Университет ИТМО

Лабораторная работа №5 «Численное дифференцирование и задача Коши»

по дисциплине: Вычислительная математика

Вариант: метод Рунге-Кутты 4-го порядка

Выполнил: Неграш Андрей, P3230

Преподаватель: Перл Ольга Вячеславовна

Санкт-Петербург, 2022

Описание метода решения

Метод Рунге-Кутты 4-го порядка находит приближенное значение y для заданного x. С помощью данного метода можно решить только обыкновенные дифференциальные уравнения первого порядка. Является одношаговым методом.

По формуле, указанной в разделе ниже, высчитываются значения на каждом шаге. Меньший размер шага означает большую точность. Формула вычисляет значение y_{n+1}, основываясь на значении y_n и средневзвешенном значении четырёх приращений:

k_1 – это приращение, основанное на наклоне в начале интервала, используя y

k_2 – это приращение, основанное на наклоне в середине интервала, используя $y + hk_{\frac{1}{2}}$

k_3 – это приращение, основанное на наклоне в средней точке, используя $y + hk_{\frac{2}{2}}$

k_4 – это приращение, основанное на наклоне в конце интервала, используя $y + hk_3$

Расчётные формулы

$$y' = f(x, y),\ \ y(x_0) = y_0$$

$$y_{n+1} = y_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

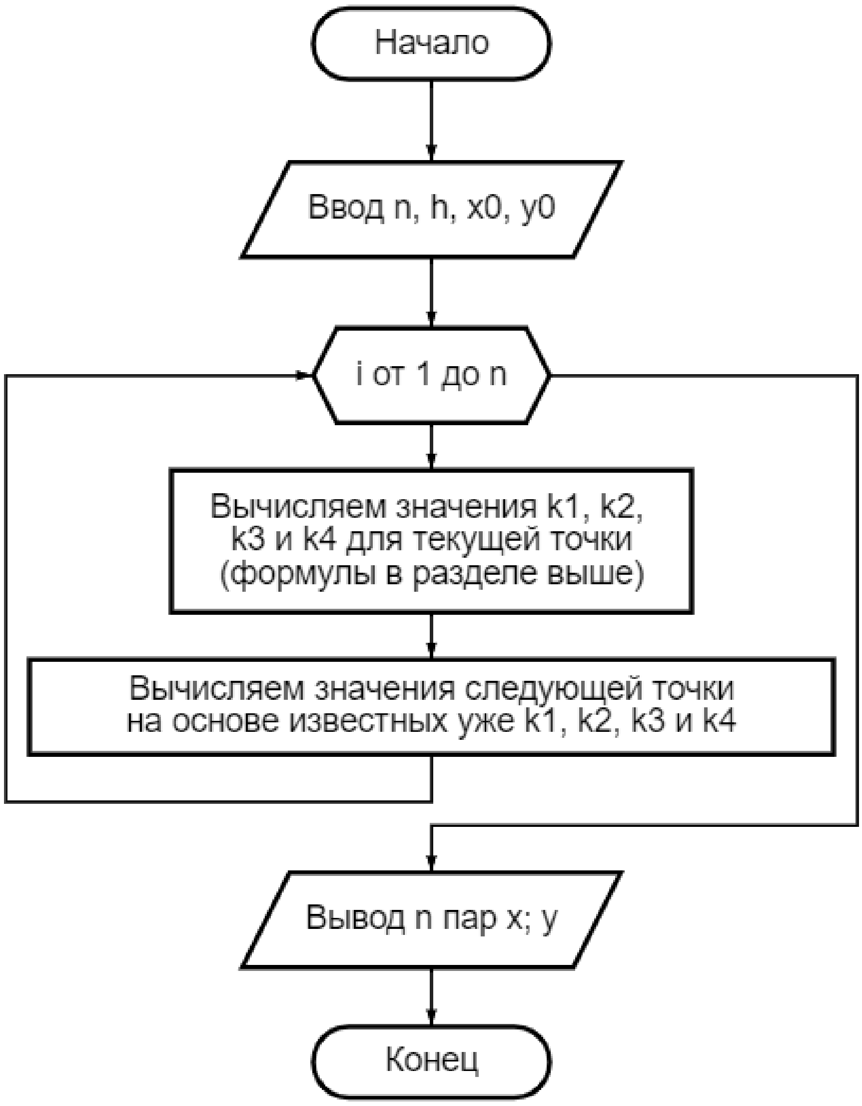
$$k_1 = f(x_n, y_n)$$

$$k_2 = f(x_n + \frac{h}{2}, y_n + \frac{h}{2}k_1)$$

$$k_3 = f(x_n + \frac{h}{2}, y_n + \frac{h}{2}k_2)$$

$$k_4 = f(x_n + h, y_n + hk_3)$$

Блок-схема численного метод



Листинг численного метода

```
def runge_kutta_solve(initial_point, function_id, h, n):
    x_current = initial_point[0]
    y_current = initial_point[1]
    points = [[x_current, y_current]]
    for i in range(1, n):
        k1 = functions.get_function(function_id, x_current, y_current)
        k2 = functions.get_function(function_id, x_current + h / 2, y_current
+ h * k1 / 2)
        k3 = functions.get_function(function_id, x_current + h / 2, y_current
+ h * k2 / 2)
        k4 = functions.get_function(function_id, x_current + h, y_current + h
* k3)

        dyi = (k1 + 2 * k2 + 2 * k3 + k4) * h / 6
        x_current += h
        y_current += dyi
        points.append([x_current, y_current])

    return points
```

Работа программы

Пример 1

Лабораторная №5: "Численное дифференцирование и задача Коши"
Вариант: метод Рунге-Кутты 4-го порядка
Автор: Неграш А.В., P3230
Выберите уравнение, которое необходимо решить:
1: y' = -2 * y
2: y' = y * (x² + 1)
3: y' = sin(x) + y

> *1*

Введите значение n (количество точек):

> *10*

Введите значение h (шаг):

> *0.1*

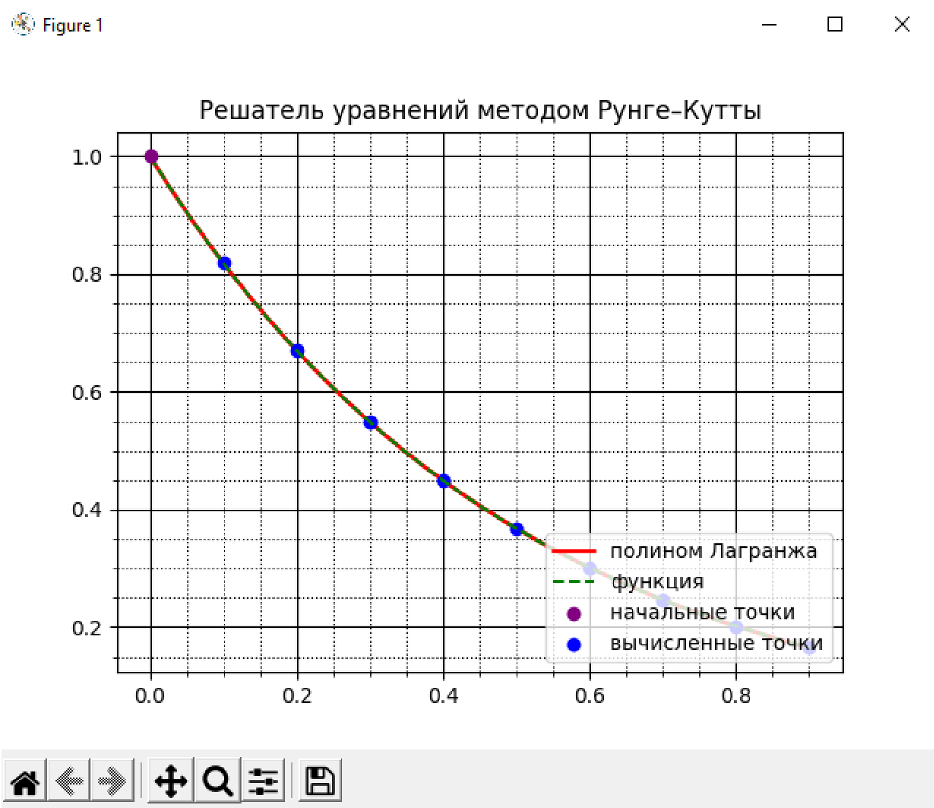
Введите начальное значение x:

> *0*

Введите начальное значение y:

> *1*

Вычисленные точки:
x = 0.00000; y = 1.000000
x = 0.10000; y = 0.818733
x = 0.20000; y = 0.670324
x = 0.30000; y = 0.548817
x = 0.40000; y = 0.449335
x = 0.50000; y = 0.367885
x = 0.60000; y = 0.301200
x = 0.70000; y = 0.246602
x = 0.80000; y = 0.201902
x = 0.90000; y = 0.165304



Пример 2

Лабораторная №5: "Численное дифференцирование и задача Коши"
Вариант: метод Рунге-Кутты 4-го порядка
Автор: Неграш А.В., P3230

Выберите уравнение, которое необходимо решить:
1: y' = -2 * y
2: y' = y * (x² + 1)
3: y' = sin(x) + y

> 2

Введите значение n (количество точек):

> 20

Введите значение h (шаг):

> 0.1

Введите начальное значение x:

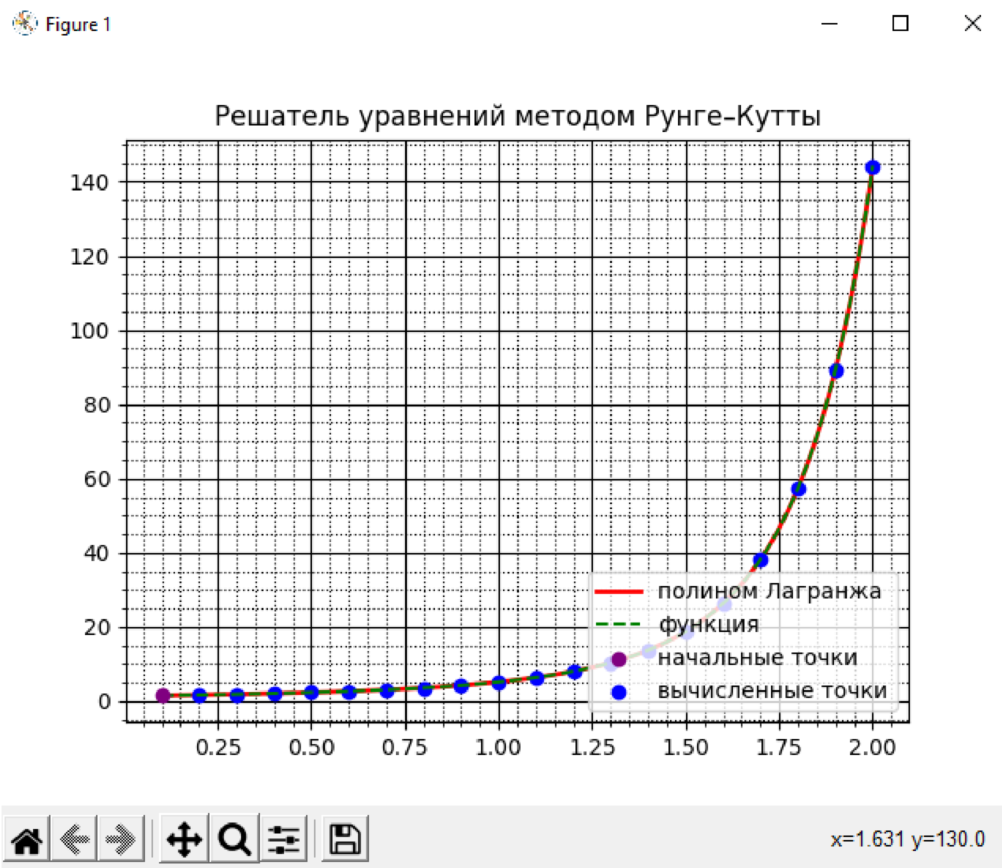
> 0.1

Введите начальное значение y:

> 1.5

Вычисленные точки:

x = 0.10000; y = 1.500000
x = 0.20000; y = 1.661629
x = 0.30000; y = 1.848051
x = 0.40000; y = 2.067758
x = 0.50000; y = 2.332167
x = 0.60000; y = 2.656823
x = 0.70000; y = 3.063211
x = 0.80000; y = 3.581553
x = 0.90000; y = 4.255144
x = 1.00000; y = 5.147237
x = 1.10000; y = 6.352132
x = 1.20000; y = 8.013431
x = 1.30000; y = 10.354742
x = 1.40000; y = 13.732507
x = 1.50000; y = 18.729131
x = 1.60000; y = 26.321448
x = 1.70000; y = 38.193786
x = 1.80000; y = 57.336600
x = 1.90000; y = 89.226072
x = 2.00000; y = 144.223142



Вывод

В результате проделанной работы я реализовал метод Рунге-Кутты 4-го порядка для решения обыкновенных дифференциальных уравнений первого порядка.

Задача Коши состоит в нахождении решения дифференциального уравнения, удовлетворяющего начальным данным.

Методы делятся на одношаговые и многошаговые. Первые проще в реализации и быстрее вычисляют одну итерацию, вторые же точнее и устойчивее. Кроме того, многошаговые методы для решения задачи Коши требуют вычисления нескольких первых точек при помощи некоторого одношагового метода.

- Метод Эйлера. Одношаговый. Прост в реализации, быстр на одну итерацию, но сильно теряет в точности и устойчивости по сравнении с другими.
- Усовершенствованный метод Эйлера. Одношаговый. Слегка сложнее в реализации, и медленнее на итерацию, но гораздо более точный и устойчивый, чем метод выше. Это достигается с помощью корректировки изменений y через использование дополнительной точки в середине отрезка.
- Метод Рунге-Кутты 4-го порядка. Одношаговый. Использует ещё больше промежуточных значений, чем предыдущие, поэтому сложнее вычислять каждую итерацию, но значительно точнее и устойчивее.
- Метод Адамса. Многошаговый. Основан на использовании полинома Лагранжа, использует в вычислениях четыре предыдущие точки. Имеет высокую точность и устойчивость.
- Метод Милна. Многошаговый. Очень похож на предыдущий, но основан на использовании полинома Ньютона.