

1. Постановка задачи

Найти приближенное решение задачи Коши (1)(2) методом Рунге-Кутты третьего (3) порядка с минимальной локальной погрешностью. Метод Рунге-Кутты четвертого порядка (4) используется для уточнения решения и оценки локальной погрешности на шаге. Оценить локальную погрешность на основе комбинации методов порядка точности 3 и 4.

Задача Коши имеет вид:

$$(1) \quad y' = f(x, y), \quad x \in [A, B]$$

$$(2) \quad y(c) = y_c$$

причем точка c совпадает с началом или концом отрезка интегрирования.

Метод Рунге-Кутты третьего порядка (3):

$$y_1 = y_0 + \frac{1}{4}(K_1 + 3K_3),$$

$$K_1 = hf(x_0, y_0), \quad K_2 = hf\left(x_0 + \frac{1}{3}h, y_0 + \frac{1}{3}K_1\right), \quad K_3 = hf\left(x_0 + \frac{2}{3}h, y_0 + \frac{2}{3}K_2\right),$$

Метод Рунге-Кутты четвертого порядка (4):

$$y_1 = y_0 + \frac{1}{8}(K_1 + 3K_2 + 3K_3 + K_4),$$

$$K_1 = hf(x_0, y_0), \quad K_2 = hf\left(x_0 + \frac{1}{3}h, y_0 + \frac{1}{3}K_1\right),$$

$$K_3 = hf\left(x_0 + \frac{2}{3}h, y_0 - \frac{1}{3}K_1 + K_2\right), \quad K_4 = hf(x_0 + h, y_0 + K_1 - K_2 + K_3)$$

2. Алгоритм решения

2.1 На вход алгоритму подаются значения $A, B, C, y_c, h_{\min}, h_{\max}, e$, где A, B - соответственно начало и конец отрезка интегрирования, C - точка совпадающая с началом или концом отрезка интегрирования, y_c - начальное значение искомой функции в точке c , h_{\min}, h_{\max} - минимальный и максимальный соответственно допустимый шаг интегрирования, e - максимальная допустимая абсолютная погрешность. Входные данные считываются из файла.

Проверка правильности входных данных проверяется в отдельной процедуре.

2.2 Длина самого первого шага интегрирования берется равной $(B-A)/10$.

2.3 Идея основного алгоритма в том, что значения искомой функции вычисляются последовательно от точки к точке, начиная с x_0 и заканчивая точкой x_n . Значение вычисляются в соответствии с заданными методами Рунге-Кутты (3) и (4).

Пусть x_n - точка, в которой приближенное решение известно. Для получения следующей точки в соответствии с методом выбора максимальной длины шага для заданной точности ε определяется рекомендуемая длина шага h_ε .

Алгоритм метода выбора максимальной длины шага h_ε :

Считается погрешность на шаге ε_{n+1} (п. 2.4), сравнивается с заданным значением погрешности ε .

Если абсолютное значение погрешности ε_{n+1} превосходит значение заданной погрешности ε , то шаг h считается неудовлетворительным и выбирается новый шаг интегрирования h_ε из соотношения (5)

$$(5) \quad h_\varepsilon = \alpha h,$$

где параметр α вычисляется по формуле (6) и введен для того, чтобы главная часть локальной погрешности была точно равна заданному ε .

$$(6) \quad \alpha = \sqrt[s+1]{\frac{\varepsilon}{|\varepsilon_{n+1}|}} < 1.$$

Новым узлом интегрирования будет являться узел $x_{n+1} = x_n + h_\varepsilon$.

Если абсолютное значение локальной погрешности ε_{n+1} не превосходит заданного ε , шаг h считается удовлетворительным и в качестве следующего узла интегрирования принимается узел $x_{n+1} = x_n + h$, при этом также определяется шаг h_ε по формуле (5), где α (6) будет уже больше единицы, т.е. шаг h_ε будет больше шага h .

Дальнейшее интегрирование уравнения (1) из точки x_{n+1} начинается с проверки удовлетворительности шага h_ε .

В качестве реального шага интегрирования из точки x_n принимается значение:

$$h_n = \begin{cases} h_{\min}, & \text{если } h_\varepsilon \leq h_{\min} \\ h_\varepsilon, & \text{если } h_{\min} < h_\varepsilon \leq h_{\max} \\ h_{\max}, & \text{если } h_\varepsilon > h_{\max} \end{cases}$$

2.4 Оценка локальной погрешности

Оценка локальной погрешности производится на основе комбинации методов порядка точности 3 и 4.

Для вычисления локальной погрешности ρ^3 метода 3-го порядка (3) используется выражение вида

$$\rho^s \cong y_1^p - y_1^s = \sum_{i=1}^r q_i k_i$$

где s - порядок метода, y_1^p - вычисленное значение y_1 методом порядка p , y_1^s - вычисленное значение y_1 методом порядка s , причем $p > s$.

Полученная оценка погрешности требует 6 вычислений правой части выражения (1).

2.5 Проверка вычисленного шага h на конец интервала.

Для каждого вычисленного шага h_n делается проверка на конец интервала с тем, чтобы последний шаг не оказался слишком малым.

Алгоритм выбора двух последних шагов у конца отрезка интегрирования:

Итерационное вычисление с методом выбором шага, описанным в п.2.4 прекращается за два шага до конца отрезка интегрирования.

Вычисляется расстояние от текущей точки интегрирования x_i и заданной точки B . Полученное расстояние делится на равные отрезки, при этом проверяется условия, что вычисляемый шаг h меньше h_{\max} и больше h_{\min} . Последние два шага будут равными вычисленному h .

2.6 Замечания по реализации

2.6.1 После вычисления очередного приближенного значения решения оно сразу выводится в файл.

2.6.2 Параметр α вычисляемый по формуле (6) заменяется на параметр $\alpha^* = 0,9\alpha$.

3. Основные процедуры

`integrator_3(double x, double y, double h)` - процедура принимается для вычисления y_{i+1} методом Рунге-Кутты 3-го порядка (3). На вход процедуре передаются значения значение точки интегрирования x_i , y_i и шаг h . Процедура возвращает вычисленное значение y_{i+1} .

`integrator_4(double x, double y, double h)` - процедура принимается для вычисления y_{i+1} методом Рунге-Кутты 4-го порядка (4). На вход процедуре передаются значения значение точки интегрирования x_i , y_i и шаг h . Процедура возвращает вычисленное значение y_{i+1} .

`main(string filename)` - процедура принимает на вход имя файла входных данных и реализует основную часть алгоритма, описанного в ходе решения(п.2). Процедура возвращает код ошибки $icod = 2$, при ошибке входных данных. Результаты вычислений записываются в файл в соответствии с замечанием 2.6.1.

`new_h(double eps_1, double eps, double h)` - процедура принимает на вход вычисленное значение локальной погрешности eps_1 , заданное значение локальной погрешности eps и значение текущего шага интегрирования h . Процедура реализует вычисление нового шага интегрирования h на основе алгоритма, описанного в п.2.3.

`testInput(string filename)` - процедура принимает на вход имя файла входных данных и проверяет корректность полученных данных. В случае ошибки возвращает код `icod = 2`. При корректных входных данных возвращает обработанные данные.