ИТМО

ПИИКТ

**Лабораторная работа №4**

**“Вычислительная математика”**

Группа Р3201

**Метод Адамса**

Выполнил: Братчиков Иван Станиславович

Приняла: Перл Ольга Вячеславовна уанщк

Санкт-Петербург

2020

**Описание метода:**

Метод Адамса имеет четвёртый порядок точности. Используемая в нём формула прогноза получена интегрированием обратной интерполяционной формулы Ньютона и имеет вид:

A screenshot of a cell phone

Description automatically generated

**Код вычислительного метода:**

private void setInitialData(double[] xData, double[] yData, double h){  
 for(int i = 1; i < 4; i++){  
 double x = xData[i - 1];  
 double y = yData[i - 1];  
  
 double k0 = function.getValue(x, y);  
 double k1 = function.getValue(x + h / 2, y + h / 2 \* k0);  
 double k2 = function.getValue(x + h / 2, y + h / 2 \* k1);  
 double k3 = function.getValue(x + h, y + h \* k2);  
  
 yData[i] = y + h / 6 \* (k0 + 2 \* k1 + 2 \* k2 + k3);  
 }  
}  
  
public double[][] solve(double x0, double y0, double endPoint, double accuracy){  
 int intervals = 5;  
 double[] xData;  
 double[] yData;  
 double averageError;  
 double segmentLength = endPoint - x0;  
 do {  
 intervals \*= 2;  
 if(intervals > 10000000)  
 throw new SolutionException("Достигнут максимум разбиений");  
 double h = segmentLength / intervals;  
  
 xData = new double[intervals + 1];  
 xData[0] = x0;  
 yData = new double[intervals + 1];  
 yData[0] = y0;  
  
 for(int i = 1; i < xData.length; i++)  
 xData[i] = xData[i - 1] + h;  
 setInitialData(xData, yData, h);  
  
 double[] error = new double[yData.length - 4];  
 System.out.println(Arrays.toString(yData));  
 System.out.println(yData.length);  
 for(int i = 4; i < yData.length; i++){  
 double prediction = yData[i - 1] + (1 \* h / 24) \* (55 \* function.getValue(xData[i - 1], yData[i - 1])  
 - 59 \* function.getValue(xData[i - 2], yData[i - 2])  
 + 37 \* function.getValue(xData[i - 3], yData[i - 3]) - 9 \* function.getValue(xData[i-4], yData[i-4]));  
  
 yData[i] = yData[i-1] + h / 24 \* (- 5 \* function.getValue(xData[i - 2], yData[i - 2])  
 + 19 \* function.getValue(xData[i - 1], yData[i - 1])  
 + 9 \* function.getValue(xData[i], prediction) + function.getValue(xData[i - 3], yData[i - 3]));  
 if(!Double.isFinite(yData[i]))  
 throw new SolutionException(String.format("При аргументе, лежащем внутри отрезка от %f до %f, " +  
 "значения функции выходят за пределы допустимых значений",  
 Math.min(x0, endPoint), Math.max(x0, endPoint)));  
 error[i - 4] = Math.abs(yData[i] - prediction);  
 }  
 averageError = Arrays.stream(error).average().orElse(0);  
 } while (averageError >= accuracy);  
 return new double[][]{xData, yData};  
}

A close up of text on a white background

Description automatically generatedA close up of text on a white background

Description automatically generated **Блок схема:**

**Пример:**

**A screenshot of a computer

Description automatically generated**

**Вывод:**

Методы Адамса k {\displaystyle k} k-го порядка требуют предварительного вычисления решения в k {\displaystyle k} k начальных точках. Для вычисления начальных значений обычно используют одношаговые методы, например, 4-стадийный метод Рунге — Кутты 4-го порядка точности. Расчёты выполняются так же, как и по методу Милна, однако в отличие от последнего ошибка, внесённая на каком-либо шаге, не имеет тенденции к экспоненциальному росту.