

Министерство цифрового развития, связи и
массовых коммуникаций Российской Федерации

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

Кафедра прикладной математики и
кибернетики

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
к курсовой работе
по дисциплине «**Вычислительная
математика**»

Выполнил:
студент гр. ИС-242
«20» мая 2024 г.

/Журбенко В.Е./

Проверил:
преподаватель
«__» мая 2024 г.

/Лукинов В.Л./

Оценка «_____»

Новосибирск 2024 г.

ОГЛАВЛЕНИЕ

ПОСТАНОВКА ЗАДАЧИ	3
ОПИСАНИЕ ТЕОРИИ МЕТОДА РЕШЕНИЯ ЗАДАЧИ	4
СХЕМА АЛГОРИТМА РЕШЕНИЯ.....	6
ЛИСТИНГ ПРОГРАММЫ	8
РЕЗУЛЬТАТ РАБОТЫ ПРОГРАММЫ.....	9

ПОСТАНОВКА ЗАДАЧИ

Методом Рунге-Кутты 4 порядка решить задачу Коши для ОДУ:

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

$$y(0) = 1.$$

на отрезке $[0, 10]$ с точностью до 4 знака.

Не допускается использовать сторонние вычислительные библиотеки для решения задачи.

ОПИСАНИЕ ТЕОРИИ МЕТОДА РЕШЕНИЯ ЗАДАЧИ

Метод Рунге-Кутты — это один из численных методов, используемых для решения обыкновенных дифференциальных уравнений (ОДУ). Он позволяет находить приближённые решения без необходимости аналитически интегрировать уравнение.

Теория метода Рунге-Кутты четвёртого порядка

Метод Рунге-Кутты четвёртого порядка (RK4) является одним из наиболее используемых численных методов для решения задачи Коши для ОДУ. Его особенности включают:

1. Общая форма:

Рассматриваем ОДУ вида:

$$y' = f(x, y)$$

с начальным условием:

$$y(x_0) = y_0$$

где y' — производная функции y по переменной x .

2. Процесс вычисления:

Метод RK4 использует 4 промежуточных значения (коэффициента k) для расчёта следующего значения y :

- $k_1 = h \cdot f(x_n, y_n)$
- $k_2 = h \cdot f(x_n + \frac{h}{2}, y_n + \frac{k_1}{2})$
- $k_3 = h \cdot f(x_n + \frac{h}{2}, y_n + \frac{k_2}{2})$
- $k_4 = h \cdot f(x_n + h, y_n + k_3)$

Следующее значение y вычисляется по формуле:

$$y_{n+1} = y_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

3. Этапы метода:

- Начинаем с заданного начального условия.
- Устанавливаем шаг h (размер изменения x).
- Повторяем расчёты, пока не достигнем максимального значения x .

Решение задачи Коши

Для данной задачи Коши:

$$y' = x + y, y(0) = 1$$

Здесь:

- - Функция $f(x, y) = x + y$
- - Начальное условие $y_0 = 1$

- - Диапазон интегрирования - от $x_0 = 0$ до $x_{\max} = 10$

1. Установка параметров

- Начальное значение $x_0 = 0$
- Начальное значение $y_0 = 1$
- Шаг $h = 0.1$ (можно уменьшить для повышения точности)
- Максимальное значение $x_{\max} = 10$

2. Применение метода Рунге-Кутты

Теперь, используя метод **RK4**, мы последовательно вычисляем значения y на каждом шаге до достижения x_{\max} :

- Для $x = 0$:

$$k_1 = 0.1 \cdot f(0, 1) = 0.1 \cdot (0 + 1) = 0.1$$

$$k_2 = 0.1 \cdot f(0.05, 1.05) = 0.1 \cdot (0.05 + 1.05) = 0.11$$

$$k_3 = 0.1 \cdot f(0.05, 1.055) = 0.1 \cdot (0.05 + 1.055) = 0.1105$$

$$k_4 = 0.1 \cdot f(0.1, 1.1105) = 0.1 \cdot (0.1 + 1.1105) = 0.11105$$

Обновим y :

$$y_1 = 1 + \frac{1}{6}(0.1 + 2 \cdot 0.11 + 2 \cdot 0.1105 + 0.11105) \approx 1.10517$$

3. Повторение процесса

Повторяем этот процесс на каждом шаге до $x = 10$. Весь процесс программируется, и результаты выводятся с точностью до 4 знаков.

СХЕМА АЛГОРИТМА РЕШЕНИЯ

Схема алгоритма решения:

1. Инициализация параметров:

- Задать начальные условия:
 $x_0 = 0.0$
 $y_0 = 1.0$
- Задать шаг интегрирования:
 $h = 0.1$
- Задать максимальное значение:
 $x_{\max} = 10.0$

2. Определение функции:

Определить функцию $f(x, y)$:

$$f(x, y) = x + y$$

3. Вывод заголовка таблицы:

Вывести заголовок для таблицы значений x и y .

4. Основной цикл:

- Пока $x < x_{\max}$ делать:
 - **Вычисление k :**
 - ◆ Вычислить k_1 :
 $k_1 = h \cdot f(x_n, y_n)$
 - ◆ Вычислить k_2 :
 $k_2 = h \cdot f(x_n + \frac{h}{2}, y_n + \frac{k_1}{2})$
 - ◆ Вычислить k_3 :
 $k_3 = h \cdot f(x_n + \frac{h}{2}, y_n + \frac{k_2}{2})$
 - ◆ Вычислить k_4 :
 $k_4 = h \cdot f(x_n + h, y_n + k_3)$
 - **Обновление значения y :**
 - ◆ Обновить y :
 $y = y + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$

- **Обновление значения x :**

Обновить x : $x = x + h$

- **Вывод текущих значений x и y в таблице с точностью до 4 знака.**

5. Завершение программы:

Возврат значения 0 из функции ``main`` для завершения программы.

ЛИСТИНГ ПРОГРАММЫ С КОММЕНТАРИЯМИ

```
1  #include <stdio.h>
2
3  double f(double x, double y) {
4      return x + y;
5  }
6
7  void runge_kutta(double x0, double y0, double h, double x_max) {
8      double x = x0;
9      double y = y0;
10
11     printf("x\t y\n");
12     printf("%0.4f\t %0.4f\n", x, y);
13
14     while (x < x_max) {
15         double k1 = h * f(x, y);
16         double k2 = h * f(x + h / 2, y + k1 / 2);
17         double k3 = h * f(x + h / 2, y + k2 / 2);
18         double k4 = h * f(x + h, y + k3);
19
20         y = y + (k1 + 2*k2 + 2*k3 + k4) / 6;
21         x = x + h;
22
23         printf("%0.4f\t %0.4f\n", x, y);
24     }
25 }
26
27 int main() {
28     double x0 = 0.0;
29     double y0 = 1.0;
30     double h = 0.1;
31     double x_max = 10.0;
32
33     runge_kutta(x0, y0, h, x_max);
34
35     return 0;
36 }
```


РЕЗУЛЬТАТ РАБОТЫ ПРОГРАММЫ

x	y		
0.0000	1.0000	5.1000	321.9425
0.1000	1.1103	5.2000	356.3430
0.2000	1.2428	5.3000	394.3720
0.3000	1.3997	5.4000	436.4110
0.4000	1.5836	5.5000	482.8818
0.5000	1.7974	5.6000	534.2505
0.6000	2.0442	5.7000	591.0322
0.7000	2.3275	5.8000	653.7962
0.8000	2.6511	5.9000	723.1716
0.9000	3.0192	6.0000	799.8539
1.0000	3.4366	6.1000	884.6114
1.1000	3.9083	6.2000	978.2934
1.2000	4.4402	6.3000	1081.8386
1.3000	5.0386	6.4000	1196.2842
1.4000	5.7104	6.5000	1322.7766
1.5000	6.4634	6.6000	1462.5829
1.6000	7.3061	6.7000	1617.1033
1.7000	8.2479	6.8000	1787.8852
1.8000	9.2993	6.9000	1976.6389
1.9000	10.4718	7.0000	2185.2545
2.0000	11.7781	7.1000	2415.8210
2.1000	13.2323	7.2000	2670.6467
2.2000	14.8500	7.3000	2952.2833
2.3000	16.6483	7.4000	3263.5503
2.4000	18.6463	7.5000	3607.5640
2.5000	20.8649	7.6000	3987.7685
2.6000	23.3274	7.7000	4407.9699
2.7000	26.0594	7.8000	4872.3748
2.8000	29.0892	7.9000	5385.6320
2.9000	32.4482	8.0000	5952.8794
3.0000	36.1710	8.1000	6579.7952
3.1000	40.2958	8.2000	7272.6548
3.2000	44.8649	8.3000	8038.3936
3.3000	49.9251	8.4000	8884.6762
3.4000	55.5280	8.5000	9819.9736
3.5000	61.7307	8.6000	10853.6475
3.6000	68.5963	8.7000	11996.0443
3.7000	76.1944	8.8000	13258.5985
3.8000	84.6021	8.9000	14653.9470
3.9000	93.9046	9.0000	16196.0560
4.0000	104.1960	9.1000	17900.3604
4.1000	115.5802	9.2000	19783.9185
4.2000	128.1722	9.3000	21865.5824
4.3000	142.0991	9.4000	24166.1872
4.4000	157.5012	9.5000	26708.7590
4.5000	174.5336	9.6000	29518.7458
4.6000	193.3679	9.7000	32624.2717
4.7000	214.1936	9.8000	36056.4188
4.8000	237.2199	9.9000	39849.5383
4.9000	262.6786	10.0000	44041.5938
5.0000	290.8252	10.1000	48674.5418

Результат выполнения кода изображён в виде

скриншота из-за размера полученного вывода