НАЦИОНАЛЬНЫЙ

ИССЛЕДОВАТЕЛЬСКИЙ

УНИВЕРСИТЕТ ИТМО

Факультет программной инженерии и компьютерной техники

Направление подготовки 09.03.04 - Программная инженерия

Дисциплина – Вычислительная математика

Лабораторная работа №6

Вариант №11

Выполнил: Мухсинов С.П

Группа: P3217

Преподаватель: Малышева Т.А

Санкт-Петербург

2024

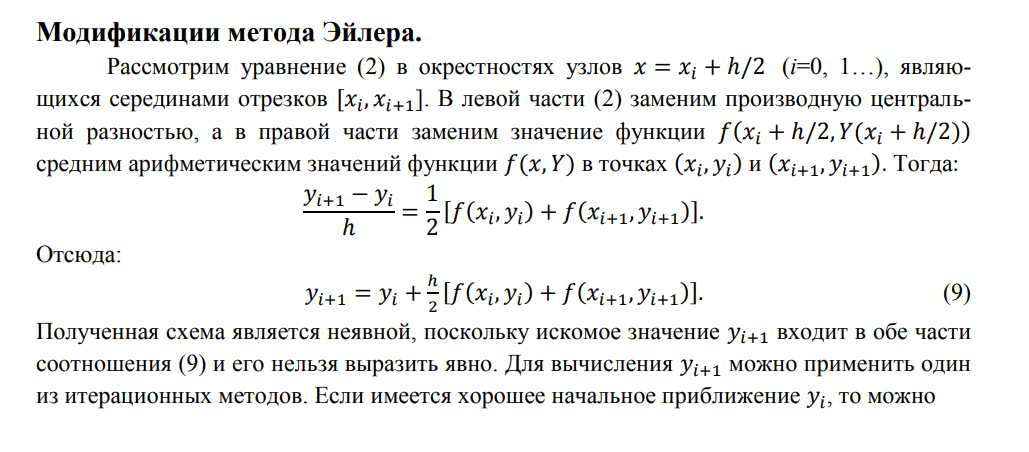
**Цель работы**

Решить задачу Коши для обыкновенных дифференциальных уравнений численными методами.

**Задание**

1. В программе численные методы решения обыкновенных дифференциальных уравнений (ОДУ) должен быть реализован в виде отдельного класса /метода/функции;
2. Пользователь выбирает ОДУ вида(не менее трех уравнений), из тех, которые предлагает программа;
3. Предусмотреть ввод исходных данных с клавиатуры: начальныеусловия , интервал дифференцирования , шаг h, точность ;
4. Для исследования использовать одношаговые методы и многошаговые методы (см. табл.1);
5. Составить таблицу приближенных значений интеграла дифференциального уравнения, удовлетворяющего начальным условиям, для всех методов, реализуемых в программе;
6. Для оценки точности одношаговых методов использовать правило Рунге;
7. Для оценки точности многошаговых методов использовать точное решение задачи.
8. Построить графики точного решения и полученного приближенного решения (разными цветами);
9. Программа должна быть протестирована при различных наборах данных, в том числе и некорректных.
10. Проанализировать результаты работы программы

**Используемые формулы и методы**

****

**Изображение выглядит как текст, снимок экрана, Шрифт, алгебра

Автоматически созданное описание**

**Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание**

**Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание**

**Изображение выглядит как текст, снимок экрана, Шрифт, документ

Автоматически созданное описание**

**Изображение выглядит как текст, Шрифт, снимок экрана, число

Автоматически созданное описание**

**Изображение выглядит как текст, снимок экрана, Шрифт, информация

Автоматически созданное описание**

**Изображение выглядит как текст, Шрифт, снимок экрана, число

Автоматически созданное описание**

**Программная реализация**

**Метод Эйлера (Модифицированный)**

@Override  
 protected double[] calculate(double h1) {  
 int n1 = (int) ((xn - x0) / h1);  
 double[] x1 = new double[n1 + 1];  
 double[] y1 = new double[n1 + 1];  
 x1[0] = x0;  
 y1[0] = y0;  
  
 for (int i = 1; i <= n1; i++) {  
 x1[i] = x0 + i \* h1;  
 }  
 for (int i = 1; i <= n1; i++) {  
 y1[i] = y1[i - 1] + (h1 / 2) \* (f.at(x1[i - 1], y1[i - 1]) +  
 f.at(x1[i], y1[i - 1] + h1 \* f.at(x1[i - 1], y1[i - 1])));  
  
 }  
 return y1;  
 }  
}

**Метод Рунге-Кутта 4-го порядка**

@Override  
 protected double[] calculate(double h1) {  
 int n1 = (int) ((xn - x0) / h1);  
 double[] x1 = new double[n1 + 1];  
 double[] y1 = new double[n1 + 1];  
 x1[0] = x0;  
 y1[0] = y0;  
  
 for (int i = 1; i <= n1; i++) {  
 x1[i] = x0 + i \* h1;  
 }  
  
 for (int i = 1; i <= n1; i++) {  
 double k1, k2, k3, k4;  
 k1 = h1 \* f.at(x1[i - 1], y1[i - 1]);  
 k2 = h1 \* f.at(x1[i - 1] + h1 / 2, y1[i - 1] + k1 / 2);  
 k3 = h1 \* f.at(x1[i - 1] + h1 / 2, y1[i - 1] + k2 / 2);  
 k4 = h1 \* f.at(x1[i - 1] + h1, y1[i - 1] + k3);  
 y1[i] = y1[i - 1] + (k1 + 2 \* k2 + 2 \* k3 + k4) / 6;  
 }  
 return y1;  
 }  
}

**Метод Адамса**

@Override  
 public Result solve() {  
 return new Result(x, calculate(h));  
 }  
  
 @Override  
 protected double[] calculate(double h1) {  
 Result result = new Runge(x0, xn, y0, h1, eps, f, 4).solve();  
 y[1] = result.getY()[1];  
 y[2] = result.getY()[2];  
 y[3] = result.getY()[3];  
 for (int i = 4; i <= n; i++) {  
 double yp = y[i-1] + (h1/24) \* (55\*f.at(x[i-1], y[i-1]) - 59\*f.at(x[i-2], y[i-2]) +  
 37\*f.at(x[i-3], y[i-3]) - 9\*f.at(x[i-4], y[i-4]));  
 double yk = y[i-1] + (h1/24) \* (9\*f.at(x[i], yp) + 19\*f.at(x[i-1], y[i-1]) -  
 5\*f.at(x[i-2], y[i-2]) + f.at(x[i-3], y[i-3]));  
 while (Math.*abs*(yk-yp) > eps){  
 yp = yk;  
 yk = y[i-1] + (h1/24) \* (9\*f.at(x[i], yp) + 19\*f.at(x[i-1], y[i-1]) -  
 5\*f.at(x[i-2], y[i-2]) + f.at(x[i-3], y[i-3]));  
 }  
 y[i] = yk;  
 };  
 return y;  
 }  
}

**Результаты работы программы**

**Пример №1**

**Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание**

**Изображение выглядит как текст, снимок экрана, Шрифт, дизайн

Автоматически созданное описание**

**Изображение выглядит как текст, снимок экрана, Шрифт, дизайн

Автоматически созданное описаниеИзображение выглядит как текст, Шрифт, снимок экрана

Автоматически созданное описание**

**Изображение выглядит как текст, линия, График, снимок экрана

Автоматически созданное описание**

**Пример №2**

**Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание**

**Изображение выглядит как текст, снимок экрана, Шрифт, документ

Автоматически созданное описание**

**Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание**

**Изображение выглядит как текст, линия, График, число

Автоматически созданное описание**

**Пример №3**

**Изображение выглядит как текст, снимок экрана, Шрифт, документ

Автоматически созданное описание**

**Изображение выглядит как текст, снимок экрана, Шрифт, документ

Автоматически созданное описание**

**Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание**

**Изображение выглядит как линия, График, текст, диаграмма

Автоматически созданное описание**

**Вывод**

В процессе выполнения лабораторной работы была реализована программа, решающая задачу Коши численными методами Рунге-Кутта 4-го порядка, Адамса, Эйлера (Модифицированного).