Федеральное государственное бюджетное образовательное учреждение высшего образования
«Сибирский государственный университет телекоммуникаций и информатики»
(СибГУТИ)

Кафедра вычислительных систем

**ОТЧЕТ**
по курсовой работе

по дисциплине «**Вычислительная математика**»

Выполнил:
студент гр. ИС-142
«__» июня 2023 г. _____________ /Наумов А.А./

Проверил:
преподаватель
«__» июня 2023 г. _____________ /Бублей Д.А./

Оценка «_____________»

Новосибирск 2023

## ОГЛАВЛЕНИЕ

# ПОСТАНОВКА ЗАДАЧИ

Методом Рунге-Кутта 4 порядка решить задачу Коши для ОДУ.

$y(x)' = x + y(x);$

$y(0) = 1;$

на отрезке [0;10] с точностью до 4-х знаков.

## МЕТОД РЕШЕНИЯ ЗАДАЧИ

$$\begin{cases} y' = f(x, y), \\ y(x_0) = y_0. \end{cases} \qquad (1)$$

$$y_{i+1} = y_i + \int\limits_{x_i}^{x_{i+1}} f(x, y(x))dx. \qquad (2)$$

Уравнение (2) эквивалентно задаче Коши (1).

Метод Рунге-Кутта.

Метод Рунге-Кутта имеет погрешность, пропорциональную $h^4$ и поэтому является одним из наиболее употребительных методов численного решения задачи Коши для ОДУ. Для вычисления интеграла в (2) используем квадратную формулу Симпсона. Для формулы Симпсона, необходимо три узла. В качестве недостающего узла возьмем середину отрезка $[x_i, x_{i+1}]$, т.е. $x_{i+1/2} = x_i + h\ /\ 2$. Формула (2) преобразуется к виду:

$$y_{i+1} = y_i + \frac{h}{6}\left[f(x_i, y_i) + 4f\left(x_{i+1/2}, y_{i+1/2}\right) + f(x_{i+1}, y_{i+1})\right].$$

Два последних слагаемых вычисляются следующим образом:

$$4f\left(x_{i+1/2}, y_{i+1/2}\right) + f(x_{i+1}, y_{i+1}) = 2f\left(x_{i+1/2}, y_i + \frac{\tilde{\Delta} y_i}{2}\right) + 2f\left(x_{i+1/2}, y_i + \frac{\tilde{\tilde{\Delta}} y_i}{2}\right) + f(x_{i+1}, y_i + \Delta y_i).$$

Если обозначить через:

$$k_i^1 = hf(x_i, y_i), \quad k_i^2 = hf(x_{i+1/2}, y_i + \frac{k_i^1}{2}),$$

$$k_i^3 = hf\left(x_{i+1/2}, y_i + \frac{k_i^2}{2}\right), \ \ k_i^3 = hf\left(x_{i+1}, y_i + k_i^3\right),$$

То схема вычислений будет следующая:

$$\left\{ y_{i+1} = y_i + \Delta y_i, \quad \Delta y_i = \frac{1}{6}\left(k_i^1 + 2k_i^2 + 2k_i^3 + k_i^4\right). \right.$$

## АЛГОРИТМ РЕШЕНИЯ

Задача состоит в том, чтобы найти значение неизвестной функции y в заданной точке x. Метод Рунге-Кутты находит приблизительное значение y для заданного x. Только обыкновенные дифференциальные уравнения первого порядка могут быть решены с использованием метода Рунге-Кутты 4-го порядка.

Ниже приведена формула, используемая для вычисления следующего значения $y_{n+1}$ из предыдущего значения $y_n$. Значение n равно 0, 1, 2, 3, ….$(x - x_0)$/h. Здесь h — высота ступени , а $x_{n+1} = x_0 + h$. Меньший размер шага означает большую точность.

$$K_1 = hf(x_n, y_n)$$
$$K_2 = hf(x_n + \frac{h}{2}, y_n + \frac{k_1}{2})$$
$$K_3 = hf(x_n + \frac{h}{2}, y_n + \frac{k_2}{2})$$
$$K_4 = hf(x_n + h, y_n + k_3)$$
$$y_{n+1} = y_n + k_1/6 + k_2/3 + k_3/3 + k_4/6 + O(h^5)$$

Формула в основном вычисляет следующее значение $y_{n+1}$, используя текущее значение $y_n$ плюс средневзвешенное значение четырех приращений.

- $k_1$— приращение, основанное на наклоне в начале интервала, с использованием $y$
- $k_2$ — приращение, основанное на наклоне в средней точке интервала с использованием $y + hk_{\frac{1}{2}}$
- $k_3$ снова является приращением, основанным на наклоне в средней точке, используя $y + hk_{\frac{2}{2}}$.
- $k_4$ представляет собой приращение, основанное на наклоне в конце интервала с использованием $y + hk_3$.

# ЛИСТИНГ ПРОГРАММЫ

Файл **main.cpp**

```
#include <bits/stdc++.h>
using namespace std;

float dydx(float x, float y) // y(x)'= x+y(x)
{
    return(x + y);
}

//Находит значение y для данного x, используя размер шага h и начальное
значение y0 в точке x0.
float rungeKutta(float x0, float y0, float x, float h)
{
    int n = (int)((x - x0) / h);// Подсчитываем количество итераций,
используя размер шага
                                // или высоту h
    float k1, k2, k3, k4;
    float y = y0;

    cout << "x" << "\t" << "y" << "\n";
    for (int i=1; i<=n; i++)
    {
        //Применяем формулу Рунге Кутты, чтобы найти последующие значения y
        k1 = h*dydx(x0, y);
        k2 = h*dydx(x0 + 0.5*h, y + 0.5*k1);
        k3 = h*dydx(x0 + 0.5*h, y + 0.5*k2);
        k4 = h*dydx(x0 + h, y + k3);
        y = y + (1.0/6.0)*(k1 + 2*k2 + 2*k3 + k4);;//Обновляем значение y
        x0 = x0 + h;// обновляем значение x
        cout << x0 <<"\t" << y << "\n";// Выводим значение X и Y;
    }

    return y;
}

int main()
{
    float x0 = 0, y = 1, x = 10, h = 0.2;//y(0)=1; x(0;10);
    cout << "Значение y в точке x равно:     \n" <<
            rungeKutta(x0, y, x, h) << "\n";

    return 0;
}
```

# РЕЗУЛЬТАТ РАБОТЫ ПРОГРАММЫ

Скриншот:

```
alexeynaumov@Lenovo-Legion-5-15ARH05H-267a6435:~/Vmath/course$ ./main
Значение y в точке x равно:
x       y
0.2     1.2428
0.4     1.58364
0.6     2.04421
0.8     2.65104
1       3.4365
1.2     4.44014
1.4     5.71027
1.6     7.30589
1.8     9.29905
2       11.7778
2.2     14.8496
2.4     18.6458
2.6     23.3267
2.8     29.0883
3       36.1697
3.2     44.8633
3.4     55.5259
3.6     68.5935
3.8     84.5985
4       104.191
4.2     128.166
4.4     157.494
4.6     193.358
4.8     237.208
5       290.81
5.2     356.323
5.4     436.386
5.6     534.219
5.8     653.756
6       799.803
6.2     978.229
6.4     1196.2
6.6     1462.48
6.8     1787.76
7       2185.09
7.2     2670.44
7.4     3263.3
7.6     3987.45
7.8     4871.97
8       5952.38
8.2     7272.03
8.4     8883.89
8.6     10852.7
8.8     13257.4
9       16194.5
9.2     19782
9.4     24163.8
9.6     29515.8
9.8     36052.7
10      44037
44037
```