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 219 учебной группы факультета ВМК МГУ

Ратникова Тимофея Георгиевича

гор. Москва

2022 г.

Решение задачи Коши для систем ОДУ первого порядка

Постановка задачи

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

$$\frac{dy}{dx} = f(x, y),\ x_0 < x, \qquad (1)$$

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

$$y(x_0) = y_0. \qquad (2)$$

Предполагается, что правая часть уравнения (1) функция $f = f(x, y)$ такова, что гарантирует существование и единственность решения задачи Коши (1)-(2).

В том случае, если рассматривается не одно дифференциальное уравнение вида (1), а система обыкновенных дифференциальных уравнений первого порядка, разрешенных относительно производных неизвестных функций, то соответствующая задача Коши имеет вид (на примере двух дифференциальных уравнений):

$$\begin{cases} \dfrac{dy_1}{dx} = f_1(x, y_1, y_2), \\ \dfrac{dy_2}{dx} = f_2(x, y_1, y_2),\ x > x_0. \end{cases} \qquad (3)$$

Дополнительные (начальные) условия задаются в точке $x = x_0$:

$$y_1(x_0) = y_1^{(0)},\ y_2(x_0) = y_2^{(0)}. \qquad (4)$$

Также предполагается, что правые части уравнений из (3) заданы так, что это гарантирует существование и единственность решения задачи Коши (3)-(4), но уже для системы обыкновенных дифференциальных уравнений первого порядка в форме, разрешенной относительно производных неизвестных функций.

Цели и задачи практической работы

1) Решить задачу Коши (1)-(2) (или (3)-(4)) наиболее известными и широко используемыми на практике методами Рунге-Кутта второго и четвертого порядка точности, аппроксимировав дифференциальную задачу соответствующей разностной схемой (на равномерной сетке); полученное конечно-разностное уравнение (или уравнения в случае системы), представляющее фактически некоторую рекуррентную формулу, просчитать численно;
2) Найти численное решение задачи и построить его график;
3) Найденное численное решение сравнить с точным решением дифференциального уравнения.

Алгоритмы решения

Метод Рунге-Кутты 2-го порядка

Пусть имеем точку принадлежащую искомому решению. Для того, чтобы найти следующую точку проведем касательную к кривой в точке (x_m, y_m) До пересечения с прямой $x = x_{m+1/2}$ где $x_{m+1/2} = x_{m+h/2}$. Тогда, получим координату (по формуле Эйлера)

$$y_{m+1/2} = y_m + \frac{h}{2} f(x_m, y_m).$$

Таким образом

$$y_{m+\frac{1}{2}} = y_m + \frac{h}{2} * f(x_m, y_m), x_{m+1/2} = x_m + h/2$$

$$y_{m+1} = y_m + h * f\left(x_{m+1/2}, y_{m+1/2}\right), x_{m+1} = x_m + h$$

для системы дифференциальных уравнений

$$y_i' = f_i(x, y_1, y_2, \dots y_k), y_i(x_0) = y_1^2, i = 1,2,..k;$$

расчетные формулы имеют вид:

$$y_i^{m+\frac{1}{2}} = y_i^m + \frac{h}{2} * f_i(x_m, y_1^m, y_2^m, \dots y_k^m), x_{m+\frac{1}{2}} = x_m + \frac{h}{2};$$

$$y_i^{m+1} = y_i^m + h * f_i\left(x_{m+1/2}, y_1^{m+1/2}, y_2^{m+1/2}, \dots y_k^{m+1/2}\right)$$

Метод Рунге-Кутты 4-го порядка

Для одиночного дифференциального уравнения $y' = f(x, y), y(x_0) = y_0$ расчетные формулы имеют следующий вид:

$$y_{m+1} = y_m + \tfrac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4),$$ где

$$k_1 = f(x_m, y_m);$$

$$k_2 = f\left(x_{m+h/2}, y_{m+\frac{hk_1}{2}}\right);$$

$$k_3 = f\left(x_{m+h/2}, y_{m+\frac{hk_2}{2}}\right);$$

$k_4 = f\left(x_{m+h}, y_{m+hk_3}\right);\ x_{m+1} = x_{m+h}$

Для системы дифференциальных уравнений

$y_i' = f_i(x, y_1, y_2, \dots y_k), y_i(x_0) = y_1^0, i = 1,2,..k;$

Для системы дифференциальных уравнений

расчетные формулы запишутся следующим образам:

$y_i^{n+1} = y_i^m + h/6(k_{1i} + 2k_{2i} + 2k_{3i} + k_{4i}), i = 1,2,..k;$

$k_{1i} = f_i(x_m, y_1^m, y_2^m, \dots y_k^m);$

$k_{2i} = f_i\left(x_m + \frac{h}{2}, y_1^m + \frac{h}{2} * \ k_{i1}, y_2^m + \frac{h}{2} * k_{12}, \dots y_k^m + \frac{h}{2} * k_{1k}\right);$

$k_{3i} = f_i\left(x_m + \frac{h}{2}, y_1^m + \frac{h}{2} * \ k_{11}, y_2^m + \frac{h}{2} * k_{22}, \dots y_k^m + \frac{h}{2} * k_{2k}\right);$

$k_{4i} = f_i\left(x_m + h, y_1^m + \frac{h}{2} * \ k_{21}, y_2^m + h * k_{32}, \dots y_k^m + h * k_{3k}\right);$

Текст программы

```
from math import exp, sqrt, tan, atan, cos, sin, log
import matplotlib.pyplot as plt
import numpy as np

# Функции для тестирования метода Рунге
def f1(x, y): # тестовая функция 1
    return sin(x)-y

def f2(x, y): # тестовая функция 2
    return y*cos(x)+sin(2*x)

def f31(x, u, v): # тестовая функция 3_1
    return cos(x+1.1*v)+u

def f32(x, u, v): # тестовая функция 3_2
    return -v**2+2.1*u+1.1

def f41(x, u, v): # тестовая функция 4_1
    return -v+sin(x)

def f42(x, u, v): # тестовая функция 4_2
    return u+cos(x)

def ans1(x): # точное решение для функции 1
    return -0.5*cos(x)+0.5*sin(x)+21/2*exp(-x)

def ans2(x): # точное решение для функции 2
    return -2*sin(x)+2*exp(sin(x))-2

def ans4_1(x): # точное решение u для системы 4
    return sin(x)+cos(x)

def ans4_2(x): # точное решение v для системы 4
    return 2*sin(x)-cos(x)

# Функции для тестирования решения краевой задачи
# Функции для теста 1
def p1(x):
    return 2*x**2

def q1(x):
    return 1

def f1(x):
    return x

# Функции для теста 2
def p2(x):
    return 0

def q2(x):
    return 1

def f2(x):
    return x+2*exp(x)

def ans22(x):
    return -sin(x)+cos(x)+exp(x)+x
```

```
# Функции для теста 3
def p3(x):
    return -3

def q3(x):
    return 2

def f3(x):
    return sin(x)

def ans23(x):
    return sin(x)/10+3*cos(x)/10+exp(x)+exp(2*x)

# Функции для теста 3
def p4(x):
    return 1/x

def q4(x):
    return 0

def f4(x):
    return 0

def ans24(x):
    return 6*log(x)+5

# Основные функции, осуществляющие вычисления
def arange(start, stop, n):
    step = (stop - start) / n
    return [round(x*step, 10) for x in range(int(start/step),
int(stop/step))]


def make_graphic(xmas, ymas, marker='', cl = None, lw = 1): #строит график
    return plt.plot(xmas, ymas, marker, color=cl, linewidth = lw)

def runge2(f, a, b, y0, n, xmas, ymas): # метод Рунге 2-го порядка для ОДУ
    xmas.clear()
    ymas.clear()
    d = (b-a)/n
    xmas.append(a)
    ymas.append(y0)
    for i in range(n):
        res = f(a, y0)
        y0 += d/2*(res+f(a+d, y0+res*d))
        a += d
        xmas.append(a)
        ymas.append(y0)

def runge4(f, a, b, y0, n, xmas, ymas): # метод Рунге 4-го порядка для ОДУ
    xmas.clear()
    ymas.clear()
    d = (b-a)/n
    xmas.append(a)
    ymas.append(y0)
    for i in range(n):
        res1 = f(a, y0)
        res2 = f(a+d/2, y0+d/2*res1)
        res3 = f(a+d/2, y0+d/2*res2)
        res4 = f(a+d, y0+d*res3)
```

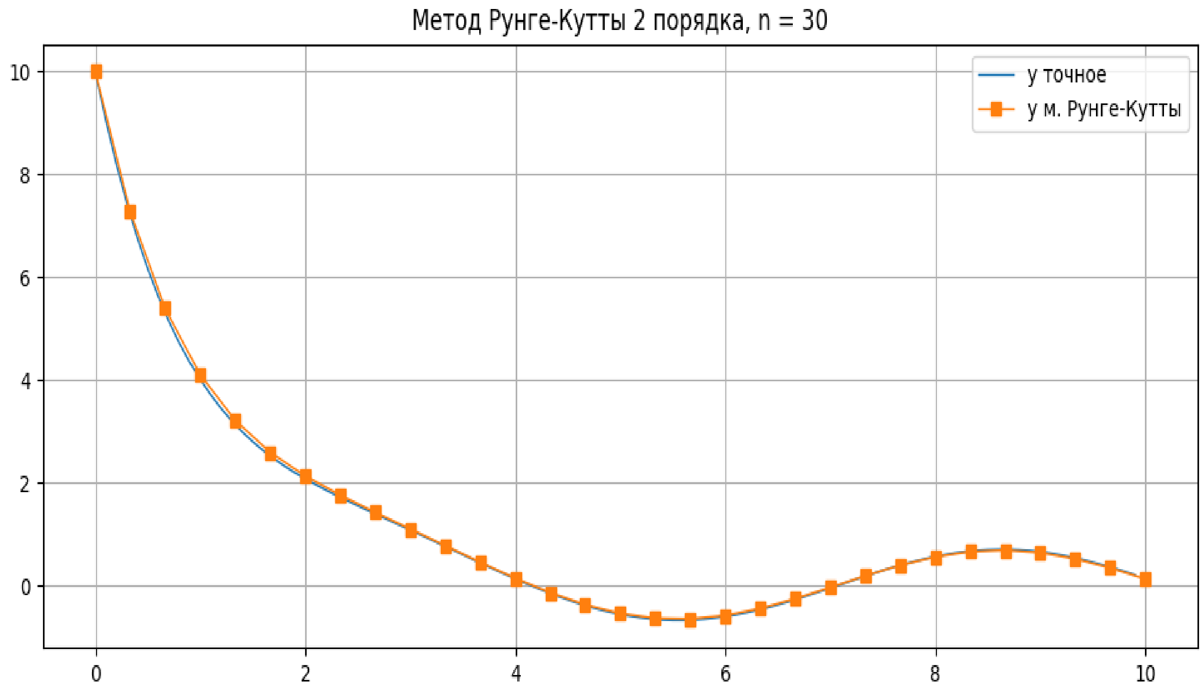
```
        y0 += d/6*(res1+2*res2+2*res3+res4)
        a += d
        xmas.append(a)
        ymas.append(y0)

def runge2_sys(f1, f2, a, b, y01, y02, n, xmas, ymas1, ymas2): # метод
Рунге 2-го порядка для системы ОДУ
    xmas.clear()
    ymas1.clear()
    ymas2.clear()
    d = (b-a)/n
    xmas.append(a)
    ymas1.append(y01)
    ymas2.append(y02)
    n -= 1
    for i in range(n):
        res1 = f1(a, y01, y02)
        res2 = f2(a, y01, y02)
        y01 += d/2*(res1+f1(a+d, y01+res1*d, y02+res2*d))
        y02 += d/2*(res2+f2(a+d, y01+res1*d, y02+res2*d))
        a += d
        xmas.append(a)
        ymas1.append(y01)
        ymas2.append(y02)

def runge4_sys(f1, f2, a, b, y01, y02, n, xmas, ymas1, ymas2): # метод
Рунге 4-го порядка для системы ОДУ
    xmas.clear()
    ymas1.clear()
    ymas2.clear()
    d = (b-a)/n
    xmas.append(a)
    ymas1.append(y01)
    ymas2.append(y02)
    n -= 1
    for i in range(n):
        res11 = f1(a, y01, y02)
        res21 = f2(a, y01, y02)
        res12 = f1(a+d/2, y01+d/2*res11, y02+d/2*res21)
        res22 = f2(a+d/2, y01+d/2*res11, y02+d/2*res21)
        res13 = f1(a+d/2, y01+d/2*res12, y02+d/2*res22)
        res23 = f2(a+d/2, y01+d/2*res12, y02+d/2*res22)
        res14 = f1(a+d, y01+d*res13, y02+d*res23)
        res24 = f2(a+d, y01+d*res13, y02+d*res23)
        y01 += d/6*(res11+2*res12+2*res13+res14)
        y02 += d/6*(res21+2*res22+2*res23+res24)
        a += d
        xmas.append(a)
        ymas1.append(y01)
        ymas2.append(y02)

def run_through_method(a, b, p, q, f, n, sigma, gamma, delta): # Метод
прогонки решения краевой задачи
    d = (b-a)/n
    n+=1
    amas = []
    bmas = []
    cmas = []
    fmas = []
    alpha = np.zeros(n-1)
    beta = np.zeros(n-1)
```

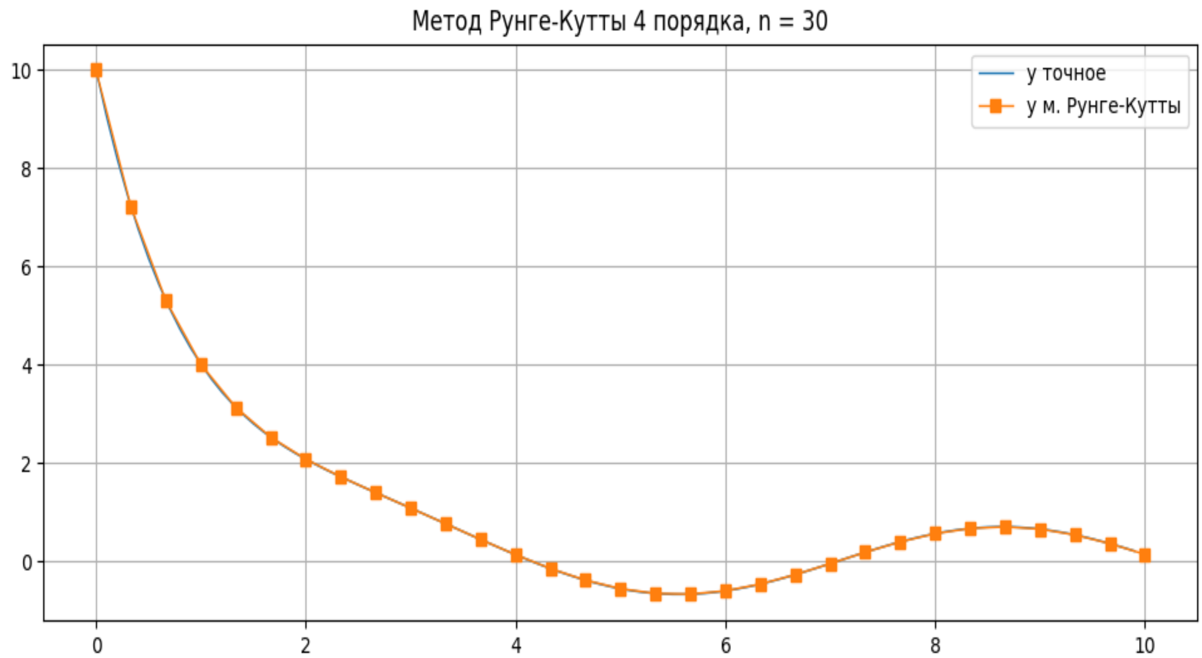
```
    result = np.zeros(n)
    def coef_a(m):
        return 1 - p(m)*d/2
    def coef_b(m):
        return 1 + p(m)*d/2
    def coef_c(m):
        return -2 + q(m)*(d**2)
    def coef_f(m):
        return f(m)*d**2


    for i in range(0,n):
        amas.append(coef_a(a+i*d))
        bmas.append(coef_b(a+i*d))
        cmas.append(coef_c(a+i*d))
        fmas.append(coef_f(a+i*d))
    amas[n-1] = -gamma[1]/d
    bmas[0] = gamma[0]/d
    cmas[0] = sigma[0]-gamma[0]/d
    cmas[n-1] = sigma[1]+gamma[1]/d
    fmas[0] = delta[0]
    fmas[n-1] = delta[1]
    alpha[0] = -bmas[0]/cmas[0]
    beta[0] = fmas[0]/cmas[0]

    for i in range(1, n-1):
        alpha[i] = -bmas[i]/(amas[i]*alpha[i-1]+cmas[i])
        beta[i] = (fmas[i] - amas[i] * beta[i-1])/(amas[i]*alpha[i-1]+cmas[i])

    result[n-1] = (fmas[n-1] - amas[n-1] * beta[n-2])/(amas[n-1]*alpha[n-2]+cmas[n-1])
    for i in range(n-2, -1, -1):
        result[i] = alpha[i]*result[i+1]+beta[i]
    return result
```

Тестирование

1. Таблица 1-2

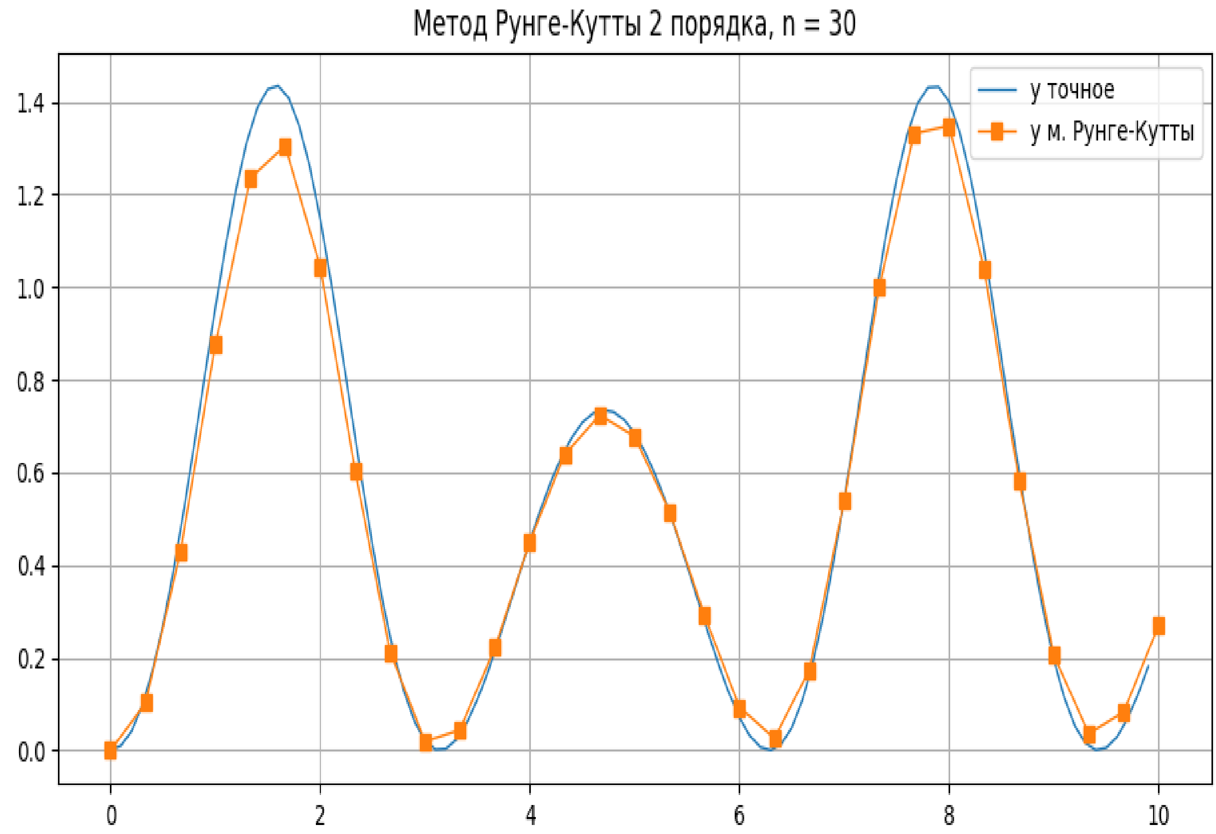
$f(x,y)$	(x_0,y_0)	Точное решение $y = y(x)$
$\sin(x) - y$	$(0,10)$	$-0.5\cos(x) + 0.5\sin(x) + \frac{21}{2}e^{-x}$

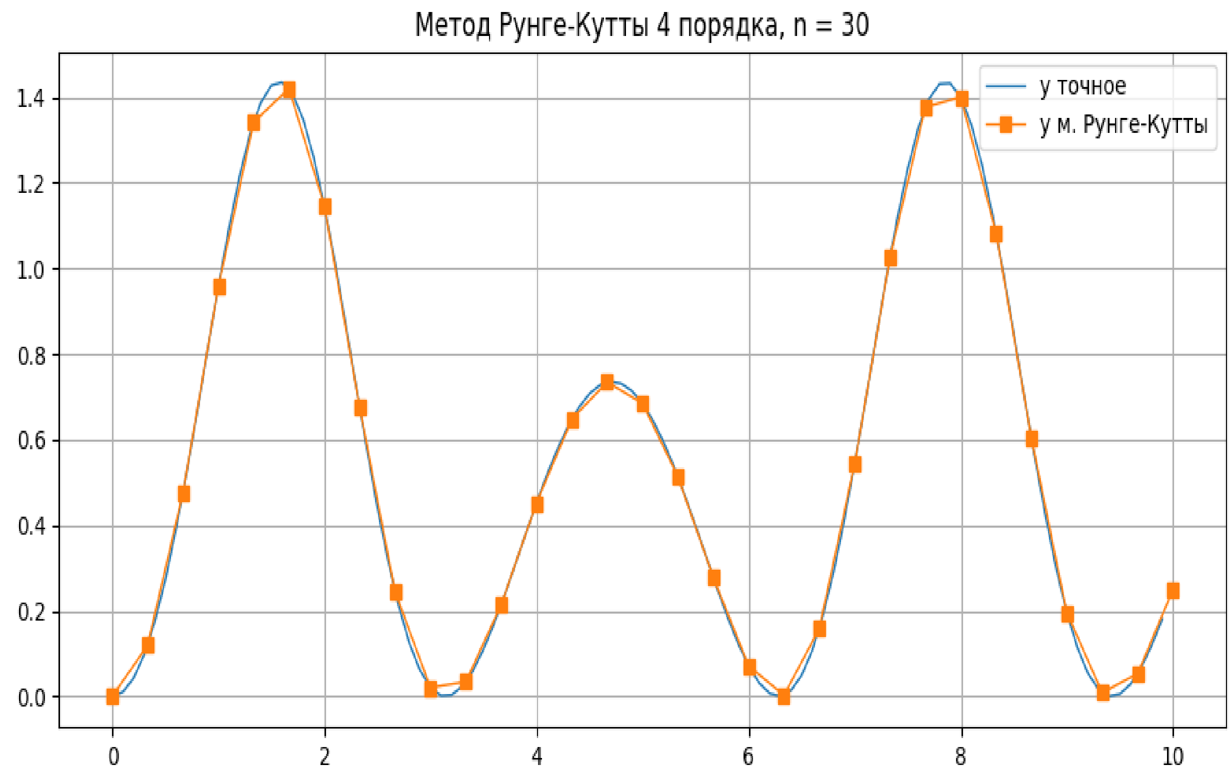




2. Собственный тест (не из таблицы)

$f(x,y)$	(x_0, y_0)	Точное решение $y = y(x)$
ycos(x) + sin(2x)	(0, 0)	$-2\sin(x) + 2e^{\sin(x)}-2$

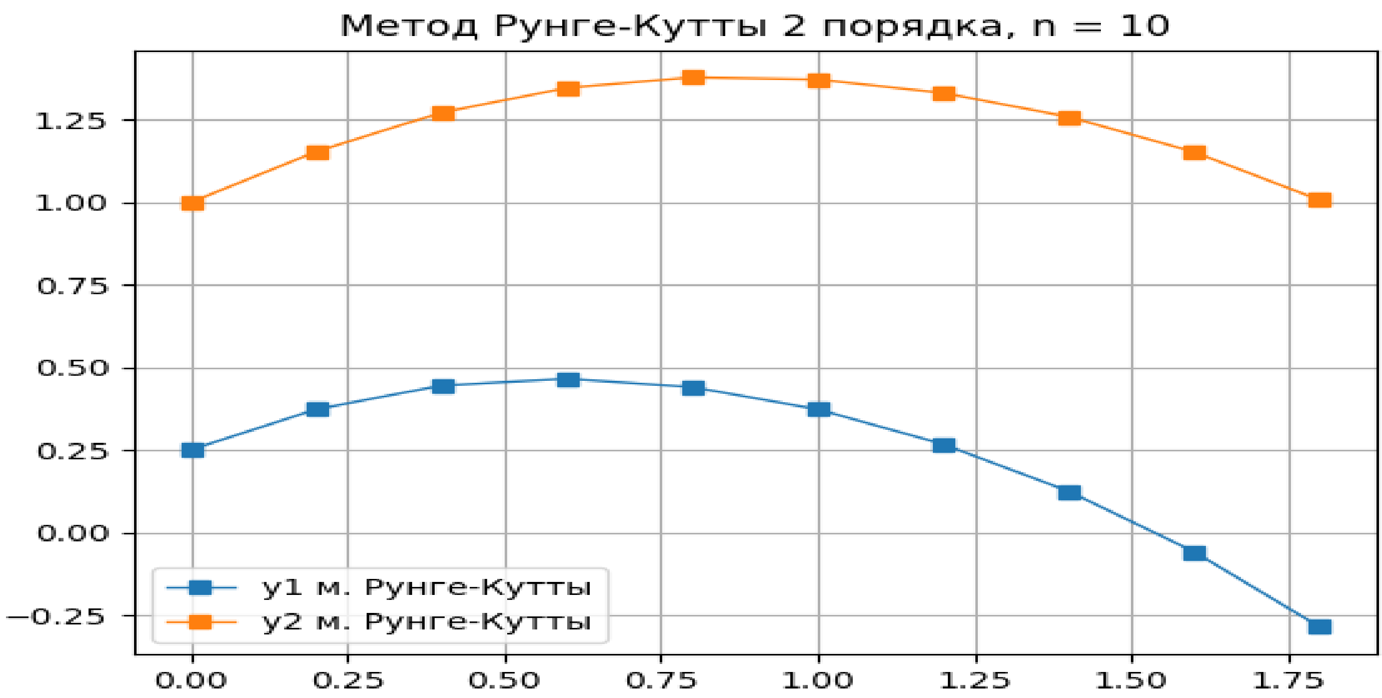


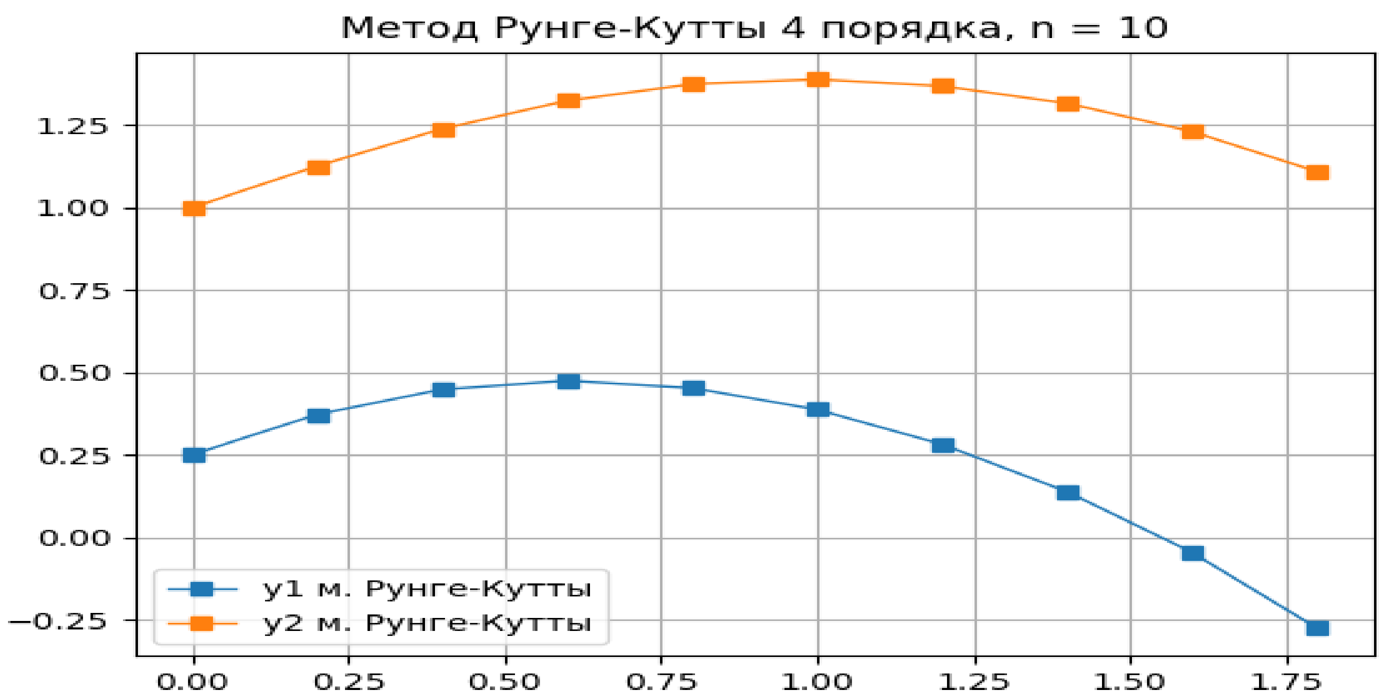


3. Таблица 2-8

$f_1(x,u,v)$	$f_2(x,u,v)$	x_0	$y_1^{(0)}$	$y_2^{(0)}$
$\cos(x+1.1\cdot v)+u$	$-v^2+2.1\cdot u+1.1$	0	0.25	1

Точное решение не удалось найти в элементарных функциях (Онлайн сервис https://www.wolframalpha.com/ не может найти решение данной задачи)



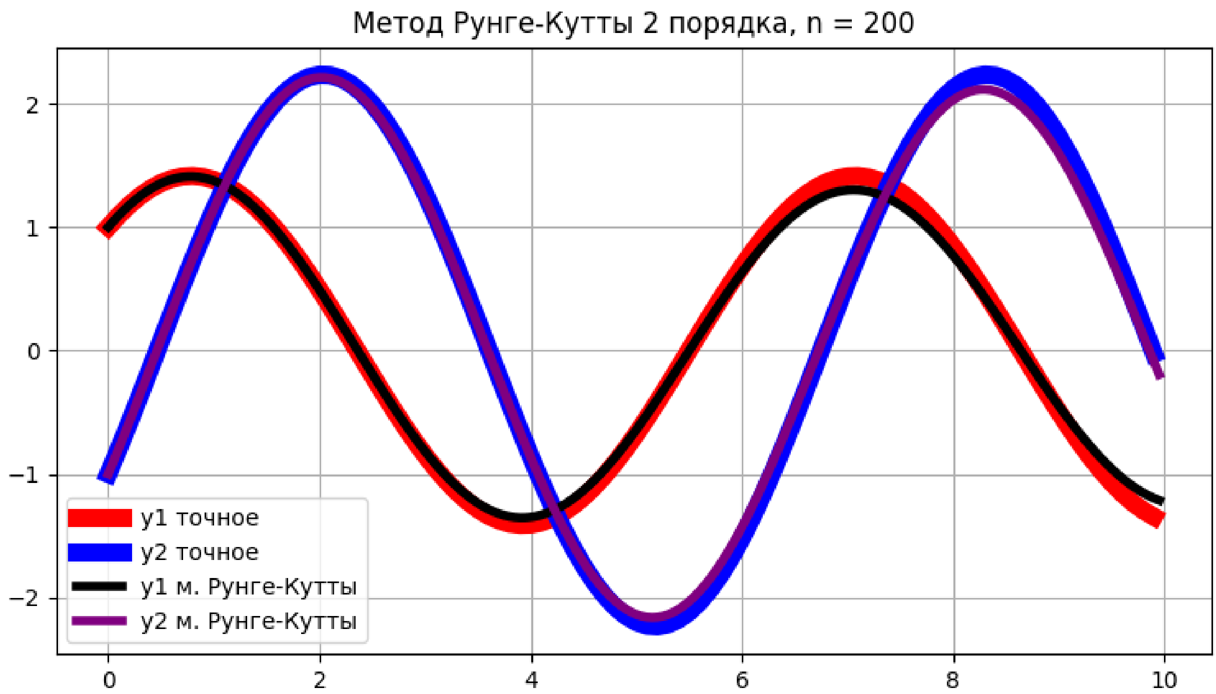


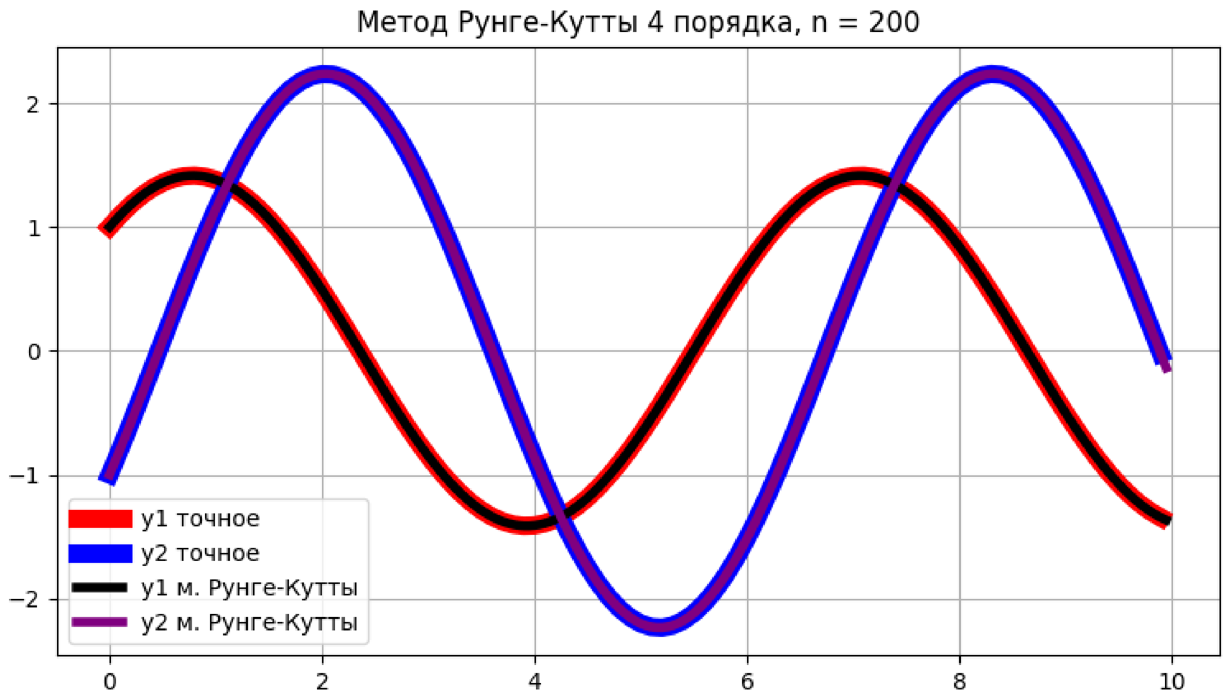
4. Собственный тест (не из таблицы)

$f_1(x,u,v)$	$f_2(x,u,v)$	x_0	$y_1^{(0)}$	$y_2^{(0)}$
-v+sin(x)	u+cos(x)	0	1	-1

Точное решение: u = sin(x) + cos(x)

v = 2sin(x)-cos(x)





РЕШЕНИЕ КРАЕВОЙ ЗАДАЧИ ДЛЯ ОБЫКНОВЕННОГО ДИФФЕРЕНЦИАЛЬНОГО УРАВНЕНИЯ ВТОРОГО ПОРЯДКА, РАЗРЕШЕННОГО ОТНОСИТЕЛЬНО СТАРШЕЙ ПРОИЗВОДНОЙ

Постановка задачи:

Рассматривается линейное дифференциальное уравнение второго порядка вида

$$y'' + p(x) \cdot y' + q(x) \cdot y = -f(x),\ 1 < x < 0, \tag{1}$$

с дополнительными условиями в граничных точках

$$\begin{cases} \sigma_1 y(0) + \gamma_1 y'(0) = \delta_1, \\ \sigma_2 y(1) + \gamma_2 y'(1) = \delta_2. \end{cases} \tag{2}$$

Цели и задачи практической работы

1) Решить краевую задачу (1)-(2) методом конечных разностей, аппроксимировав ее разностной схемой второго порядка точности (на равномерной 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.

Алгоритмы решения

Для решения задачи (1)-(2) применим разностный метод, получаемый путем аппроксимации первой и второй производной разностными соотношениями:

$$\frac{y_{i+1} - y_{i-1}}{2h}$$

и

$$\frac{y_{i-1} - 2y_i + y_{i+1}}{h^2}$$

В итоге получаем систему соотношений:

$$\frac{y_{i-1} - 2y_i + y_{i+1}}{h^2} + p(x_i)\frac{y_{i+1} - y_{i-1}}{2h} + q(x_i)y_i = f(x_i)$$

Которую можно преобразовать как:

$$y_{i-1}\left(\frac{1}{h^2} - \frac{p(x_i)}{2h}\right) + y_i\left(-\frac{2}{h^2} + q(x_i)\right) + y_{i+1}\left(\frac{1}{h^2} + \frac{p(x_i)}{2h}\right) = f(x_i)\, i = 1 \ldots n$$

Если записать данную систему, получим трехдиагональную матрицу. Аппроксимируя первую производную в краевых условиях с помощью приведенных выше разностных отношений можем получить следующее:

$$\begin{cases} y(x_0)(\sigma_1 - \frac{\gamma_1}{h}) + y(x_1)\frac{\gamma_1}{h} = \delta_1 \\ y(x_n)(\sigma_2 + \frac{\gamma_2}{h}) - y(x_{n-1})\frac{\gamma_2}{h} = \delta_2 \end{cases}$$

Полученную систему будем решать методом прогонки. В этом методе предполагается, что искомые неизвестные связаны рекуррентным соотношением. Сначала нужно найти пригоночные коэффициенты:

$$\begin{cases} \alpha_{i+1} = \frac{-B_i}{A_i\alpha_i + C_i} \\ \beta_{i+1} = \frac{F_i - A_i\beta_i}{A_i\alpha_i + C_i} \end{cases}$$

где

$$A_i y_{i-1} + C_i y + B_i y_{i+1} = F_i$$

После вычисления прогоночных коэффициентов необходимо решить следующую систему:

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

Текст программы:

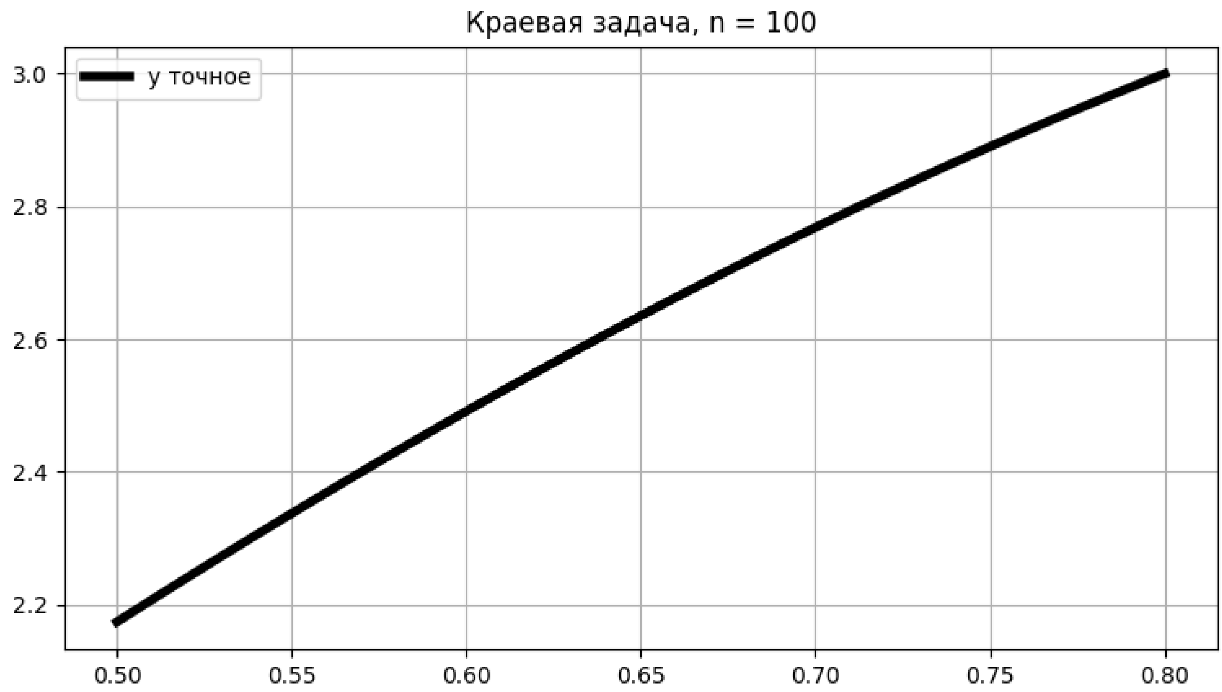
Текст программы был представлен выше на страницах 6 - 9

Тестирование

1.

Краевая задача 14. $y'' + 2x^2 y' + y = x$; $2y(0.5) - y'(0.5) = 1$; $y(0.8) = 3$.

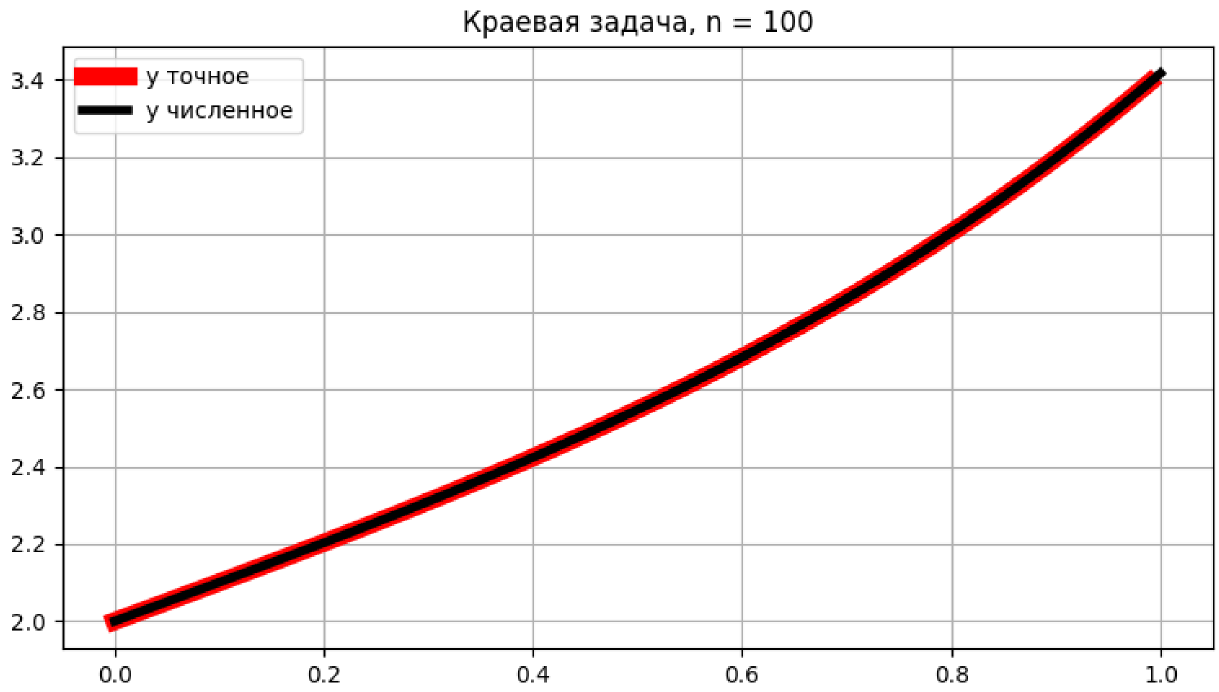
Точное решение не удалось найти в элементарных функциях (Онлайн сервис https://www.wolframalpha.com/ не может найти решение данной задачи)



2.

Краевая задача (Не из таблицы). $y'' + y = x + 2e^x$; $y'(0) = 1$; $y(1) = 3.4171131$

Точное решение: $-\sin(x)+\cos(x)+e^x+x$



3.

Краевая задача (Не из таблицы). $y'' - 3y' + 2y = \sin(x)$; $y(0) = 2.3$; $y'(\pi/2) = 27.65117$

Точное решение: $\sin(x)/10 + 3\cos(x)/10 + e^{x} + e^{2x}$

4.

Краевая задача (Не из таблицы). $y'' + y'/x = 0$; $y'(1) = 6$; $y(e) = 11$

Точное решение: $6\ln(x)+5$