МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ

Ордена Трудового Красного Знамени федеральное государственное бюджетное образовательное учреждение высшего образования

**«Московский технический университет связи и информатики»**

Кафедра «Информатика»

**Отчет по ЛР №5**

**по дисциплине**

**«Численные методы»**

Выполнил: студент гр. БЭИ2202

Тогузов А. А.

Вариант 24.

Проверил: доц. каф. «Информатика»

Мацкевич А. Г.

Москва, 2023 г.

**1) Задание для численного решения обыкновенных ДУ**

* ДУ:
* Интервал [0; 1.2]
* Начальные условия
* Шаг изменения аргумента

**2) Точное аналитическое решение заданного ДУ**

Найдем точное аналитическое решение заданного дифференциального уравнения (решение y=y(x)) методом разделения переменных. Для этого запишем уравнение в виде  и проинтегрируем с учетом начальных условий. Получим . Из начальных условий следует, что C=1. Аналитическое решение дифференциального уравнения .

**3) Значения точного решения ОДУ**

Вычислим в сценарии значения полученного решения на отрезке [0; 1.2] с шагом изменения аргумента .

x y

0 0.0 1.0000

1 0.4 1.1664

2 0.8 1.7424

3 1.2 2.9584

**4) Численное решение заданного ОДУ методом Эйлера**

Вычислим в сценарии значения численного решения ОДУ методом Эйлера () в точках отрезка [0; 1.2] с шагом h=0.4. Для этого ОДУ записывают в виде y’=f(x,y) . Общая формула для определения очередного значения функции по методу Эйлера имеет вид yi+1=yi+h⋅f(xi,yi), где , :

Приведем решения на Python

def f(x, y):

    return 2 \* x \* np.sqrt(y)

a = 0

b = 1.2

h = 0.4

x = np.arange(a, b + h, h)

y = [1]

for i in range(3):

    y\_cur = round(y[i] + h \* f(x[i], y[i]), 5)

    print(f"i={i}, y={y[i]:.5f}, f(x[i],y[i])={f(x[i], y[i]):.5f}, h\*f={h\*f(x[i], y[i]):.5f}, y\_cur={y\_cur:.5f}")

    y.append(y\_cur)

euler\_df = pd.DataFrame({"x": x, "y(x)" : y})

print(euler\_df)

x y(x)

0 0.0 1.0000

1 0.4 1.0000

2 0.8 1.3200

3 1.2 2.0553

**5) Значения погрешностей**

Вычислим в сценарии значения погрешностей 

x E

0 0.0 0

1 0.4 0.1664

2 0.8 0.4223

3 1.2 0.9031

**6) Результаты решения ОДУ методом Рунге-Кутта 4-го порядка.**

Вычислим в программе значения численного решения ОДУ с точностью 10-4, и получим решение в точках отрезка [0;1.2] с шагом h=0.4 () методом Рунге-Кутта 4-го порядка, используя формулы:



import numpy as np

import pandas as pd

def f(x, y):

    return 2 \* x \* np.sqrt(y)

def runge\_kutta(h, x0, y0, xn, precision=1e-4):

    data = {'x': [x0], 'y': [y0]}

    while x0 < xn:

        k1 = h \* f(x0, y0)

        k2 = h \* f(x0 + h/2, y0 + k1/2)

        k3 = h \* f(x0 + h/2, y0 + k2/2)

        k4 = h \* f(x0 + h, y0 + k3)

        y\_pred = y0 + (k1 + 2\*k2 + 2\*k3 + k4)/6

        y0 = y\_pred

        x0 = x0 + h

        data['x'].append(x0)

        data['y'].append(y0)

        if abs(y\_pred - data['y'][-2]) < precision:

            break

    df = pd.DataFrame(data)

    return df

x0, y0, xn = 0, 1, 1.2

h = 0.4

solution = runge\_kutta(h, x0, y0, xn, precision=1e-4)

print(solution)

В нашем случае получены следующие значения.

x y

0 0.0 1.000000

1 0.4 1.166356

2 0.8 1.742213

3 1.2 2.957891

**7) Таблица результатов**

Все решения, полученные выше, сведем в табл. результатов:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **xi** | **y(x)** | **E\_y** | **E\_error** | **R\_y** | **R\_error** |
| 0 | 1.0000 | 1.0000 | 0.0000 | 1.0000 | 0.0000 |
| 0.4 | 1.1664 | 1.0000 | 0.1664 | 1.166397 | 0.000003 |
| 0.8 | 1.7424 | 1.3200 | 0.4224 | 1.742388 | 0.000012 |
| 1.2 | 2.9584 | 2.0553 | 0.9031 | 2.958365 | 0.000035 |

Точное значение

x y

0 0.0 1.000000

1 0.5 0.713061

2 1.0 0.735759

3 1.5 0.946260

4 2.0 1.270671

Метод эйлера c шагом 0.5

x y(x)

0 0.0 1.0000

1 0.5 0.6000

2 1.0 0.5600

3 1.5 0.7360

4 2.0 1.0416

Метод эйлера c шагом 1

x y(x)

0 0.0 1.00

1 1.0 0.60

2 2.0 0.76

Оценить погрешность

x E

1 1.0 0.4

2 2.0 0.152