Санкт-Петербургский национальный
исследовательский университет ИТМО

Факультет программной инженерии и компьютерной техники
Направление подготовки 09.03.04 «Программная инженерия»
Дисциплина «Вычислительная математика»

Отчет
По лабораторной работе №5
«Метод Рунге-Кутта 4 порядка»

Выполнил студент:
Бабушкин А.М. (P3221)
Преподаватель:
Перл О.В.

Санкт-Петербург
2024

Описание численного метода:
Метод Рунге-Кутты 4-го порядка является численным методом для решения обыкновенных дифференциальных уравнений (ОДУ) по начальным условиям. Он обеспечивает более высокую точность и стабильность по сравнению с более простыми методами, такими как метод Эйлера.

Приближенное значение во всех точках вычисляется по итерационной формуле:

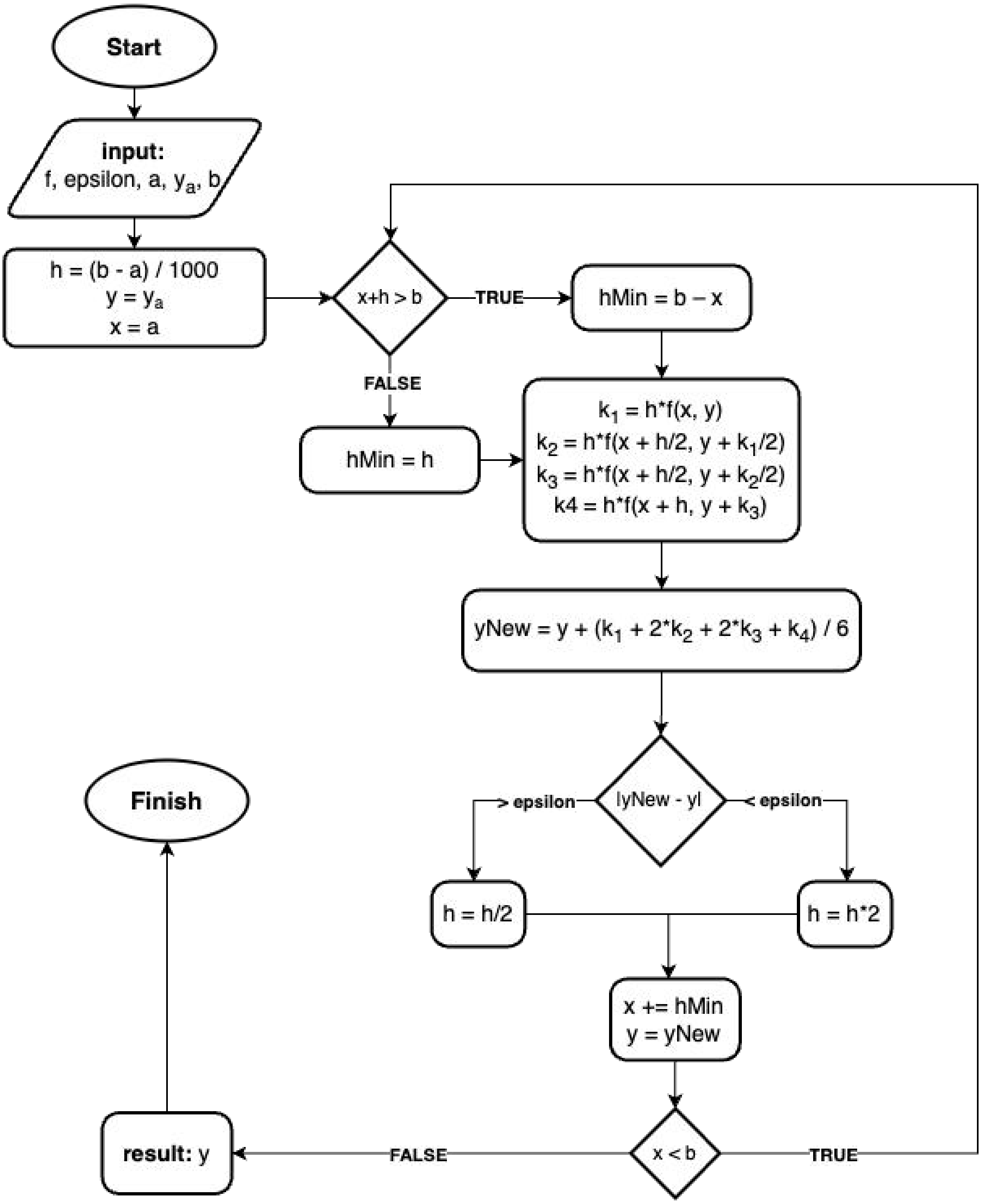
$$\mathbf{y}_{n+1} = \mathbf{y}_n + \frac{h}{6}(\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4)$$

Вычисление нового значения проходит в четыре стадии:

$$\begin{aligned}
\mathbf{k}_1 &= \mathbf{f}\left(x_n, \mathbf{y}_n\right), \\
\mathbf{k}_2 &= \mathbf{f}\left(x_n + \frac{h}{2}, \mathbf{y}_n + \frac{h}{2}\mathbf{k}_1\right), \\
\mathbf{k}_3 &= \mathbf{f}\left(x_n + \frac{h}{2}, \mathbf{y}_n + \frac{h}{2}\mathbf{k}_2\right), \\
\mathbf{k}_4 &= \mathbf{f}\left(x_n + h, \mathbf{y}_n + h\,\mathbf{k}_3\right).
\end{aligned}$$

где $\boldsymbol{h}$ — величина шага сетки по $\boldsymbol{x}$.

Блок-схема:


Start
input:
f, epsilon, a, ya, b
h = (b - a) / 1000
y = ya
x = a
x+h > b
TRUE
hMin = b – x
FALSE
hMin = h
k1 = h*f(x, y)
k2 = h*f(x + h/2, y + k1/2)
k3 = h*f(x + h/2, y + k2/2)
k4 = h*f(x + h, y + k3)
yNew = y + (k1 + 2*k2 + 2*k3 + k4) / 6
|yNew - y|
> epsilon
< epsilon
h = h/2
h = h*2
x += hMin
y = yNew
x < b
TRUE
FALSE
result: y
Finish

Метод реализованный на языке Java:

```java
public static double solveByRungeKutta(int f, double epsilon, double a, double y_a, double b) {
    double h = (b - a) / 1000;
    double y = y_a;
    double x = a;

    while (x < b) {
        var hMin = (x + h > b) ? (b - x) : h;
        double k1 = hMin * get_function(f).apply(x, y);
        double k2 = hMin * get_function(f).apply(x + hMin/2, y + k1/2);
        double k3 = hMin * get_function(f).apply(x + hMin/2, y + k2/2);
        double k4 = hMin * get_function(f).apply(x + hMin, y + k3);
        var yNew = y + (k1 + 2 * k2 + 2 * k3 + k4) / 6;

        if (Math.abs(yNew - y) > epsilon) {
            h /= 2;
        } else if (Math.abs(yNew - y) < epsilon) {
            h *= 2;
        }

        x += hMin;
        y = yNew;
    }

    return y;
}
```

*(Добавил адаптацию **h** для каждого шага, так как были неточные результаты)*

Тесты:

Тест 1

Ввод	Вывод
1	1.9999999999992366
0.00001	
0	
0	
3.14159265358979	

Тест 2

Ввод	Вывод
2	0.0
0.05	
1	
0	
2	

Тест 3

Ввод	Вывод
3	1.7320508075689305
0.001	
0	
1	
1	

Тест 4

Ввод	Вывод
4	103.1963000684839
0.001	
1	
0	
5	

Тест 5

Ввод	Вывод
99	66.0
0.000333	
1	
66	
2	

Тест 6

Ввод	Вывод
1	0.050265665152662606
0.001	
0	
0	
2048	

Вывод:

В ходе этой лабораторной работы я изучил метод Рунге-Кутты 4-го порядка, который является мощным инструментом для численного решения обыкновенных дифференциальных уравнений (ОДУ).

В сравнении с другими методами, такими как метод Эйлера, метод Рунге-Кутты 4-го порядка обеспечивает более высокую точность вычислений за счет использования более сложной формулы вычисления коэффициентов. Этот метод также более стабилен и обеспечивает более плавные изменения значений функции на больших интервалах.

Метод Рунге-Кутты 4-го порядка имеет алгоритмическую сложность O(n), где n - количество точек разбиения отрезка интегрирования. Это означает, что метод является достаточно эффективным с точки зрения вычислительной сложности.

Анализ численной ошибки метода показывает, что она зависит от шага сетки h. Чем меньше значение h, тем меньше численная ошибка и больше точность вычислений.

В целом, метод Рунге-Кутты 4-го порядка является мощным инструментом для численного решения обыкновенных дифференциальных уравнений и может быть применен в различных областях.