Российский университет дружбы народов им. П. Лумумбы

Факультет физико-математических и естественных наук

Лабораторная работа №5

Дисциплина: Вычислительные методы

Студент: Шуплецов Александр Андреевич

Группа: НФИбд-01-22

Москва

2024 г.

Оглавление

Задание

Задача Коши

1. Реализовать в программе метод Рунге-Кутты 2-го порядка точности для численного решения задачи Коши, приведенной ниже:

$$\begin{cases} y' = f(x,y), & x \in [a,b], \\ y(a) = y_0, \end{cases}$$

где функция $f(x,y)$, концы отрезка a, b и начальное значение y_0 заданы в индивидуальном варианте (использовать тот же вариант, что и в лабораторной работе № 4).

2. В программе вывести таблицу данных следующего вида:

$$\begin{array}{cccc} x_0 & \tilde{y}(x_0) & y(x_0) & \delta(x_0) \\ x_1 & \tilde{y}(x_1) & y(x_1) & \delta(x_1) \\ \vdots & \vdots & \vdots & \vdots \\ x_N & \tilde{y}(x_N) & y(x_N) & \delta(x_N) \end{array}$$

где целое число $N = 32$ определяет равномерное разбиение отрезка $[a,b]$, $\tilde{y}(x_j)$ – значение численного решения задачи Коши, полученного методом Рунге-Кутты 2-го порядка точности, в узлах разбиения отрезка; $y(x_j)$ – значение аналитического решения задачи Коши в узлах разбиения отрезка; $\delta(x_j) = |\tilde{y}(x_j) - y(x_j)|$ – погрешность.

3. Определить минимальное значение N, при котором $\delta(x_j) < 10^{-2}$, $j = \overline{0,N}$.

4. Сравнить полученное в п.3 значение N с аналогичным значением, полученным для метода Эйлера. Проанализировать результат.

Теоретическая справка

Данная лабораторная работа представляет собой компьютерную реализацию метода Рунге-Кутта для численного решения задачи Коши.

3.2.3 Метод Рунге-Кутта (5)

Заменяя с помощью (53) левую часть уравнения (52) и отбрасывая члены порядка $O\left(h^2\right)$, получаем однопараметрическое семейство разностных схем Рунге-Кутта:

$$\frac{y_{i+1}-y_i}{h}=(1-\alpha)\,f(x_i,y_i)+\alpha\,f\left\{x_i+\frac{h}{2\alpha},y_i+\frac{h}{2\alpha}\,f(x_i,y_i)\right\} \qquad (57)$$

Уравнение (57), как и (29), можно записать в виде удобного для расчетов рекуррентного соотношения:

$$y_{i+1}=y_i+\left[(1-\alpha)\,f(x_i,y_i)+\alpha\,f\left\{x_i+\frac{h}{2\alpha},y_i+\frac{h}{2\alpha}\,f(x_i,y_i)\right\}\right]h. \qquad (58)$$

69

Полный программный код, с подробным описанием функции, реализующей метод Рунге-Кутта.

```
import math

def func(x,y):
    return 4*(x**3)*y
    # return x*y

def splitter(a,b,N):
    step = (b-a)/N
    arr_x = [0]*(N+1)
    for i in range (0,N+1):
        arr_x[i] = a + step*i
    return arr_x

def metod_Runge_Kutta(u,h, N):
    arr_x = splitter(a,b,N)
    arr_y = [0]*(N+1)
    arr_y[0] = u

    for i in range (0,N):
            k1 = func(arr_x[i],arr_y[i])
            k2 = func(arr_x[i] + h, arr_y[i] + h*k1)
            arr_y[i+1] = arr_y[i] + h*((k1 + k2)/2)
    return arr_y

def anal_resh(N):
    arr_x = splitter(a,b,N)
    arr_y = [0]*(N+1)
    for i in range (0,N+1):
        arr_y[i] = math.e**arr_x[i]**4
        # arr_y[i] = math.e**(arr_x[i]**2/2)
    return arr_y

global a, b
a = 0
b = 1
N = 32
h = (b-a)/N
max_delta = 0

for i in range (N+1):
```

```
    print("{:<10f}{:<10f}{:<10f}{:<10f}".format(splitter(a,b,N)[i],
metod_Runge_Kutta(1,h,N)[i], anal_resh(N)[i], abs(anal_resh(N)[i] -
metod_Runge_Kutta(1,h,N)[i])))

    if abs(anal_resh(N)[i] - metod_Runge_Kutta(1,h,N)[i]) > max_delta:
        max_delta = abs(anal_resh(N)[i] - metod_Runge_Kutta(1,h,N)[i])

print("--------------------------")

print("Макс ошибка: ", max_delta)

print("--------------------------")

for n in range(1, 1000):
    h = (b-a)/n
    max_delta = 0
    resh = anal_resh(n)
    metod_Runge_Kutta_cur = metod_Runge_Kutta(1,h,n)
    for i in range (n+1):
        if abs(resh[i] - metod_Runge_Kutta_cur[i]) > max_delta:
            max_delta = abs(resh[i] - metod_Runge_Kutta_cur[i])
    if max_delta < 0.01:
        print(n)
        break
```

1. Функция func(x, y):
   - Эта функция принимает два параметра x и y.
   - Она возвращает результат вычисления выражения 4 * (x**3) * y, то есть умножает четверку на куб числа x и на y.
2. Функция splitter(a, b, N):
   - Принимает три параметра: a (начало отрезка), b (конец отрезка) и N (число интервалов).
   - Вычисляет шаг step как (b-a)/N.
   - Возвращает список arr_x, содержащий N+1 элементов, которые представляют собой координаты точек, равномерно расположенных от a до b с шагом step.
3. Функция metod_Runge_Kutta(u, h, N):
   - Принимает три параметра: u (начальное значение y), h (шаг интегрирования) и N (число интервалов).
   - Использует метод Рунге-Кутты второго порядка для численного решения дифференциального уравнения.
   - Возвращает список arr_y, в котором хранятся вычисленные значения решения в точках arr_x.
4. Функция anal_resh(N):
   - Принимает параметр N, который задает число интервалов.
   - Возвращает список arr_y, содержащий аналитическое решение на каждом шаге из списка arr_x.
5. Глобальные переменные a, b:
   - Устанавливают границы интервала интегрирования: от a = 0 до b = 1.
6. Основной код:
   - Выполняется численное и аналитическое решение дифференциального уравнения на интервале [a, b] с количеством шагов N = 32.
   - Вычисляет и печатает значения и погрешности численного метода.
   - Перебирает значения n от 1 до 1000 для поиска минимального числа шагов, при котором максимальная погрешность становится меньше 0.01. При нахождении подходящего n выводит его значение и завершает цикл.

Численные расчеты

```
0.000000  1.000000  1.000000  0.000000
0.031250  1.000002  1.000001  0.000001
0.062500  1.000019  1.000015  0.000004
0.093750  1.000086  1.000077  0.000009
0.125000  1.000259  1.000244  0.000015
0.156250  1.000620  1.000596  0.000024
0.187500  1.001271  1.001237  0.000034
0.218750  1.002339  1.002292  0.000047
0.250000  1.003975  1.003914  0.000061
0.281250  1.006354  1.006277  0.000078
0.312500  1.009678  1.009582  0.000096
0.343750  1.014177  1.014061  0.000116
0.375000  1.020111  1.019972  0.000139
0.406250  1.027777  1.027612  0.000164
0.437500  1.037508  1.037316  0.000192
0.468750  1.049686  1.049464  0.000222
0.500000  1.064750  1.064494  0.000255
0.531250  1.083202  1.082910  0.000292
0.562500  1.105627  1.105296  0.000331
0.593750  1.132712  1.132337  0.000375
0.625000  1.165266  1.164845  0.000422
0.656250  1.204258  1.203786  0.000472
0.687500  1.250852  1.250326  0.000527
0.718750  1.306464  1.305880  0.000584
0.750000  1.372829  1.372188  0.000641
0.781250  1.452098  1.451401  0.000697
0.812500  1.546955  1.546209  0.000745
0.843750  1.660784  1.660007  0.000777
0.875000  1.797892  1.797113  0.000779
0.906250  1.963809  1.963082  0.000727
0.937500  2.165704  2.165121  0.000583
0.968750  2.412968  2.412680  0.000288
1.000000  2.718027  2.718282  0.000255
--------------------------
Макс ошибка:  0.000778944710640106
--------------------------
8
```

Вывод

Я реализовал метод Рунге-Кутта 2-го порядка точности для численного решения задачи Кошис помощью компьютерной программы на языке Python.