Федеральное государственное автономное образовательное учреждение высшего образования

«Национальный исследовательский университет ИТМО»

Факультет Программной Инженерии и Компьютерной Техники

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

A picture containing logo

Description automatically generated

Лабораторная работа №6 по дисциплине  
«Вычислительная математика»

" ЧИСЛЕННОЕ РЕШЕНИЕ ОБЫКНОВЕННЫХ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ»"

Выполнил: Дау Конг Туан Ань

Группа: P32151

Преподаватель: Машина Е.А

г. Санкт-Петербург

2023

**Цель лабораторной работы**: решить задачу Коши для обыкновенных дифференциальных уравнений численными методами.

Text

Description automatically generated with low confidence

Text

Description automatically generated

Text, letter

Description automatically generated

Chart

Description automatically generated

Text

Description automatically generated

Text, letter

Description automatically generated

Text

Description automatically generated

Text, letter

Description automatically generated

Text, letter

Description automatically generated

Text

Description automatically generated

1. Программная реализация задачи
2. private final String algo = "Adam";  
   @Override  
   public Double[][] execute(InputSet inputSet, Function func) {  
    func.ResolveC(inputSet.getA(), inputSet.getY0());  
    int n = (int) ((inputSet.getB()- inputSet.getA())/ inputSet.getH());  
    Double[][] tb = new Double[n][5];  
    for(int i = 0; i < n; ++i) {  
    tb[i][0] = inputSet.getA() + i\* inputSet.getH();  
    }  
    tb[0][1] = inputSet.getY0();  
    tb[0][2] = func.ValueAt(tb[0][0] , tb[0][1]);  
    tb[0][3] = func.yAt(tb[0][0]);  
    tb[0][4] = Math.*abs*(tb[0][1] - tb[0][3]);  
    for(int i = 1; i < n;++i) {  
    Double[] f = new Double[4];  
    f[0] = func.ValueAt(tb[i-1][0], tb[i-1][1]);  
    if(i == 1) f[1] = 0.0;  
    else f[1] = func.ValueAt(tb[i-2][0], tb[i-2][1]);  
    if(i <= 2) f[2] = 0.0;  
    else f[2] = func.ValueAt(tb[i-3][0], tb[i-3][1]);  
    if(i <= 3) f[3] = 0.0;  
    else f[3] = func.ValueAt(tb[i-4][0], tb[i-4][1]);  
    Double deltaFx = f[0] - f[1];  
    Double deltaFx1 = f[0] - 2\*f[1] + f[2];  
    Double deltaFx2 = f[0] - 3\*f[1] + 3\*f[2] - f[3];  
    tb[i][1] = tb[i-1][1]  
    + inputSet.getH() \* func.ValueAt(tb[i-1][0], tb[i-1][1])  
    + Math.*pow*(inputSet.getH(), 2) / 2 \* deltaFx  
    + 5\* Math.*pow*(inputSet.getH(), 3)/12 \* deltaFx1  
    + 3\* Math.*pow*(inputSet.getH(), 4)/8 \* deltaFx2;  
    }  
     
    for(int i = 1; i < n; ++i) {  
    tb[i][2] = func.ValueAt(tb[i][0], tb[i][1]);  
    tb[i][3] = func.yAt(tb[i][0]);  
    tb[i][4] = Math.*max*(tb[i - 1][4], Math.*abs*(tb[i][3] - tb[i][1]));  
    }  
     
    return tb;  
   }

private final String algo = "Advanced Euler";  
  
@Override  
public Double[][] execute(InputSet inputSet, Function func) {  
 func.ResolveC(inputSet.getA(), inputSet.getY0());  
 int n = (int) ((inputSet.getB()- inputSet.getA())/ inputSet.getH());  
 Double[][] tb = new Double[n][5];  
 for(int i = 0; i < n; ++i) {  
 tb[i][0] = inputSet.getA() + i \* inputSet.getH();  
 }  
 tb[0][1] = inputSet.getY0();  
 for(int i = 1; i < n; ++i) {  
 Double fxy = func.ValueAt(tb[i-1][0], tb[i - 1][1]);  
 tb[i][1] = tb[i- 1][1] +  
 (inputSet.getH()/2) \*  
 (fxy + func.ValueAt(tb[i][0],tb[i-1][1] + inputSet.getH()) \*fxy);  
 }  
  
 for(int i = 0 ;i < n;++i) {  
 tb[i][2] = func.ValueAt(tb[i][0], tb[i][1]);  
 tb[i][3] = func.yAt(tb[i][0]);  
 tb[i][4] = Math.*abs*((Math.*pow*(tb[i][1], inputSet.getH())  
 - Math.*pow*(tb[i][1], inputSet.getH() / 2))  
 /(Math.*pow*(2, i + 1) - 1));  
 }  
 return tb;  
}

private String algo = "Runge - Kytta";  
@Override  
public Double[][] execute(InputSet inputSet, Function func) {  
 func.ResolveC(inputSet.getA(), inputSet.getY0());  
 int n = (int) ((inputSet.getB()- inputSet.getA())/ inputSet.getH());  
 Double[][] tb = new Double[n][5];  
 for(int i = 0; i < n; ++i) {  
 tb[i][0] = inputSet.getA() + i\* inputSet.getH();  
 }  
 tb[0][1] = inputSet.getY0();  
 for(int i = 1 ;i < n;++i) {  
 Double k1 = func.ValueAt(tb[i - 1][0], tb[i - 1][1]);  
 Double k2 = func.ValueAt(tb[i - 1][0] + inputSet.getH()/ 2,tb[i - 1][1] + k1/2 );  
 Double k3 = func.ValueAt(tb[i - 1][0] + inputSet.getH()/ 2,tb[i - 1][1] + k2/2 );  
 Double k4 = func.ValueAt(tb[i - 1][0] + inputSet.getH(), tb[i-1][1] + k3);  
 tb[i][1] = tb[i - 1][1]  
 + (inputSet.getH()/6)  
 \* (k1 + 2\*k2 +2\*k3 + k4);  
 }  
  
 for(int i = 0 ;i < n; ++i) {  
 tb[i][2] = func.ValueAt(tb[i][0], tb[i][1]);  
 tb[i][3] = func.yAt(tb[i][0]);  
 tb[i][4] = Math.*abs*((Math.*pow*(tb[i][1], inputSet.getH())  
 - Math.*pow*(tb[i][1], inputSet.getH() / 2))  
 /(Math.*pow*(2, i + 1) - 1));  
 }  
  
 return tb;  
}

**C:\Users\Never\.jdks\openjdk-16.0.2\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2023.1\lib\idea\_rt.jar=58917:C:\Program Files\JetBrains\IntelliJ IDEA 2023.1\bin" -Dfile.encoding=UTF-8 -classpath C:\Users\Never\IdeaProjects\Lab6\_CM\out\production\Lab6\_CM;C:\Users\Never\.m2\repository\org\jfree\jfreechart\1.5.4\jfreechart-1.5.4.jar Main**

**Please type Y\_0:**

**-1**

**Please type h:**

**0.1**

**Please type a:**

**1**

**Please type b:**

**1.5**

**Please type error:**

**0.1**

**List of Algorithm:**

**1 : Advanced Euler**

**2 : Runge - Kytta**

**3 : Adam**

**Please choose one algorithm:**

**1**

**List Of Function:**

**1 : 2x**

**2 : y + (1+x)y^2**

**3 : 2y\*sin(x)**

**Please choose one function:**

**2**

**x y fxy y\_exact**

**1.0 -1.0 1.0 -1.0**

**1.1 -0.9099 0.8288 -0.909**

**1.2 -0.8422 0.7184 -0.8333**

**1.3 -0.7874 0.6388 -0.7692**

**1.4 -0.7412 0.5774 -0.7142**

**Chart, line chart

Description automatically generated**

**Вывод: узнал много нового, как пользоваться программой для решения дифференциальных уравнений**