МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

имени М.В.Ломоносова

Факультет вычислительной математики и кибернетики

Компьютерный практикум по учебному курсу

«ВВЕДЕНИЕ В ЧИСЛЕННЫЕ МЕТОДЫ»

ЗАДАНИЕ № 2.

Численные методы решения дифференциальных уравнений

ОТЧЕТ

о выполненном задании

студента 205 учебной группы факультета ВМК МГУ

Фазылова Рамазана Рамилевича

гор. Москва

2020 г.

Содержание

Решение задачи Коши для систем ОДУ первого порядка

Постановка задачи

Рассматривается обыкновенное дифференциальное уравнение первого порядка, разрешенное относительно производной и имеющее вид:

$$\frac{dy}{dx} = f(x, y),\ x > x_0,$$

с дополнительным начальным условием, заданным в точке $x = x_0$:

$$y(x_0) = y_0$$

Преполагается, что функция $f = f(x, y)$ такова, что гарантирует существование и единственность решения задачи Коши.

Для системы ОДУ первого порядка, разрешенных относительно производных неизвестных функций, соотвествующая задача Коши имеет следующий вид:

$$\begin{cases} \dfrac{dy_1}{dx} = f_1(x, y_1, y_2), \\ \dfrac{dy_2}{dx} = f(x, y_1, y_2), \\ y_1(x_0) = y_1^{(0)}, \\ y_2(x_0) = y_2^{(0)}, \\ x > x_0. \end{cases}$$

Также предполагается, что функции f_1, f_2 заданы так, что гарантируется существование и единственность решения задачи Коши для данной системы ОДУ.

Цели и задачи практической работы

1. Решить задачу Коши методом Рунге-Кутта второго и четвертого порядка точности, аппроксимировав дифференциальную задачу соответствующей разностной схемой на равномерной сетке. Полученное конечно-разностное уравнение (или уравнения) просчитать численно

2. Найти численное решение задачи и построить его график

3. Найденное численное решение сравнить с точным решением дифференциального уравнения

Описание алгоритмов решения

Метод Рунге-Кутты 2го порядка

Для того, чтобы получить значения функции решения задачи Коши на отрезке $[a, b]$, разобьем его на n равномерных отрезков. Таким образом, получим следующее разбиение:

$$h = \frac{b-a}{n}$$
$$x_i = a + h * i$$
$$y_i = y(x_i)$$

Воспользуемся следующим рекуррентым соотношением для ОДУ 1го порядка:

$$y_{i+1} = y_i + \frac{f(x_i, y_i) + f(x_{i+1}, \widetilde{y}_{i+1})}{2} h,$$
$$\text{где } \widetilde{y}_{i+1} = y_i + f(x_i, y_i)h$$

Для системы ОДУ 1го порядка:

$$y_{1,i+1} = y_{1i} + \frac{f_1(x_i, y_{1i}, y_{2i}) + f_1(x_{i+1}, y_{1i} + f_1(x_i, y_{1i}, y_{2i})h, y_{2i} + f_2(x_i, y_{1i}, y_{2i})h)}{2} h,$$
$$y_{2,i+1} = y_{2i} + \frac{f_2(x_i, y_{1i}, y_{2i}) + f_2(x_{i+1}, y_{1i} + f_1(x_i, y_{1i}, y_{2i})h, y_{2i} + f_2(x_i, y_{1i}, y_{2i})h)}{2} h$$

Метод Рунге-Кутты 4го порядка

В методе Рунге-Кутты 4го порядка точности соотношение для ОДУ 1го порядка будет следующим:

$$y_{i+1} = y_i + \frac{h(k_1 + 2k_2 + 2k_3 + k_4}{6}, \text{ где}$$

$$k_1 = f(x_i, y_i)$$

$$k_2 = f(x_i + \frac{h}{2}, y_i + \frac{h}{2}k_1)$$

$$k_3 = f(x_i + \frac{h}{2}, y_i + \frac{h}{2}k_2)$$

$$k_4 = f(x_i + h, y_i + hk_3)$$

Для системы ОДУ 1го порядка будут справедливы следующие соотношения:

$$\begin{cases} k_{j1} = f_j(x_i, y_{1,i}, y_{2,i}) \\ k_{j2} = f_j(x_i + \frac{h}{2}, y_{1,i} + \frac{h}{2}k_{11}, y_{2,i} + \frac{h}{2}k_{21}) \\ k_{j3} = f_j(x_i + \frac{h}{2}, y_{1,i} + \frac{h}{2}k_{12}, y_{2,i} + \frac{h}{2}k_{22}) \qquad j = 1, 2 \\ k_{j4} = f_j(x_i + h, y_{1,i} + hk_{13}, y_{2,i} + hk_{23}) \\ y_{j,i+1} = y_{j,i} + \frac{k_{j,1} + 2k_{j,2} + 2k_{j,3} + k_{j,4}}{6}h \end{cases}$$

Описание программы

Вспомогательные функции

```
# некоторые элементарные функции
from math import exp, sqrt, tan, atan, cos, sin, log
# библиотека для вывода графиков функции
import matplotlib.pyplot as plt

# разбить [start, stop) на n точек
def arange(start, stop, n):
    step = (stop - start) / n
    return [round(x*step, 10) for x in range(int(start/step), int(stop/step))]

# нарисовать график функции
def draw(X, Y, marker='', cl=None, lw=1):
    plt.plot(X, Y, marker, color=cl, linewidth = lw)

# правая часть дифф. уравнения (таблица 1-3)
def f(x, y):
    return -y-x**2

# точное решение дифф. уравнения
def ans1(x):
    return -x**2+2*x-2+12*exp(-x)

# функции, опсывающие систему ОДУ 1го порядка(таблица 2-12)
def f1(x, u, v):
    return -2*x*u**2+v**2-x-1

def f2(x, u, v):
    return 1/(v**2)-u-x/u
```

Реализация метода Рунге-Кутты 2го порядка

```
# метод Рунге-Кутты 2го порядка для обыкновенного дифф. уравнения
def runge_2(f, a, b, y0, n):
    h = (b - a)/n # диаметр разбиения
    x = a # x0
    y = y0 # начальное условие
    X = [x] # вектор x, сетка
    Y = [y] # вектор y, сеточная функция
    # рекуррентное вычисление функции
    for i in range(0, n):
        x = a + h*i
        y = y + h/2*(f(x,y)+f(x+h, y+f(x,y)*h))
        X.append(x+h)
        Y.append(y)
    return X, Y

# метод Рунге-Кутты 2го порядка для системы ОДУ
def runge_2s(f1, f2, a, b, y01, y02, n):
    h = (b - a)/n # диаметр разбиения
    x = a # x0
    y1 = y01 # начальные условия
    y2 = y02 # начальные условия
    Y1 = [y1] # сеточная функция y1
    Y2 = [y2] # сеточная функция y2
    X = [x] # сетка
    # реализация рекуррентного соотношения
    for i in range(0, n):
        x = a + h*i
        y1_n = y1 + h/2*(f1(x, y1, y2)+f1(x+h, y1+f1(x,y1,y2)*h, y2+f2(x,y1,y2)*h))
        y2_n = y2 + h/2*(f2(x, y1, y2)+f2(x+h, y1+f1(x,y1,y2)*h, y2+f2(x,y1,y2)*h))
        y1 = y1_n
        y2 = y2_n
        X.append(x+h)
        Y1.append(y1)
        Y2.append(y2)
    return X, Y1, Y2
```

Реализация метода Рунге-Кутты 4го порядка

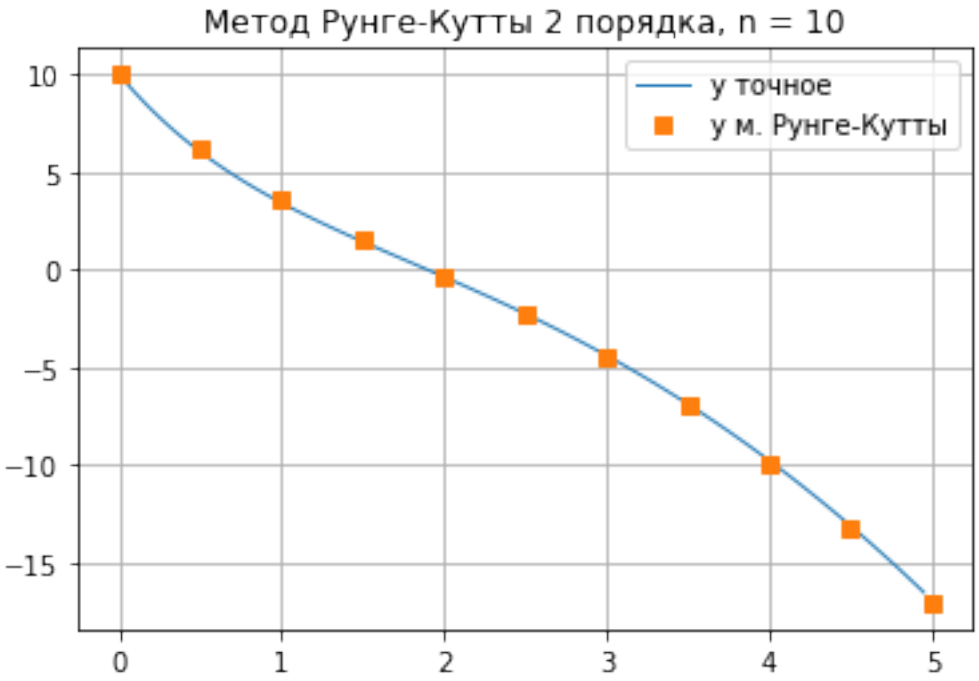
```
# метод Рунге-Кутты 4го порядка для обыкновенного дифф. уравнения
def runge_4(f, a, b, y0, n):
    h = (b - a)/n # диаметр
    x = a
    y = y0 # начальное условие
    Y = [y] # сеточная функция
    X = [x] # сетка
    # рекурентное вычисление функции
    for i in range(0, n):
        x = a + h*i
        k1 = f(x, y)
        k2 = f(x+h/2,y+h/2*k1)
        k3 = f(x+h/2,y+h/2*k2)
        k4 = f(x+h,y+h*k3)
        y = y + h/6*(k1+2*k2+2*k3+k4)
        X.append(x+h)
        Y.append(y)
    return X, Y

# метод Рунге-Кутты 4го порядка для системы ОДУ
def runge_4s(f1, f2, a, b, y01, y02, n):
    h = (b - a)/n # диаметр разбиения
    y1 = y01 # начальные условия
    y2 = y02
    Y1 = [y1] # сеточная функция y1
    Y2 = [y2] # сеточная функция y2
    X = [a] # сетка
    for i in range(0, n-1):
        x = a + h*i
        k11 = f1(x, y1, y2)
        k12 = f2(x, y1, y2)
        k21 = f1(x+h/2, y1+h/2*k11, y2+h/2*k12)
        k22 = f2(x+h/2, y1+h/2*k11, y2+h/2*k12)
        k31 = f1(x+h/2, y1+h/2*k21, y2+h/2*k22)
        k32 = f2(x+h/2, y1+h/2*k21, y2+h/2*k22)
        k41 = f1(x+h, y1+h*k31, y2+h*k32)
        k42 = f2(x+h, y1+h*k31, y2+h*k32)
        y1 = y1 + h/6*(k11+2*k21+2*k31+k41)
        y2 = y2 + h/6*(k12+2*k22+2*k32+k42)
        X.append(x+h)
        Y1.append(y1)
        Y2.append(y2)
    return X, Y1, Y2
```

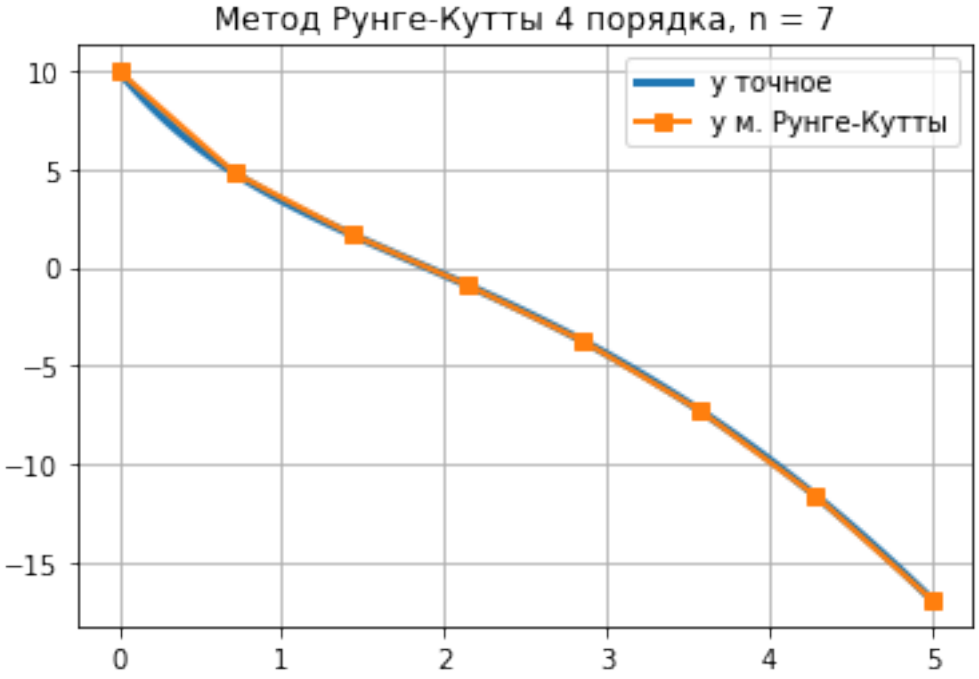
Тестирование программы

1. Таблица 1-3

$$\begin{cases} \dfrac{dy}{dx} = -y - x^2, \\ y(0) = 10 \end{cases}$$

Точное решение: $y = -x^2 + 2x - 2 + 12e^{-x}$

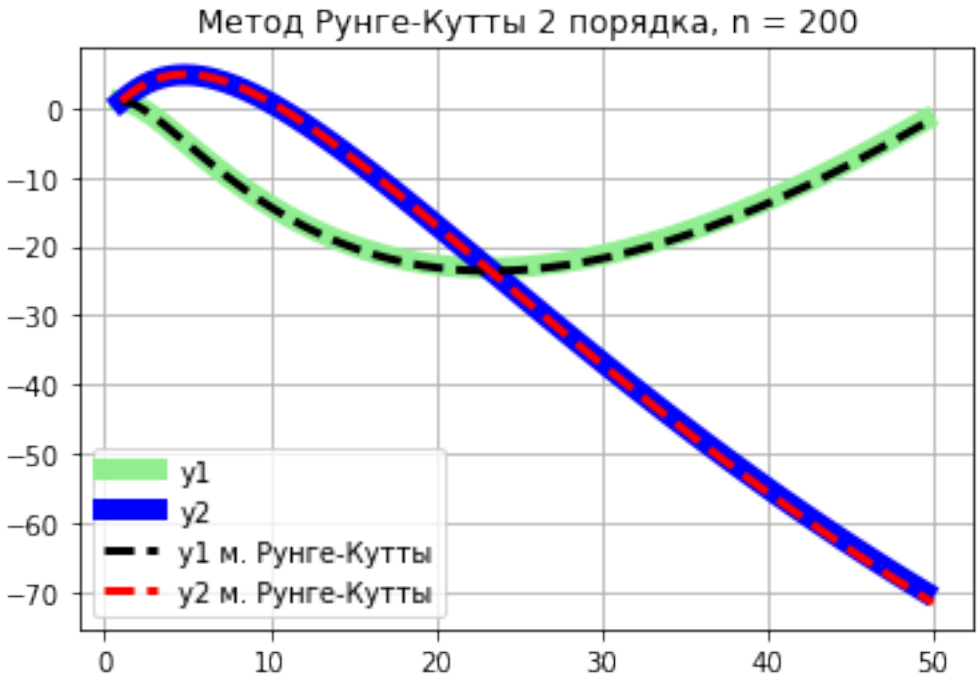


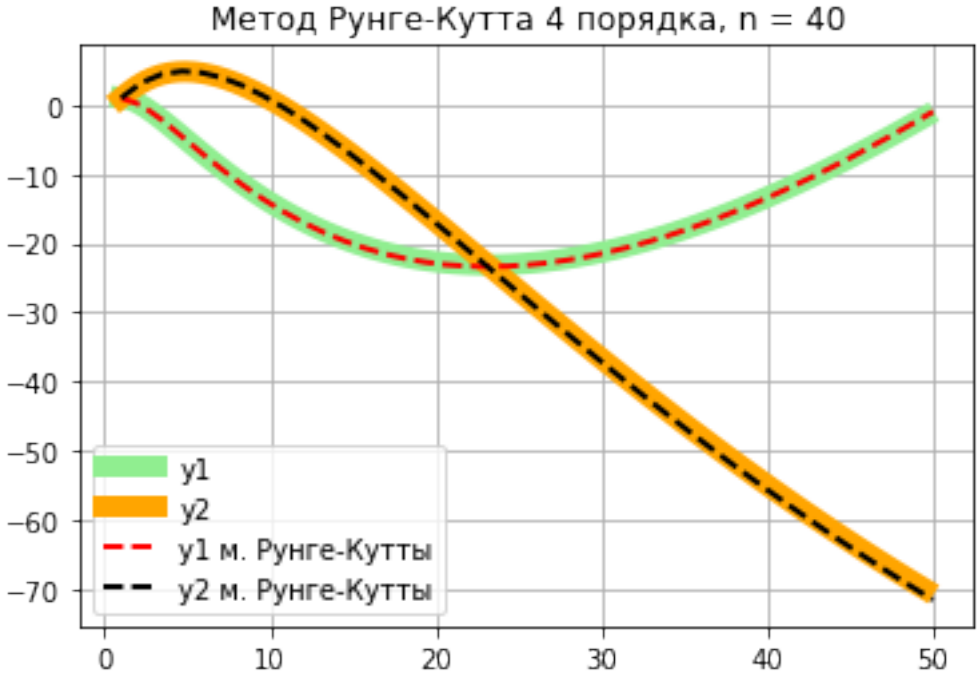


2. Таблица 2-1

$$\begin{cases} \dfrac{du}{dx} = \dfrac{u-v}{x}, & u(1) = 1 \\ \dfrac{dv}{dx} = \dfrac{u+v}{x}, & v(1) = 1 \end{cases}$$

Точное решение: $\begin{cases} u = x(\cos(\log(x)) - \sin(\log(x))) \\ v = x(\cos(\log(x)) + \sin(\log(x))) \end{cases}$

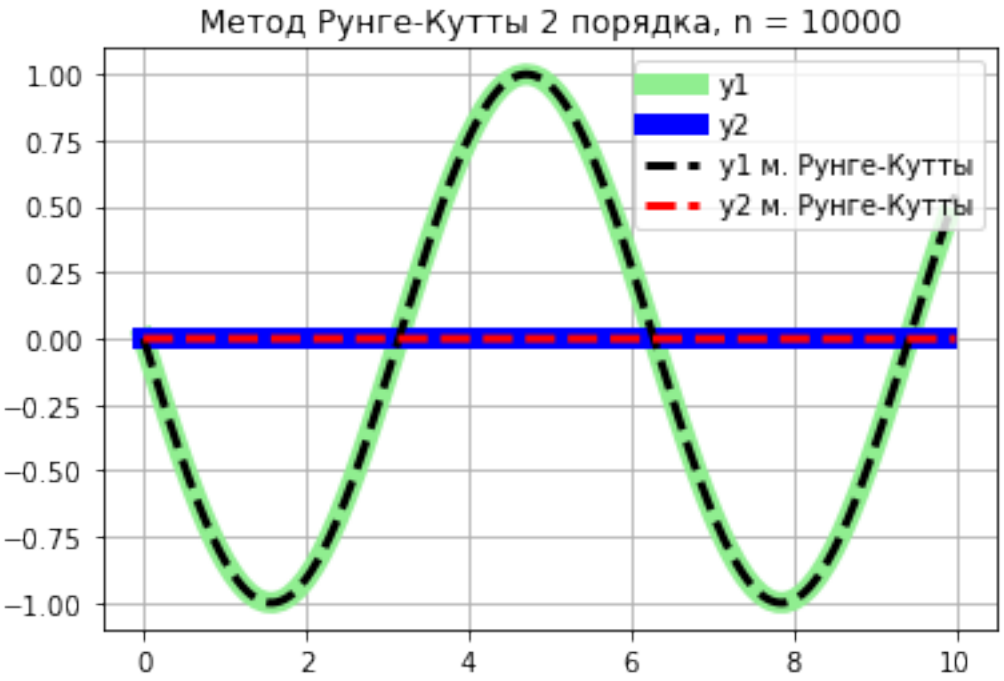


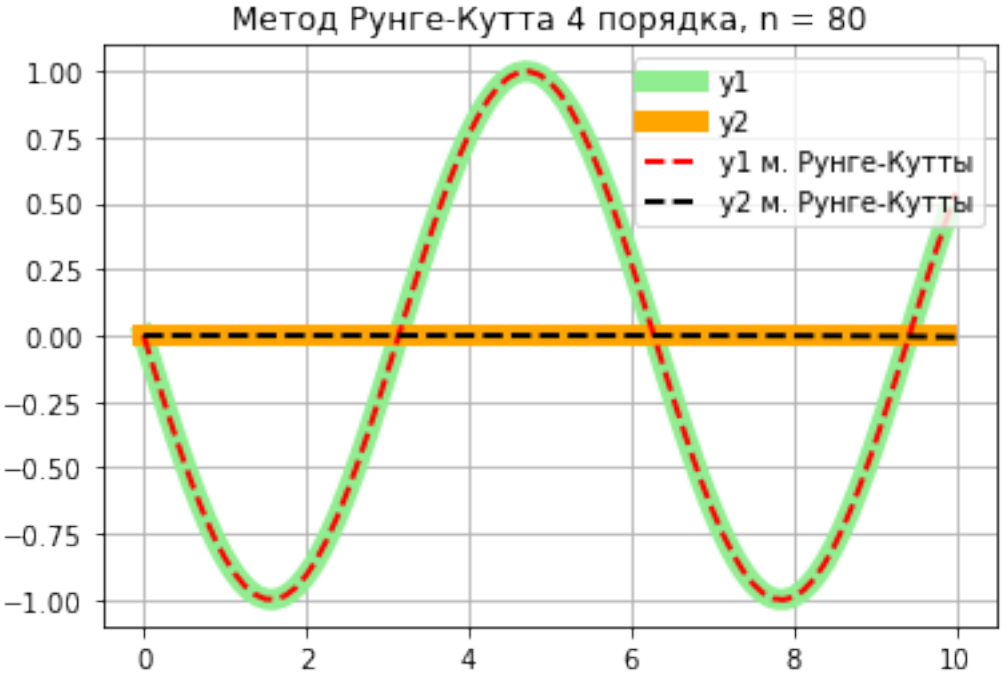


3. Таблица 2-10

$$\begin{cases} \dfrac{du}{dx} = v - \cos(x), \quad u(0) = 0 \\ \dfrac{dv}{dx} = u + \sin(x), \quad v(0) = 0 \end{cases}$$

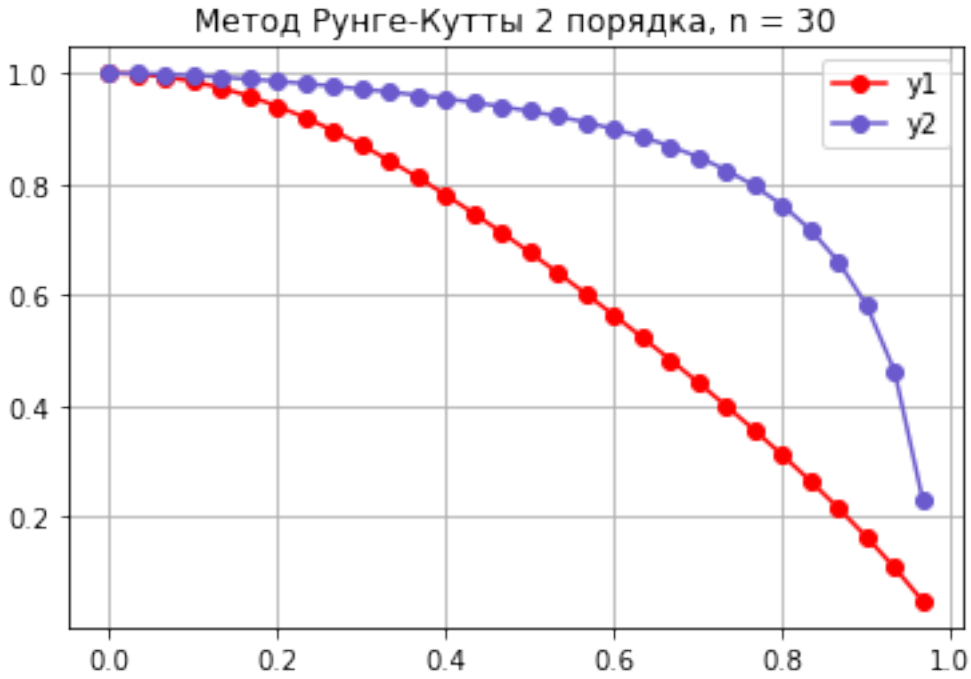
Точное решение: $\begin{cases} u = -\sin(x) \\ v = 0 \end{cases}$

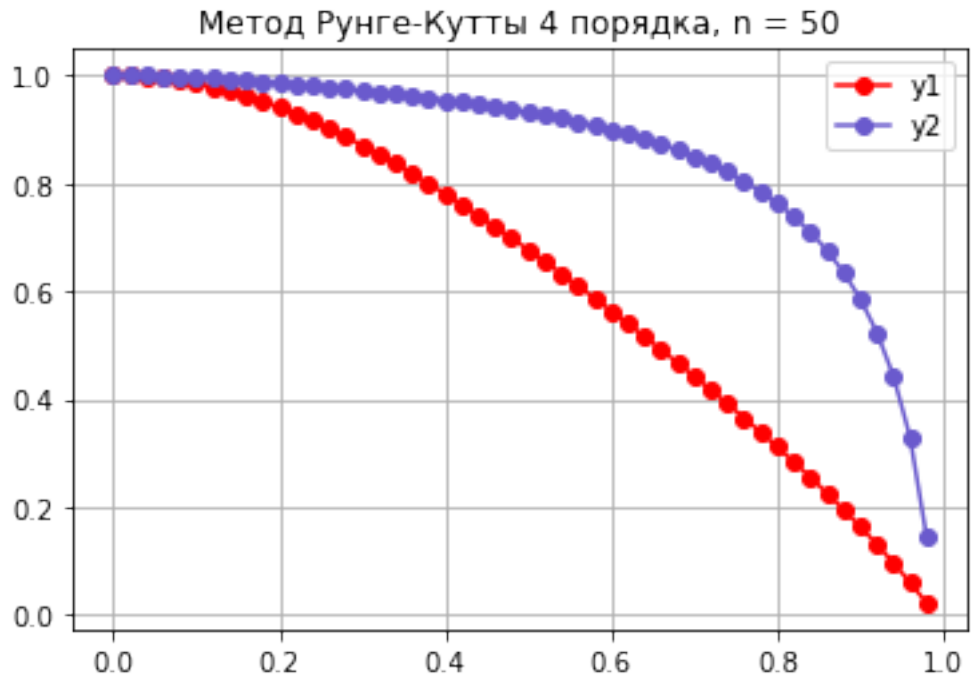




4. Таблица 2-12

$$\begin{cases} \dfrac{du}{dx} = -2xy^2 - x - 1, \quad u(0) = 1 \\ \dfrac{dv}{dx} = \dfrac{1}{v^2} - u - \dfrac{x}{u}, \quad v(0) = 1 \end{cases}$$





Решение краевой задачи для ОДУ 2го порядка

Постановка задачи

Рассматривается линейное дифференциальное уравнение второго порядка вида:

$$y'' + p(x) \cdot y' + q(x) \cdot y = -f(x),\ b < x < a$$

с дополнительными условиями в граничных точках:

$$\begin{cases} \sigma_1 y(a) + \gamma_1 y'(a) = \delta_1, \\ \sigma_2 y(b) + \gamma_2 y'(b) = \delta_2 \end{cases}$$

Цели и задачи практической работы

1. Решить кравевую задачу методом конечных разностей, аппроксимимровав её разностной схемой второго порядка точности на равнморной сетке. Полученную систему конечно-разностных уравнений решить методом прогонки

2. Найти разностное решение задачи и построить его график

3. Найденное разностное решение сравнить с точным решением дифференциального уравнения

Описание алгоритмов решения

Используя разностные приближения первой и второй производной

$$\begin{cases} y_i' = \dfrac{y_{i+1} - y_{i-1}}{2h}, \\ y_i'' = \dfrac{y_{i+1} - 2y_i + y_{i-1}}{h^2} \end{cases}$$

заменим исходное дифференциальное уравнение его разностным аналогом, при этом сгруппируем коэффициенты при слагаемых y_{i-1}, y_i, y_{i+1}. Получим следующую систему:

$$\begin{cases} y_0 \left(\sigma_1 - \dfrac{\gamma_1}{h}\right) + y_1 \dfrac{\gamma_1}{h} = \delta_1 \\ y_{i-1} \left(1 - \dfrac{p_i h}{2}\right) + y_i \left(-2 + q_i h^2\right) + y_{i+1} \left(1 + \dfrac{p_i h}{2}\right) = f_i h^2 \,, \qquad i = \overline{1, n-1} \\ y_{n-1} \dfrac{\gamma_2}{h} + y_n \left(\sigma_2 + \dfrac{\gamma_2}{h}\right) = \delta_2 \end{cases}$$

Заметим, что данная система является трехдиагональной. Заменим коэффициенты матрицы следующим образом:

$$\begin{cases} C_0 y_0 + B_0 y_1 = F_0 \\ A_i y_{i-1} + B_i y_i + C_i y_{i+1} = F_i \,, \qquad i = \overline{1, n-1} \\ A_n y_{n-1} + C_n y_n = F_n \end{cases}$$

Полученную систему решим методом прогонки.
Сначала найдем прогоночные коэффициенты:

$$\begin{cases} \alpha_0 = \dfrac{-B_0}{C_0} \\ \beta_0 = \dfrac{F_0}{C_0} \\ \alpha_i = \dfrac{-B_i}{A_i \alpha_{i-1} + C_i} \\ \beta_i = \dfrac{F_i - A_i \beta_{i-1}}{A_i \alpha_{i-1} + C_i} \end{cases} \qquad i = \overline{1, n-1}$$

Затем вычислим корни уравнения, используя следующее рекуррентное соотношение:

$$\begin{cases} y_n = \dfrac{F_n - A_n \beta_{n-1}}{A_n \alpha_{n-1} + C_n} \\ y_i = \alpha_i y_{i+1} + \beta_i, \quad i = \overline{0, n-1} \end{cases}$$

Описание программы

```python
# функция формирует и решает трехдиагональную систему краевой задачи
def boundary(p, q, f, sigma, gamma, delta, n, a = 1, b = 1.3):
    h = (b-a)/n
    def A_func(x):
        return 1 - p(x)*h/2
    def C_func(x):
        return -2 + q(x)*(h**2)
    def B_func(x):
        return 1 + p(x)*h/2
    def F_func(x):
        return f(x)*(h**2)
    # формирование коэффициентов трехдиагональной матрицы
    A = [A_func(a+i*h) for i in range(0, n+1)]
    B = [B_func(a+i*h) for i in range(0, n+1)]
    C = [C_func(a+i*h) for i in range(0, n+1)]
    F = [F_func(a+i*h) for i in range(0, n+1)]
    # задание краевых условий
    F[0] = delta[0]
    F[n] = delta[1]
    C[0] = sigma[0]-gamma[0]/h
    B[0] = gamma[0]/h
    A[n] = -gamma[1]/h
    C[n] = sigma[1]+gamma[1]/h
    # инициализация прогоночных коэффициентов
    alpha = [0]*n
    beta = [0]*n
    alpha[0] = -B[0]/C[0]
    beta[0] = F[0]/C[0]

    # рекуррентное вычисление всех прогоночных коэффициентов
    for i in range(1, n):
        alpha[i] = -B[i]/(A[i]*alpha[i-1]+C[i])
        beta[i] = (F[i]-A[i]*beta[i-1])/(C[i]+A[i]*alpha[i-1])

    # инициализация вектора неизвестных
    y = [0]*(n+1)
    y[n] = (F[n]-A[n]*beta[n-1])/(C[n]+A[n]*alpha[n-1])

    # рекуррентное вычисление неизвестных
    for i in reversed(range(0, n)):
        y[i] = alpha[i]*y[i+1]+beta[i]

    return y
```

Функции, описывающие дифференциальное уравнение варианта 9

$$y'' - \frac{y'}{2} + 3y = 2x^2$$

И его точное решение для заданных начальных условий, полученное с помощью WolframAlpha:

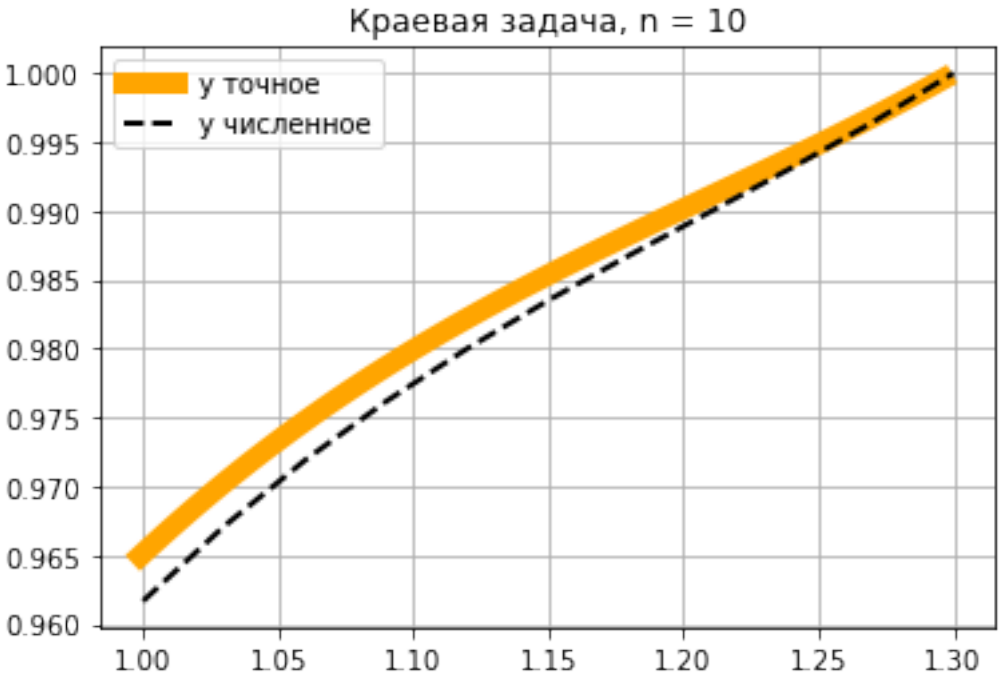
$$C_1 e^{\frac{x}{4}} \sin\frac{\sqrt{47}x}{4} + C_2 e^{\frac{x}{4}} \cos\frac{\sqrt{47}x}{4} + \frac{2x^2}{3} + \frac{2x}{9} + \frac{11}{27}, \text{ где}$$

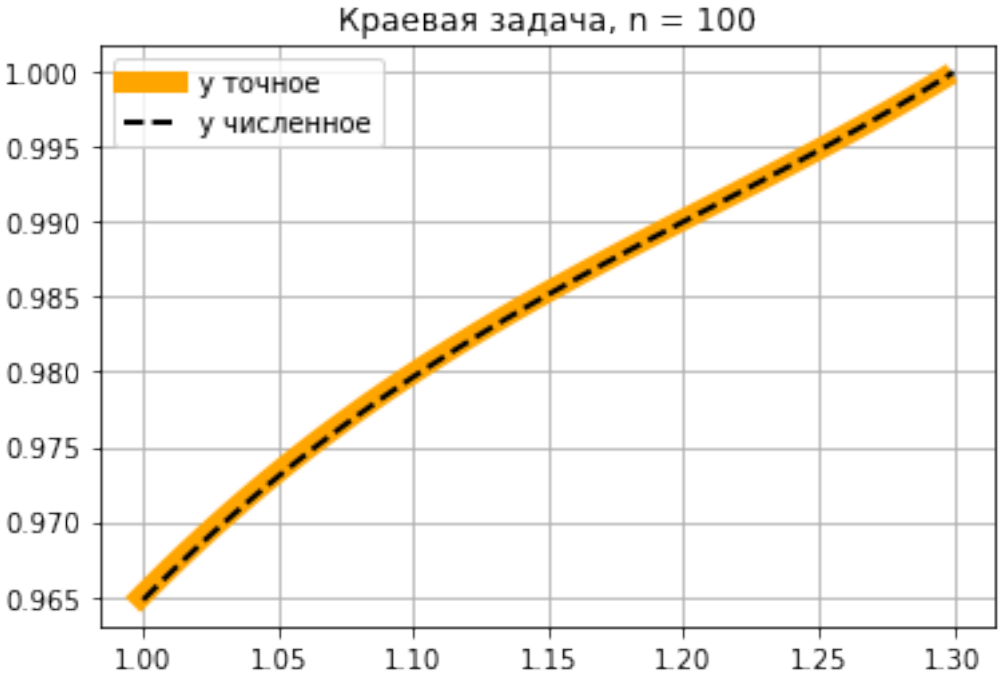
$$C_1 = 0.469645$$
$$C_2 = 0.618165$$

```
# точное решение данного дифференциального решения
def ans(x):
    c_1 = 0.469645
    c_2 = 0.618165
    y = c_1*exp(x/4)*sin((sqrt(47)*x)/4) +
        c_2*exp(x/4)*cos((sqrt(47)*x)/4) +
        (2*(x**2))/3 + (2*x)/9 - 11/27
    return y

# функции, описывающие данное дифференциальное уравнение
def p(x):
    return -1/2
def q(x):
    return 3
def f(x):
    return 2*(x**2)
```

Тестирование программы

1. Вариант 9



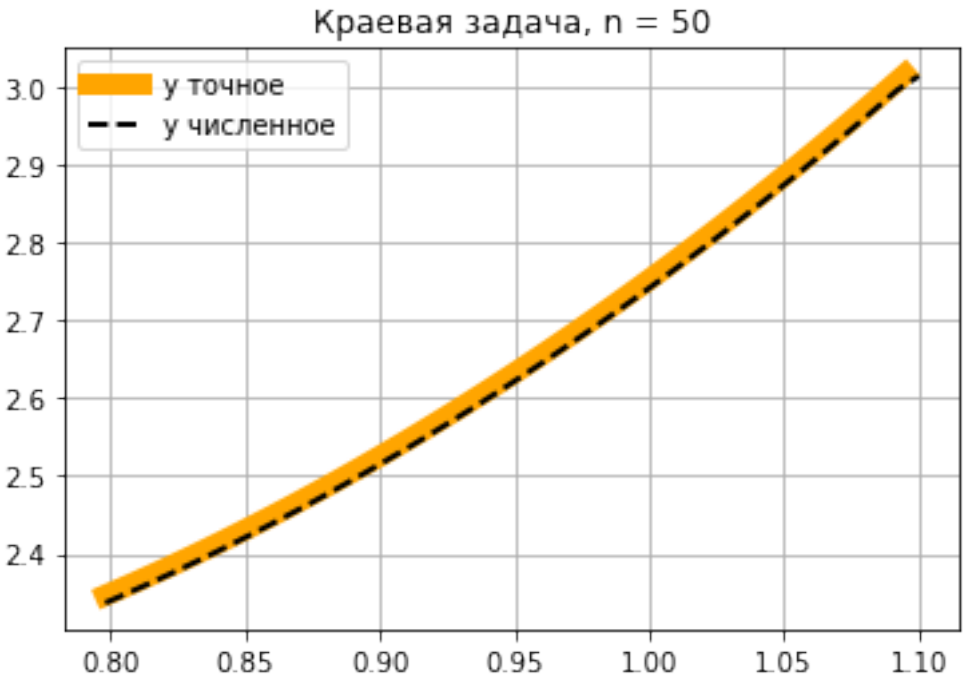


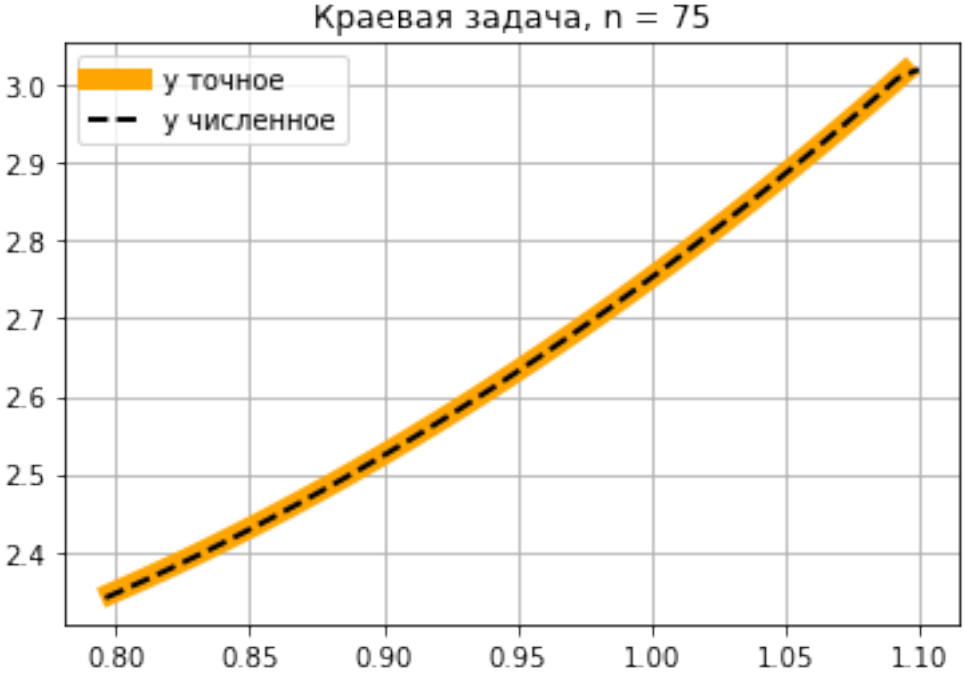
2. Вариант 13

$$\begin{cases} y'' + \dfrac{2y'}{x} - 3y = 2 \\ y'(0.8) = 1.5 \\ 2y(1.1) - y'(1.1) = 3 \end{cases}$$

Точное решение:

$$y = \frac{-0.0426075\exp(-\sqrt{3}x)}{x} + \frac{0.606253\exp(\sqrt{3}x)}{x} - 0.666667$$





3. Дифференциальное уравнение:

$$\begin{cases} y'' + y' = 1 \\ y'(0) = 0 \\ y(1) = 1 \end{cases}$$

Точное решение: $y = x + e^{-x} - \frac{1}{e}$