

**РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ им.
ПАТРИСА ЛУМУМБЫ**

Факультет физико-математических и естественных наук

**ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №5**

дисциплина: Вычислительные методы

Студент:

Мажитов Магомед Асхабович

Группа:

НКНбд-01-21

МОСКВА

2023 г.

Цель:

Написать программу для численного решения задачи Коши.

Теоретическая справка:

1. Реализовать в программе метод Рунге-Кутты 2-ого порядка точности для численного решения задачи Коши, приведенной ниже:

$$y' = f(x, y), x \in [a, b]$$

$$y(a) = y_0$$

2. В программе вывести таблицу данных следующего вида:

x_0	$y_1(x_0)$	$y(x_0)$	$\delta(x_0)$
x_1	$y_1(x_1)$	$y(x_1)$	$\delta(x_1)$
.	.	.	.
.	.	.	.
x_N	$y_1(x_N)$	$y(x_N)$	$\delta(x_N)$

где целое число $N = 32$ определяет равномерное разбиение отрезка $[a, b]$, $y_1(x_j)$ – значение численного решения задачи Коши, полученного методом, Рунге-Кутты 2-ого порядка точности, в узлах разбиения отрезка; $y(x_j)$ – значение аналитического решения задачи Коши в узлах разбиения отрезка $\delta(x_j)$ – погрешность

3. Определить значение N , при котором $\delta(x_j) < 10^{-2}$
4. Сравнить полученное в п.3 значение N с аналогичным значением, полученным для метода Эйлера. Проанализировать результат.

Листинг:

```
#include <iostream>
#include <cmath>
#include <iomanip>

using namespace std;
//f=2x^3y
//a = 0, b = 2^1/4, y0=1

double f(double x, double y){ //функция для расчета значения f
    return 2*pow(x, 3)*y;
}

double f_analit(double x){ //функция для расчета аналитического значения
    double xx = pow(x, 4)/2;
    return exp(xx);
}
```

```

double* X(double a, double b, int n){ //функция для расчета x
    double h = (b-a)/n; //считаем шаг
    auto* x= new double [n];
    x[0] = a;
    for(int i = 1; i <= n; i++){ //проходим по циклу и считаем x
        x[i] = a + i*h;
    }
    return x;
}

double* El_Y(double a, double b, int n, double* x){ //функция для расчета y по
методу Эйлера
    double h = (b-a)/n; //считаем шаг
    auto* y = new double [n];
    y[0] = 1;
    for(int i = 1; i <= n; i++){ //проходим по циклу и считаем y
        y[i] = y[i-1]+h*f(x[i-1], y[i-1]);
    }
    return y;
}

double* R_K(double a, double b, int n, double* x){ //функция для расчета y по
методу Рунге-Кутты
    double h = (b-a)/n;
    auto* y = new double [n];
    auto* temp = new double [n];
    y[0] = 1;
    for(int i = 1; i <= n; i++){ //проходим по циклу и считаем y
        temp[i] = y[i-1] + h * f(x[i-1], y[i-1]);
        y[i] = y[i-1] + h * ((f(x[i-1], y[i-1]) + f(x[i], temp[i])))/2);
    }
    return y;
}

void print_(double* x, double* y, int n){ //функция для красивой печати
    for(int i = 0; i < n; i++){
        cout << setprecision(5) << fixed << x[i] << setw(10) << y[i] <<
setw(10);
        cout << f_analit(x[i]) << setw(10) << abs(f_analit(x[i]) - y[i])<< endl;
    }
}

void found_N_El(double a, double b, int n){ //функция для нахождения
минимального значения n для метода Эйлера
    bool t = true;
    do{ //проходим по циклу пока погрешность не будет меньше 0.001
        double* x = X(a, b, n);
        double* y = El_Y(a, b, n, x);
        for(int i = 0; i < n; i++) {
            if(abs(f_analit(x[i]) - y[i]) >= 0.01){
                n++;
                t = true;
                break;
            }
            else t = false;
        }
    } while (t);
    cout << "Для метода Эйлера минимальное значение N = " << n << " при котором
погрешность меньше 0,01"<< endl;
}

void found_N(double a, double b, int n){ //функция для нахождения минимального
значения n для метода Рунге-Кутты
    bool t = true;

```

```

do{ //проходим по циклу пока погрешность не будет меньше 0.001
double* x = X(a, b, n);
double* y = R_K(a, b, n, x);
for(int i = 0; i < n; i++) {
    if(abs(f_analit(x[i]) - y[i]) >= 0.01){
        n++;
        t = true;
        break;
    }
    else t = false;
}
} while (t);
cout << "Для метода Рунге-Кутты минимальное значение N = " << n << " при
котором погрешность меньше 0,01"<< endl;
}

int main() {
    int n = 32;
    double a = 0, b = pow(2, 1/4);
    double* x = X(a, b, n);
    double* y = R_K(a, b, n, x);
    print_(x, y, n);
    int m = 2;
    found_N(a, b, m);
    found_N_El(a, b, m);
    return 0;
}

```

Работа программы:

Консоль отладки Microsoft Visual Studio

0.25000	1.00199	1.00196	0.00003
0.28125	1.00317	1.00313	0.00004
0.31250	1.00483	1.00478	0.00005
0.34375	1.00706	1.00701	0.00006
0.37500	1.01001	1.00994	0.00007
0.40625	1.01379	1.01371	0.00008
0.43750	1.01858	1.01849	0.00009
0.46875	1.02454	1.02443	0.00011
0.50000	1.03187	1.03174	0.00012
0.53125	1.04077	1.04063	0.00014
0.56250	1.05149	1.05133	0.00016
0.59375	1.06429	1.06411	0.00018
0.62500	1.07948	1.07928	0.00020
0.65625	1.09740	1.09717	0.00023
0.68750	1.11843	1.11818	0.00025
0.71875	1.14303	1.14275	0.00028
0.75000	1.17171	1.17140	0.00031
0.78125	1.20508	1.20474	0.00034
0.81250	1.24384	1.24347	0.00037
0.84375	1.28882	1.28841	0.00040
0.87500	1.34100	1.34056	0.00044
0.90625	1.40157	1.40110	0.00047
0.93750	1.47194	1.47143	0.00051
0.96875	1.55381	1.55328	0.00053

Для метода Рунге-Кутты минимальное значение N = 6 при котором погрешность меньше 0,01
Для метода Эйлера минимальное значение N = 204 при котором погрешность меньше 0,01

C:\Users\magom\source\repos\ConsoleApplication1\Debug\ConsoleApplication1.exe (процесс 5788) завершил работу с кодом 0.
Чтобы автоматически закрывать консоль при остановке отладки, включите параметр "Сервис" ->"Параметры" ->"Отладка" -> "Автоматически закрыть консоль при остановке отладки".
Нажмите любую клавишу, чтобы закрыть это окно:

Вывод:

В ходе работы я реализовал на языке С++ метод Рунге-Кутты 2-ого порядка точности для численного решения задачи Коши. В результате нахождения минимального значения N при котором погрешность меньше 0,01 для метода Рунге-Кутты и для метода Эйлера можно заметить, что метод Рунге-Кутты является наиболее эффективным для численного решения задачи Коши, чем метод Эйлера.