Дисциплина: Численные методы

Лабораторное задание №5

Задание 11. Вариант 1

**Отчёт**

Тема: «Решение задачи Коши с заданной точностью с автоматическим выбором максимальной длины шага»

Выполнил:  
студент 3 курса 62 группы  
Дорохов М.В.

Проверила:  
старший преподаватель  
Фролова О.А.

**1. Постановка задачи**

Разработать программу для численного решения задачи Коши для обыкновенного дифференциального уравнения (ОДУ) первого порядка с заданной точностью и автоматическим выбором шага интегрирования методом выбора максимальной длины шага. Задача Коши формулируется следующим образом:

Где C – начальная точка, совпадающая либо с A, либо с B.

**Входные параметры:**

* – границы интервала интегрирования, начальная точка и начальное значение;
* – минимальный и максимальный шаги интегрирования, допустимая локальная погрешность;

**Выходные параметры:**

* Вектор точек ,   
  где –приближённое решение, – локальная погрешность;
* Выходной файл rez, где первая и последующие строки –   
  X-координата точки интегрирования, полученное приближенное значение и минимальная погрешность в этой точке;
* Icod – индикатор ошибки:
  + Icod = 0 – нет ошибки, решение получено;
  + Icod = 1 – требуемая точность не достигнута, решение получено с меньшей точностью;
  + Icod = 2 – ошибка входных данных.

**2. Алгоритм решения**

**2.1. Метод Рунге-Кутта второго порядка**

Для вычисления приближённого решения используется явный метод Рунге-Кутта второго порядка (формула 121 из методички):

где:

Этот метод имеет порядок точности , что обеспечивает достаточно точное приближение при умеренных шагах.

**2.2. Метод Рунге-Кутта третьего порядка**

Для оценки локальной погрешности применяется метод Рунге-Кутта третьего порядка (формула 30 из методички):

Где:

Локальная погрешность вычисляется как , где и – значения, полученные методами второго и третьего порядка соответственно.

**2.3. Автоматический выбор шага**

Метод выбора максимальное длины шага (раздел 5.2 методички) адаптирует шаг так, чтобы локальная погрешность оставалась близкой к заданной :

* Если шаг уменьшается: ;
* Если шаг может увеличиться по той же формуле, но ограничивается .

Шаг выбирается следующим образом:

* если ;
* если ;
* если .

Для последнего шага у конца интервала шаг корректируется, чтобы точно достигло или превысило его не более чем на

**3. Реализация в коде**

Для решения задачи Коши с использованием методов Рунге-Кутта и адаптивного выбора шага была написана программа на Rust. Ключевые компонентов: функции для численных методов, алгоритм интегрирования с адаптацией шага, обработку входных и выходных данных, а также базовые механизмы обработки ошибок.

**3.1. Функции методов Рунге-Кутта**

Для численного решения задачи Коши используются два метода Рунге-Кутта: метод второго порядка (RK2) для получения основного приближения и метод третьего порядка (RK3) для оценки локальной погрешности. Эти методы реализованы в виде отдельных функций.

* Функция **fn rk2\_step(x: f64, y: f64, h: f64, f: &dyn Fn(f64, f64) -> f64) -> f64**:
  + Метод Рунге-Кутта второго порядка (RK2) используется для вычисления следующего значения на основе текущих и ;
  + и – промежуточные коэффициенты;
  + – шаг интегрирования;
  + f (&dyn Fn(f64, f64) -> f64)) – функция, представляющая правую часть дифференциального уравнения .
* Функция **fn rk3\_step(x: f64, y: f64, h: f64, f: &dyn Fn(f64, f64) -> f64) -> f64**:
  + Метод Рунге-Кутта третьего порядка (RK3) применяется для вычисления альтернативного приближения, что позволяет оценить локальную погрешность;
  + Локальная погрешность оценивается как разность между результатами RK3 и RK2: .

**3.2. Основной алгоритм**

Программа реализует численное интегрирование задачи Коши на заданном интервале с адаптацией шага для достижения требуемой точности . Алгоритм включает следующие этапы:

1. **Чтение входных данных**  
   Входные параметры считываются из файла формата:
   * (границы интервала, начальная точка и значение)
   * (минимальный и максимальный шаги, требуемая точность)
2. **Проверка корректности параметров**Перед начало вычислений проверяются условия:
   * должно равняться либо , либо (начальная точка интегрирования);
   * ;
   * (интервал должен быть ненулевым).
3. **Инициализация**
   * Начальный шаг задаётся как ;
   * Направление интегрирования определяется как:
     + Если , интегрирование идёт вперёд (от к );
     + Если , интегрирование идёт вперёд (от к ).
4. **Цикл интегрирования**Основной цикл выполняется, пока текущая точка не достигнет конца интервала . На каждом шаге:
   * **Вычисление пробного шага**Если следующий шаг выходит за пределы интервала, корректируется так, чтобы точно достичь но не менее .
   * **Оценка погрешности**Вычисляются и с шагом , а затем погрешность:  
     **.**
   * **Принятие или адаптация шага**
     + Если или шаг приводит к концу интервала, шаг принимается: .
     + Новый шаг адаптируется по формуле:  
       ,  
       но ограничивается пределами .
     + Если погрешность превышает , шаг уменьшается и вычисления повторяются.
5. **Запись результатов**На каждом принятом шаге в файл **rez** записываются значения В конце выводятся итоговые данные (общее количество шагов и т. д.)

**3.3. Обработка ошибок**

Программа предусматривает обработку исключительных ситуаций:

* Если не равно ни , ни , программа завершается с кодом ошибки
* Если параметры или некорректны (например, отрицательны), также возвращается .
* Если , работа завершается с кодом ошибки .
* В случае, если точность не достигнута на некоторых шагах, код .

**4. Тестирование**

**Тест 1. Стандартный случай**

* **Данные:**
* **Результаты:**
  + Конец:
  + Погрешность
  + Абсолютная ошибка:
  + Шаги: 11, 0 неточных, 1 min, 10 max,

**Тест 2. Малый интервал**

* **Данные:**
* **Результаты:**
  + Конец:
  + Погрешность
  + Абсолютная ошибка:
  + Шаги: 2, 0 неточных, 1 min, 0 max,

**Тест 3. Большой интервал**

* **Данные:**
* **Результаты:**
  + Конец:
  + Погрешность
  + Абсолютная ошибка:
  + Шаги: 44, 0 неточных, 1 min, 0 max,

**Тест 4. Некорректные параметры**

* **Данные:** (отрицательная погрешность)
* **Результаты:**
  + Шаги: 0,

**Тест 5. Строгие ограничения к точности с минимальным шагом**

* **Данные:**
* **Результаты:**
  + Начало:
  + Конец: погрешность
  + Абсолютная ошибка:
  + Шаги: 12, 11 неточных, 11 min, 0 max,

**Тест 6. Высокая точность с малым минимальным шагом**

* **Данные:**
* **Результаты:**
  + Начало:
  + Конец: погрешность
  + Абсолютная ошибка:
  + Шаги: 99, 0 неточных, 0 min, 96 max,

**Тест 7. Некорректные входные параметры (min > max)**

* **Данные:**
* **Результаты:**
  + Шаги: 0,