**ОТЧЁТ**

**ПО ЛАБОРАТОРНОЙ РАБОТЕ № 5**

**РЕШЕНИЕ ОДУ**

**(Вариант 5)**

*Выполнил студент 3 курса ПМ*

*Бут Дмитрий*

***Цель работы***: усвоить сущность и методы решения ***обыкновенных дифференциальных уравнений***. Овладеть технологией решения обыкновенного дифференциального уравнения.

Численное решение дифференциального уравнения предполагает получение числовой таблицы приближенных значений *yi* искомой функции *y* = *f*(*x)* с заданной точностью для некоторых значений аргумента *xi * [*a*, *b*].

Численное решение обыкновенных дифференциальных уравнений возможно методами:

метод Эйлера (первого порядка точности),

модифицированный метод Эйлера-Коши (второго порядка точности)

методы Рунге-Кутты

методы Адамса.

***Метод Рунге-Кутты*** четвёртого порядка точности имеет вид.

*k1* = *hf*(*xk*, *yk*),

*k2* = *hf*(*xk* + *h*/2, *yk* + *k1*/2),

*k3* = *hf*(*xk* + *h*/2, *yk* + *k2*/2),

*k4* = *hf*(*xk* + *h*, *yk* + *k3*),

*yk*=1/6(*k1* + *2k2* + *2k3* + *k4*), *yk* + 1=*yk* + *yk*, *xk* + 1=*xk* + *h.*

***Методы Адамса*** третьего и четвертого порядков точности имеют вид

*yi + 1 = yi + h (23y'i - 16y'i-1 + 5y'i-2)/12;*

*yi + 1 = yi + h (55y'i - 59y'i-1 + 37y'i-2 - 9y'i-3)/24.*

Погрешность решения, найденного этими методами, оценивается величиной O(*hm*)*,* где *m* - порядок метода.

Таким образом, метод Рунге-Кутта 4-го порядка и метод Адамса четвертого порядка имеют одинаковую оценку погрешности, но метод Адамса требует примерно вчетверо меньшего объема вычислений.

***Постановка задачи:***

**Для всех заданий точность 0,001**

Решить уравнение 1 методом Эйлера 2-го порядка точности (т.е. методом Эйлера-Коши) и методом Рунге-Кутта 4-го порядка точности.

Решить уравнение 2 методами Адамса 3-го порядка точности и 4-го порядка точности. ВНИМАНИЕ! Уравнение 2 - это дифференциальное уравнение 2-го порядка. Подробно расписать как решается уравнение.

Точность вычислений и для первого, и для второго уравнения контролировать методом двойного пересчета.

Сущность метода состоит в последовательных итерациях, каждая следующая из них соответствует удвоению числа точек разбиения. Сравниваются значения в совпадающих узлах. Вычисления прекращаются, когда максимальной модуль разности значений функции в совпадающих узлах для двух итераций становится меньше заранее заданной малой величины.

Результаты вывести в виде таблиц последних 16 точек для последней и 8 точек для предпоследней итераций, в которых первая колонка значения Хk , вторая колонка – значения найденных Yk для предпоследней итерации, третья - значения найденных Yk для последней итерации, четвертая – разность значений из 2-й и 3-й колонок.

Указать число точек разбиения для последней итерации.

***Результаты расчетов***

Первое уравнеие:

*y* = *cos*(1 + *x*) - 0,*5y2*

y(*a*) = 0, [*a*, *b*] = [0; 0,5]

**Метод Эйлера второго порядка (метод Эйлера-Коши).**

|  |  |  |  |
| --- | --- | --- | --- |
|  | (предпосл. итерация) | (послед. итерация) | Разность |
| 0.4414 | - | 0.1488 | - |
| 0.4453 | 0.1500 | 0.1492 | 0.0009 |
| 0.4492 | - | 0.1497 | - |
| 0.4531 | 0.1509 | 0.1501 | 0.0009 |
| 0.4570 | - | 0.1505 | - |
| 0.4609 | 0.1517 | 0.1509 | 0.0008 |
| 0.4648 | - | 0.1513 | - |
| 0.4688 | 0.1525 | 0.1517 | 0.0008 |
| 0.4727 | - | 0.1520 | - |
| 0.4766 | 0.1532 | 0.1524 | 0.0008 |
| 0.4805 | - | 0.1527 | - |
| 0.4844 | 0.1538 | 0.1530 | 0.0008 |
| 0.4883 | - | 0.1533 | - |
| 0.4922 | 0.1544 | 0.1536 | 0.0008 |
| 0.4961 | - | 0.1538 | - |
| 0.5000 | 0.1549 | 0.1541 | 0.0008 |

Количество точек на последней итерации 129

**Метод Рунге-Кутта (четвёртого порядка точности).**

|  |  |  |  |
| --- | --- | --- | --- |
|  | (предпосл. итерация) | (послед. итерация) | Разность |
| 0.0312 | - | 0.0165 | - |
| 0.0625 | - | 0.0321 | -0.00000000 |
| 0.0938 | - | 0.0468 | - |
| 0.1250 | 0.0607 | 0.0607 | - |
| 0.1562 | - | 0.0737 | - |
| 0.1875 | - | 0.0857 | -0.00000001 |
| 0.2188 | - | 0.0968 | - |
| 0.2500 | 0.1070 | 0.1070 | -0.00000002 |
| 0.2812 | - | 0.1162 | - |
| 0.3125 | 0.1244 | 0.1244 | -0.00000003 |
| 0.3438 | - | 0.1316 | - |
| 0.3750 | 0.1379 | 0.1379 | -0.00000003 |
| 0.4062 | - | 0.1432 | - |
| 0.4375 | 0.1475 | 0.1475 | -0.00000004 |
| 0.4688 | - | 0.1598 | - |
| 0.5000 | 0.1532 | 0.1532 | -0.00000004 |

Количество точек на последней итерации: 17

Второе уравнение:

*y* = 1 - s*in*(*x* + *y*)

y(*a*) = 0, [*a*, *b*] = [0; 0,5]

2 - y(*a*) = 1, следовательно,

Поскольку методы Адамса предназначены для дифференциальных уравнений первого порядка, нам необходимо преобразовать исходное уравнение.  
Для этого сделаем замену .  
Тогда получим систему:

Поскольку методы Адамса не являются самостартующими, то есть для того, что бы использовать метод Адамса необходимо иметь решения в первых четырех узлах для метода четвёртого порядка и в первых трёх для метода третьего порядка.

Вычислим значения в первых точек, используя метод Эйлера первого порядка:

Обозначим , тогда

Таким образом, на последней итерации возьмём значения первых 3 и 4 точек соответственно.

**Метод Адамса (третий порядок точности).**

Вычисления проводим по формулам:

*yi + 1 = yi + h (23y'i - 16y'i-1 + 5y'i-2)/12*

|  |  |  |  |
| --- | --- | --- | --- |
|  | (предпосл. итерация) | (послед. итерация) | Разность |
| 0.2665 | 0.3143 | 0.2957 | - |
| 0.2821 | - | 0.3143 | 0.0003 |
| 0.2977 | 0.3519 | 0.3331 | - |
| 0.3132 | - | 0.3519 | 0.0003 |
| 0.3288 | 0.3899 | 0.3709 | - |
| 0.3444 | - | 0.3899 | 0.0003 |
| 0.3599 | 0.4282 | 0.409 | - |
| 0.3755 | - | 0.4282 | 0.0002 |
| 0.3911 | 0.4668 | 0.4475 | - |
| 0.4066 | - | 0.4668 | 0.0002 |
| 0.4222 | 0.5056 | 0.4862 | - |
| 0.4377 | - | 0.5056 | 0.0002 |
| 0.4533 | 0.5446 | 0.5251 | - |
| 0.4689 | - | 0.5446 | 0.0002 |
| 0.4844 | 0.5837 | 0.5641 | - |
| 0.5 | - | 0.5837 | 0.0001 |

Количество точек на последней итерации: 35

**Метод Адамса (четвёртый порядок точности).**

Вычисления проводим по формуле:

*yi + 1 = yi + h (55y'i - 59y'i-1 + 37y'i-2 - 9y'i-3)/24*

|  |  |  |  |
| --- | --- | --- | --- |
|  | (предпосл. итерация) | (послед. итерация) | Разность |
| 0.2663 | 0.3196 | 0.3002 | 0.0 |
| 0.2819 | - | 0.3197 | - |
| 0.2975 | 0.359 | 0.3393 | 0.0 |
| 0.313 | - | 0.359 | - |
| 0.3286 | 0.3989 | 0.3789 | 0.0 |
| 0.3442 | - | 0.399 | - |
| 0.3598 | 0.4394 | 0.4191 | 0.0 |
| 0.3754 | - | 0.4394 | - |
| 0.3909 | 0.4802 | 0.4598 | 0.0 |
| 0.4065 | - | 0.4802 | - |
| 0.4221 | 0.5212 | 0.5007 | 0.0001 |
| 0.4377 | - | 0.5213 | - |
| 0.4533 | 0.5624 | 0.5419 | 0.0001 |
| 0.4688 | - | 0.5625 | - |
| 0.4844 | 0.6036 | 0.5831 | 0.0001 |
| 0.5 | - | 0.6037 | - |

Количество точек на последней итерации: 36

**ПРИЛОЖЕНИЕ**

***Программа реализации метода Эйлера-Коши***

import numpy as np

import math

# Заданное дифференциальное уравнение y' = cos(x + 2) - 2.3\*y^2

def f(x, y):

return math.cos(x + 2) - 2.3 \* y\*\*2

def euler(f, a, b, y0, epsilon):

n = 2

last\_y = [y0]

while True:

h = (b - a)/n

x, y = [a], [y0]

for i in range(n):

x.append(x[i] + h)

y\_t = y[i] + h \* f(x[i], y[i]) # Метод Эйлера (предварительный шаг)

y.append(y[i] + (h / 2) \* (f(x[i], y[i]) + f(x[i], y\_t))) # Итоговый шаг

diffs = [abs(y[i\*2]-last\_y[i]) for i in range(n//2)]

if max(diffs) < epsilon and n > 4:

for i in range(16):

if i % 2 == 0:

print(f'{x[-16+i]:<10.4f} {'-':<25} {y[-16+i]:<25.4f} {'-':<15}')

else:

print(f'{x[-16+i]:<10.4f} {last\_y[-8+i//2]:<25.4f} {y[-16+i]:<25.4f} {diffs[-i//2]:<15.4f}')

return len(x)

last\_y = y

n \*= 2

a, b = 0, 0.5

y0 = 0

epsilon = 0.001

print("Метод Эйлера-Коши:")

print(f"{'x\_k':<10}{'Y\_k (предпосл. итерация)':<25}{'Y\_k (послед. итерация)':<25}{'Разность':<15}")

print('Количество итераций:', euler(f, a, b, y0, epsilon))

***Программа реализации метода Рунге-Кутта ( четвёртого порядка)***

import numpy as np

import math

# Заданное дифференциальное уравнение y' = cos(x + 2) - 2.3\*y^2

def f(x, y):

return math.cos(x + 2) - 2.3 \* y\*\*2

def runge(f, a, b, y0, epsilon):

n = 2

last\_y = [y0]

while True:

h = (b - a)/n

x, y = [a], [y0]

for i in range(n):

x.append(x[i] + h)

k1 = f(x[i], y[i])

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

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

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

y\_d = h \* (k1 + 2 \* k2 + 2 \* k3 + k4) / 6

y.append(y[i] + y\_d)

diffs = [abs(last\_y[i] - y[2 \* i]) for i in range(n//2)]

if max(diffs) < epsilon and math.log(n, 2) > 3:

for i in range(16):

if i % 2 == 0:

print(f'{x[-16+i]:<10.4f} {'-':<25} {y[-16+i]:<25.4f} {'-':<15}')

else:

print(f'{x[-16+i]:<10.4f} {last\_y[-8+i//2]:<25.4f} {y[-16+i]:<25.4f} {last\_y[-8+i//2]-y[-16+i]:<15.8f}')

return len(x), h, f'{h\*\*4:.8f}'

last\_y = y

n \*= 2

a, b = 0, 0.5

y0 = 0

epsilon = 0.001

print('Метод Рунге-Кутта')

print(f"{'x\_k':<10}{'Y\_k (предпосл. итерация)':<25}{'Y\_k (послед. итерация)':<25}{'Разность':<15}")

print(runge(f, a, b, y0, epsilon))

***Программа получения n первых точек методом Эйлера первого порядка***

from math import log, sin

def euler(a, b, y0, z0, f, eps, count):

n = 2

last\_y = [y0]

last\_z = [z0]

x = []

while True:

h = (b - a) / n

x = [a]

y = [y0]

z = [z0]

for i in range(n):

y.append(y[i] + h \* z[i])

z.append(z[i] + h \* f(x[i], y[i - 1]))

x.append(x[i] + h)

max\_diff\_y\_and\_z = 0

for i in range(n // 2):

max\_diff\_y\_and\_z = max(max\_diff\_y\_and\_z, abs(last\_y[i] - y[2 \* i]), abs(last\_z[i] - z[2 \* i]))

if max\_diff\_y\_and\_z < eps and log(n, 2) > 2:

return y[:count], z[:count], x[:count]

last\_y = y

last\_z = z

n \*= 2

***Программа реализации метода Адамса третьего порядка***

def adams3(yn, zn, xn, b, f, eps):

n = 2

last\_y = yn[:]

last\_z = zn[:]

x = xn[:]

while True:

x = xn[:]

h = (b - xn[2]) / n

y = yn[:]

z = zn[:]

for i in range(2, n + 2):

x.append(x[i] + h)

y.append(y[i] + h \* (23 \* z[i] - 16 \* z[i - 1] + 5 \* z[i - 2]) / 12)

z.append(z[i] + h \* (23 \* f(x[i], y[i]) - 16 \* f(x[i - 1], y[i - 1]) + 5 \* f(x[i - 2], y[i - 2])) / 12)

max\_diff\_y\_and\_z = 0

for i in range(n // 2):

max\_diff\_y\_and\_z = max(max\_diff\_y\_and\_z, abs(last\_y[i + 2] - y[2 \* i + 2]), abs(last\_z[i + 2] - z[2 \* i + 2]))

if max\_diff\_y\_and\_z < eps and log(n, 2) > 2:

break

last\_y = y

last\_z = z

n \*= 2

print("16 последних значений последней итерации:")

for i in range(len(x) - 16, len(x)):

print(round(x[i], 4))

print("8 значений функции предпоследней итерации:")

for i in range(len(last\_y) - 8, len(last\_y)):

print(round(last\_y[i], 4))

print('-')

print("16 значений функции последней итерации:")

for i in range(len(y) - 16 , len(y)):

print(round(y[i], 4))

print("Разность значений между последней и предпоследней итерацией:")

for i in range(8):

print(round(abs(last\_y[-i - 1] - y[- 2 \* i - 1]), 5))

print('-')

print("Количество разбиений на последней итерации:", len(x))

def f(x, y):

return 1 + 0.4 \* y \* sin(x) - 3.5 \* y\*\*2

print('Адамс 3 порядок')

a, b = 0, 0.5

y0, z0 = 0, 1

eps = 0.001

yn, zn, xn = eiler(a, b, y0, z0, f, eps, 3)

adams3(yn, zn, xn, b, f, eps)

***Реализация метода Адамса четвёртого порядка***

def adams4(yn, zn, xn, b, f, eps):

n = 2

last\_y = yn[:]

last\_z = zn[:]

x = xn[:]

while True:

x = xn[:]

h = (b - xn[3]) / n

y = yn[:]

z = zn[:]

for i in range(3, n + 3):

x.append(x[i] + h)

y.append(y[i] + h \* (55 \* z[i] - 59 \* z[i - 1] + 37 \* z[i - 2] - 9 \* z[i - 3]) / 24)

z.append(z[i] + h \* (55 \* f(x[i], y[i]) - 59 \* f(x[i - 1], y[i - 1]) \

+ 37 \* f(x[i - 2], y[i - 2]) - 9 \* f(x[i - 3], y[i - 3])) / 24)

max\_diff\_y\_and\_z = 0

for i in range(n // 2):

max\_diff\_y\_and\_z = max(max\_diff\_y\_and\_z, abs(last\_y[i + 3] - y[2 \* i + 3]), abs(last\_z[i + 3] - z[2 \* i + 3]))

if max\_diff\_y\_and\_z < eps and log(n, 2) > 2:

break

last\_y = y

last\_z = z

n \*= 2

print("16 последних значений последней итерации:")

for i in range(len(x) - 16, len(x)):

print(round(x[i], 4))

print("8 значений функции предпоследней итерации:")

for i in range(len(last\_y) - 8, len(last\_y)):

print(round(last\_y[i], 4))

print('-')

print("16 значений функции последней итерации:")

for i in range(len(y) - 16 , len(y)):

print(round(y[i], 4))

print("разность значений:")

for i in range(8):

print(round(abs(last\_y[-i - 1] - y[- 2 \* i - 1]), 4))

print('-')

print("Количество разбиений :", len(x))

def f(x, y):

return 1 + 0.4 \* y \* sin(x) - 3.5 \* y\*\*2

print('Адамс 4 порядок')

a, b = 0, 0.5

y0, z0 = 0, 1

eps = 0.001

yn, zn, xn = eiler(a, b, y0, z0, f, eps, 4)

adams4(yn, zn, xn, b, f, eps)