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 заданию №5**

**по дисциплине**

**«Численные методы»**

Выполнил: студент гр. БЭИ2202

Васильев В. А.

Вариант 8.

Проверил: доц. каф. «Информатика»

Мацкевич А. Г.

Москва, 2023 г.

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

* ДУ:
* Интервал [0; 1.5]
* 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=4. Аналитическое решение дифференциального уравнения

.

**3) Значения точного решения ОДУ**

Вычислим в сценарии значения полученного решения на отрезке [0; 1.5] с шагом изменения аргумента .

Изображение выглядит как текст, Шрифт, снимок экрана, число

Автоматически созданное описание

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

Вычислим в сценарии значения численного решения ОДУ методом Эйлера () в точках отрезка [0;1.5] с шагом h=0.5. 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

*# Given data*

a = 0

b = 1.5

h = 0.5

**def** f(x, y):

**return** (x - 1)\*\*2 \* y\*\*2

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

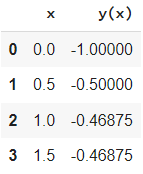
y = [-1]

**for** i **in** range(3):

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

y.append(y\_cur)

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



**5) Значения погрешностей**

Вычислим в сценарии значения погрешностей 

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

**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;0.4] с шагом h=0.1 () методом Рунге-Кутта 4-го порядка, используя формулы:



a = 0

b = 1.5

h = 0.5

i = 0

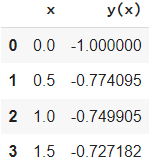
y = {i: exact\_f(i) for i in np.arange(a, b + 0.5, 0.5)}

while not(all([abs(runge\_kutta(f,0,-1,h)[i] - exact\_f(i)) < 1e-4 for i in np.arange(a, b + 0.5, 0.5)])):

  i += 1

  h /= 2

В нашем случае получены следующие значения.



**7) Зависимость количества итераций.**

Зависимость количества итераций от погрешности представлена на рисунке 1.

def runge\_kutta(f, x0, y0, h):

    y = {0: -1}

    x = np.arange(a, b + h, h)

    for i in range(len(x)-1):

        k1 = f(x[i], y[x[i]])

        k2 = f(x[i] + h/2, y[x[i]] + (k1 \* h)/2)

        k3 = f(x[i] + h/2, y[x[i]] + (k2 \* h)/2)

        k4 = f(x[i] + h, y[x[i]] + h \* k3)

        y\_cur = y[x[i]] + (h/6) \* (k1 + 2\*k2 + 2\*k3 + k4)

        y[x[i]+h] = y\_cur

    return y

def runge\_double\_prosch(f, x0, y0, h):

  m1 = runge\_kutta(f,x0,y0,h)

  m2 = runge\_kutta(f,x0,y0,h/2)

  error = []

  for i in np.arange(a, b + h, h):

    error.append(abs(m1[i] - m2[i]))

  return error

exps = range(1, 8)

iters = []

**for** exp **in** exps:

E = 10\*\*(-exp)

h = 1

i = 0

**while** runge\_double\_prosch(f, 0, -1, h)[0] > E:

i += 1

h /= 2

*# runge\_kutta(f, 0, -1, h)*

*# print(y)*

iters.append(i)

iters

Изображение выглядит как снимок экрана, текст, линия, График

Автоматически созданное описание

Рисунок 1 - Зависимость количества итераций от погрешности в методе Рунге-Кутта

**8) Таблица результатов**

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

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание