('11.png')
plt.show()
```

```
fig, ax = plt.subplots(figsize=(16,9))
ax.plot(T_05, Y_05[:, 1], color='xkcd:blue', label=r"$y_0=0.5$")
ax.plot(T_9, Y_9[:, 1], color='xkcd:green', label=r"$y_0=9$")
ax.set_xlabel('Время')
ax.set_ylabel('Опухоль')
ax.set_title(r"График зависимости размеров опухолевых клеток")
ax.legend()
plt.savefig('12.png')
plt.show()

beta2 = 5   # уровень стемуляции лимфоцитов
y0[1] = 0.5 # число опухолевых клеток
# эксперимент
T_05, Y_05 = rk3(f_main, tspan, y0, N)
# график
fig, ax = plt.subplots(figsize=(16,9))
ax.plot(Y_05[:, 0], Y_05[:, 1], color='xkcd:orange', linestyle='dashdot',
label=r"$y_0$" )
ax.set_xlabel('Лимфоциты')
ax.set_ylabel("Опухоль")
ax.set_title(f"$b_2={beta2}, y_0={y0[1]}$")
ax.legend()
plt.savefig('13.png')
plt.show()

y0[1] = 9 # число опухолевых клеток
# эксперимент
T_9, Y_9 = rk3(f_main, tspan, y0, N)
#график
fig, ax = plt.subplots(figsize=(16,9))
ax.plot(Y_9[:, 0], Y_9[:, 1], color='xkcd:orange', linestyle='dashdot', label=r"$y_0$")
ax.set_xlabel('Лимфоциты')
ax.set_ylabel("Опухоль")
ax.set_title(f"$b_2={beta2}, y_0={y0[1]}$")
ax.legend()
plt.savefig('14.png')
plt.show()

fig, ax = plt.subplots(figsize=(16,9))
ax.plot(T_05, Y_05[:, 1], color='xkcd:blue', label=r"$y_0=0.5$")
ax.plot(T_9, Y_9[:, 1], color='xkcd:green', label=r"$y_0=9$")
ax.set_xlabel('Время')
ax.set_ylabel('Опухоль')
ax.set_title(r"График зависимости размеров опухолевых клеток от времени")
```

```
ax.legend()
plt.savefig('15.png')
plt.show()

fig, ax = plt.subplots(figsize=(16,9))
ax.plot(T_05, Y_05[:, 0], color='blue', label=r"$y_0=0.5$")
ax.plot(T_9, Y_9[:, 0], color='green', label=r"$y_0=9$")
ax.set_xlabel('Время')
ax.set_ylabel('Лимфоциты')
ax.set_title(r"График количество лимфоцитов зависимое от размера опухоли")
ax.legend()
plt.savefig('16.png')
plt.show()

beta2 = 5       # уровень стемуляции лимфоцитов
tspan = [0, 250] # время

# эксперимент
y0[1] = 9 # число опухолевых клеток
T, Y = rk3(f_main, tspan, y0, N)
fig, ax = plt.subplots(figsize=(16,9))
ax.plot(T, Y[:, 0], color='red', label=f'лимфоциты')
ax.plot(T, Y[:, 1], color='green', label=f'опухоль')
ax.set_xlabel('Время')
ax.set_ylabel('Лимфоциты')
ax.set_title(f'График количество опухоли и лимфоцитов во времени при $y_0=9$')
ax.legend()
plt.savefig('17.png')
plt.show()

beta2 = 5       # уровень стемуляции лимфоцитов
tspan = [0, 250] # время

# эксперименты
y0 = [0.1, 9] # первое число кол-во лимфоцитов, второе - кол-во опухоли
T1, Y1 = rk3(f_main, tspan, y0, N)
y0 = [0.5, 9] # первое число лимфоцитов, второе - кол-во опухоли
T2, Y2 = rk3(f_main, tspan, y0, N)
# график
fig, ax = plt.subplots(figsize=(16,9))
ax.plot(Y1[:, 0], Y1[:, 1], color='red', label=f'лимфоциты 0.1')
ax.plot(Y2[:, 0], Y2[:, 1], color='black', label=f'лимфоциты 0.5')
ax.set_xlabel('Лимфоциты')
ax.set_ylabel("Опухоль")
```

```
ax.set_title(f'График зависимости количество лимфоцитов при $y_0=9$')
ax.legend()
plt.savefig('18.png')
plt.show()

beta2 = 5       # уровень стемуляции лимфоцитов
tspan = [0, 250] # время
# эксперименты
y0 = [0.1, 9] # первое число лимфоциты, второе - опухоль
T1, Y1 = rk3(f_main, tspan, y0, N)
y0 = [0.5, 9] # первое число лимфоциты, второе - опухоль
T2, Y2 = rk3(f_main, tspan, y0, N)
# график
fig, ax = plt.subplots(figsize=(16,9))
ax.plot(T1, Y1[:, 1], color='xkcd:red', label=f'лимфоциты 0.1')
ax.plot(T2, Y2[:, 1], color='xkcd:black', label=f'лимфоциты 0.5')
ax.set_xlabel('Время')
ax.set_ylabel("Опухоль")
ax.set_title(f'График зависимости размера опухоли от времени при разном
количество лимфоцитов$')
ax.legend()
plt.savefig('19.png')
plt.show()
```