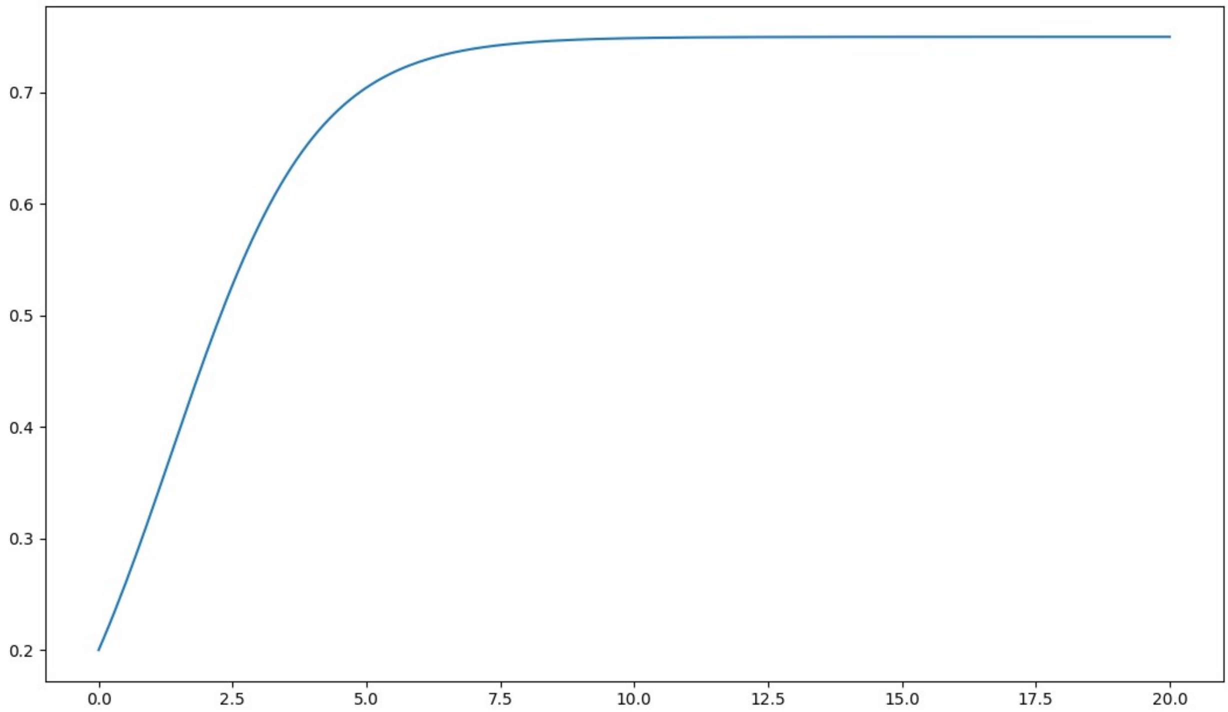
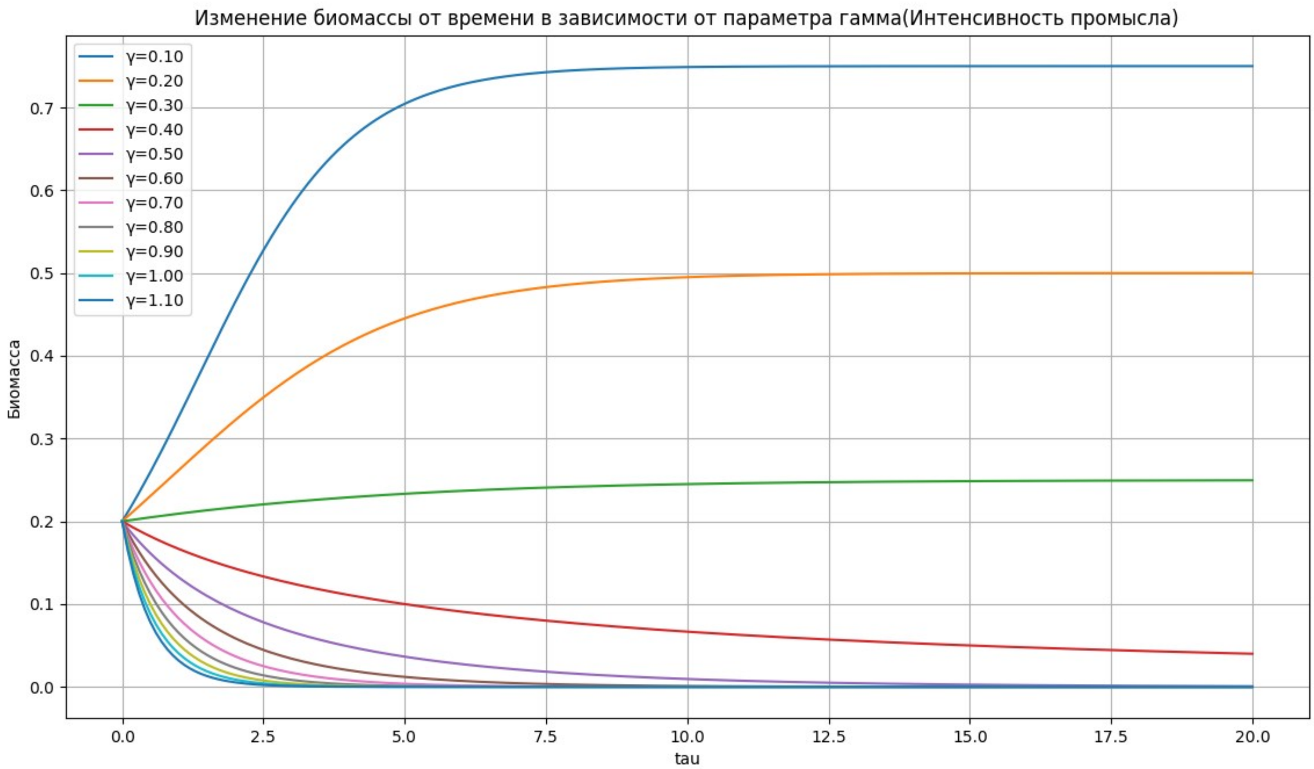
ЛАБОРАТОРНАЯ РАБОТА №4
Численное решение задачи Коши
Чудов Богдан А-17-22, Вариант 16

Задача 4.3. Задача Коши для ОДУ 1 порядка следующего вида
$x' = \frac{\alpha - x}{\alpha} \beta x - \gamma x, t > 0$
$x(0) = x0$
описывает изменение биомассы x(t) любого промыслового вида рыбы в океане. Здесь α - плотность насыщения (задающая максимально возможное количество биомассы данного вида), β - удельная скорость роста биомассы при x ≈ 0, γ - постоянная, характеризующая интенсивность промысла.
A) Промоделировать процесс изменения биомассы в зависимости от интенсивности промысла.
B) Определить, при какой интенсивности количество выловленной за время T=50 рыбы $V = \gamma \int_0^T x(t)\,dt$является наибольшим. Определить диапазон хищнического лова (т.е. значения интенсивности промысла, при которых вид полностью исчезает).
Вариант 16. Заданные параметры:
β = 0.4
γ = [0.1, 1.1]
y0 = 0.2
Метод решения задачи Коши — метод Рунге-Кутты 3 порядка (вариант III)
Метод вычисления интеграла — центральных прямоугольников с уточнением по Рунге
Решение задачи.
1. Промасштабируем исходную задачу введя новые переменные
$y = \frac{x}{\alpha}, \tau = \beta t$, получим задачу $y' = y(1-y) - py, y(0) = y0$, где $p = \frac{\gamma}{\beta}, y0 = \frac{x0}{\alpha}$
Изменим отрезок на котором решаем задачу из-за перехода к новой переменной
$t = [0{,}50], \tau = \beta t \Rightarrow \tau = [0{,}50\beta] \Rightarrow \tau = [0{,}20]$
2. Решим задачу при минимальном значении γ = 0.1 с шагом h = 0.1 методом Рунге-Кутты 3 порядка точности (код приведен в приложении 1), полученный график приведен ниже. По нему можно видеть что численность популяции становится вдвое больше при τ ≈ 2, а стабилизируется при τ ≈ 8. Следовательно популяция увеличивается вдвое в момент времени t=2/0.4=5 , а стабилизируется в момент времени t=8 /0.4 =20.



3. Решим эту же задачу но при изменении параметра γ от 0.1 до 1.1 с шагом 0.1. Построим графики. Код решения также приведен в приложении 1.

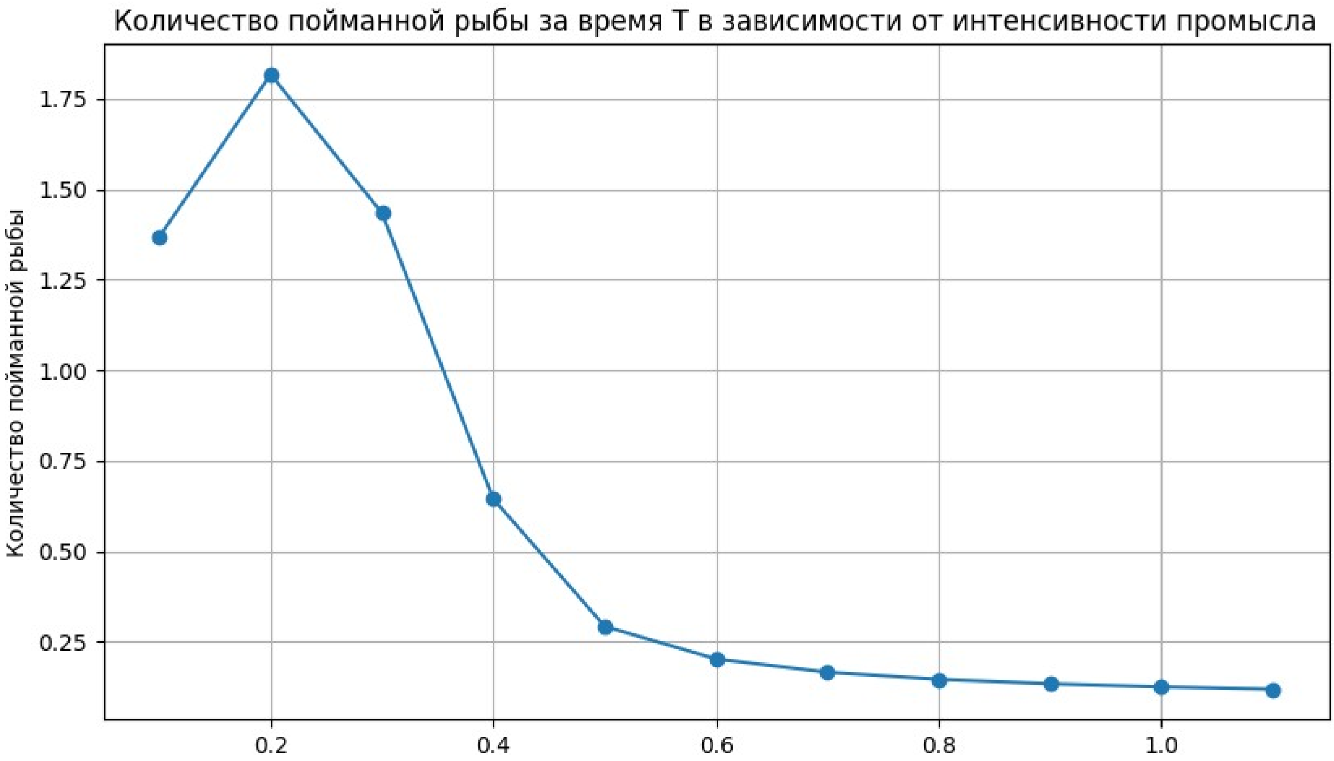


По графику можно видеть, что полное исчезновение популяции происходит при $\gamma \geq 0.5$

4. Возьмем вдвое больший шаг для решения задачи Коши при каждом γ и вычислим погрешность по правилу Рунге. Полученная погрешность:

ε=3.4347500994386894e-05

5. Для каждого полученного решения вычислим интеграл $V=\gamma\int\limits_0^{\beta t} y(\tau)\,d\tau$, используя квадратурную формулу центральных прямоугольников с уточнением по Рунге. Построим график V(γ) и определим при каком γ значение интеграла максимально. Полученный график представлен ниже.



Можно видеть, что наибольшего количества пойманной рыбы можно добиться при интенсивности промысла γ ≈ 0.2

6. Найдем с 2 верными цифрами значение γ при котором улов максимален. Для этого можно вычислить значение интеграла для каждого γ в промежутке от 0.1 до 0.3 с шагом 0.01. Получившееся значение с двумя верными цифрами: γ ≈ 0.20

Приложение 1

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

# параметры
beta = 0.4
gamma_values = np.linspace(0.1, 1.1, 11)
y0 = 0.2
T = 20
h = 0.1
time_steps = int(T / h)

# метод рунге-кутты 3 порядка
def rk3(y0, p, h, steps):
    y = np.zeros(steps + 1)
    y[0] = y0
    for i in range(steps):
        ti = i * h
        yi = y[i]
        k1 = h * (yi * (1 - yi) - p * yi)
        k2 = h * ((yi + 0.5 * k1) * (1 - (yi + 0.5 * k1)) - p * (yi + 0.5 * k1))
         k3 = h * ((yi + 0.75 * k2) * (1 - (yi + 0.75 * k2)) - p * (yi + 0.75 *
k2))
        y[i + 1] = yi + (1 / 9) * (2 * k1 + 3 * k2 + 4 * k3)
    return y

# вычисление интеграла центральными прямоугольниками с уточнением по Рунге
def integr(yh, yq, h):
    I_h = 0
    I_h2 = 0
    for i in range(1, len(yh), 2):
        I_h += yh[i]
    I_h *= h
    for i in range(1, len(yq), 2):
        I_h2 += yq[i]
    I_h2 *= h / 2
    I_runge = I_h + (I_h - I_h2) / 3

    return I_runge

# вычисление результатов и построение графиков
# решение при минимальном gamma
p = gamma_values[0] / beta
y = rk3(y0, p, h, time_steps)
t = np.arange(0, T + h, h)
plt.plot(t, y, label=f'γ={gamma_values[0]:.2f}')
plt.show()

# решение при всех gamma, вычисление интегралов и погрешности по правилу Рунге
V_values = []
tol = 0


t = np.arange(0, T + h, h)
for gamma in gamma_values:
    p = gamma / beta
    y2 = rk3(y0, p, h*2, time_steps // 2)
    y = rk3(y0, p, h, time_steps)
    for i in range(0, time_steps // 2 + 1):
        if abs(y2[i] - y[i*2]) > tol:
            tol = abs(y2[i] - y[i*2])
    yh = rk3(y0, p, h / 2, time_steps * 2)
```

```
    yq = rk3(y0, p, h / 4, time_steps * 4)
    V = gamma * integr(yh, yq, h)
    V_values.append(V)

    plt.plot(t, y, label=f'γ={gamma:.2f}')
tol = tol / 7

# вычисление gamma при котором V максимально
vmax = 0
gmax = 0
gmax_values = np.arange(0.1, 0.31, 0.01)
for gamma in gmax_values:
    p = gamma / beta
    y2 = rk3(y0, p, h*2, time_steps // 2)
    y = rk3(y0, p, h, time_steps)
    V = gamma * integr(y2, y, h)
    print(f'gamma = {gamma:.2f} V = {V}')
    if V > vmax:
        vmax = V
        gmax = gamma

# вывод результатов
plt.xlabel('tau')
plt.ylabel('Биомасса')
plt.legend()
plt.title('Изменение биомассы от времени в зависимости от параметра гамма(Интенсивность промысла)')
plt.grid(True)
plt.show()

print(f'Погрешность по правилу Рунге при вычислении решения задачи Коши: {tol}')
print(f'Интенсивность промысла при которой количество выловленной выбы максимально: {gmax:.2f}')

plt.plot(gamma_values, V_values, marker='o')
plt.xlabel('')
plt.ylabel('Количество пойманной рыбы')
plt.title('Количество пойманной рыбы за время T в зависимости от интенсивности промысла')
plt.grid(True)
plt.show()
```