

УДК 004.42

РЕАЛИЗАЦИЯ МЕТОДОВ РУНГЕ-КУТТЫ ДЛЯ РЕШЕНИЯ ОБЫКНОВЕННЫХ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ ПЕРВОГО ПОРЯДКА НА ЯЗЫКЕ ПРОГРАММИРОВАНИЯ C++

Яхова Ю.Д.¹

¹Тверской государственный университет (ТвГУ), Россия, г. Тверь, e-mail: juliayahova@mail.ru

Аннотация

В данной работе рассматривается реализация численного решения задачи Коши для обыкновенного дифференциального уравнения первого порядка двухэтапным, трёхэтапным и четырёхэтапным разностными методами Рунге-Кутты второго, третьего и четвёртого порядка аппроксимации соответственно на языке программирования C++. Данная программа может быть полезна в современном мире в силу компьютеризации в различных областях техники, науки производства. Для решения многих задач, встающих перед человеком в различных сферах деятельности, приходится составлять математические модели реальных ситуаций. Очень часто эти математические модели сводятся к решению дифференциальных уравнений, которые достаточно решить численно. В программе предусмотрен выбор одного из перечисленных методов Рунге-Кутты по желанию пользователя. Кроме того, частью кода программы является код для создания окна с визуализацией следа точного решения и разностного решения данной задачи Коши. Поэтому данная работа будет также полезна и студентам, занимающимся методами вычислений для лучшего понимания происходящего: что такое разностное решение задачи Коши, а что такое след точного решения задачи Коши; какой из перечисленных методов Рунге-Кутты является более точным.

Ключевые слова: Дифференциальное уравнение, численное решение, методы Рунге-Кутты, методы Рунге-Кутта, методы вычислений, C++

IMPLEMENTATION OF RUNGE-KUTTA METHODS FOR SOLVING FIRST-ORDER ORDINARY DIFFERENTIAL EQUATIONS IN THE C++ PROGRAMMING LANGUAGE

Yakhova Yu.D.¹

¹ Tver State University (TvSU), Tver, Russia, e-mail: juliayahova@mail.ru

Annotation

In this paper, we consider the implementation of the numerical solution of the Cauchy problem for an ordinary first-order differential equation by two-stage, three-stage and four-stage Runge-Kutta difference methods of the second, third and fourth order approximation, respectively, in the C++ programming language. This program can be useful in the modern world due to computerization in various fields of technology, production science. To solve many problems faced by a person in various fields of activity, it is necessary to make mathematical models of real situations. Very often, these mathematical models are reduced to solving differential equations, which are sufficient to solve numerically. The program provides a choice of one of the listed Runge-Kutta methods at the user's request. In addition, part of the program code is the code for creating a window with visualization of the trace of the exact solution and the difference solution of this Cauchy problem. Therefore, this work will also be useful for students involved in computational methods to better understand what is happening: what is the difference solution of the Cauchy problem, and what is the trace of the exact solution of the Cauchy problem; which of the listed Runge-Kutta methods is more accurate.

Keywords: Customer's loyalty, NPS, market research, market review

Введение

Применение дифференциальных уравнений в математических моделях реальных процессов позволяет решать задачи в различных областях техники, науки, производства. Например, в ядерной физике – при изучении процессов радиоактивного распада веществ. Поэтому в силу компьютеризации очень полезно иметь программу, позволяющую численно решить ОДУ.

Цель исследования

Целью исследования является решение задачи Коши (1)

$$\begin{cases} \frac{du}{dt} = f(t, u), t \in [a; b], \\ u(0) = u_0 \end{cases} \quad (1)$$

численными методами – методами Рунге-Кутты второго, третьего и четвёртого порядков аппроксимации.

Материал и методы исследования

Рассмотрим равномерную разностную сетку отрезка $[a; b]$:

$T = \{t_n = a + \tau n | n = \overline{0, N}\}$, где N – заданное количество равных частей, на которые разбит отрезок $[a; b]$, $\tau = \frac{b-a}{N}$ – шаг разбиения.

Пусть $y_0 = u_0$, $y_n = y(t_n)$. Тогда будут справедливы явные разностные методы Рунге-Кутты [1,2]:

явный двухэтапный метод Рунге-Кутты второго порядка аппроксимации:

$$\begin{cases} K_1 = f(t_n, y_n), \\ K_2 = f\left(t_n + \frac{\tau}{2}, y_n + \frac{\tau}{2}K_1\right), \\ y_{n+1} = \tau K_2 + y_n; \end{cases}$$

явный трёхэтапный метод Рунге-Кутты третьего порядка аппроксимации:

$$\begin{cases} K_1 = f(t_n, y_n), \\ K_2 = f\left(t_n + \frac{\tau}{2}, y_n + \frac{\tau}{2}K_1\right), \\ K_3 = f\left(t_n + \tau, y_n - \tau K_1 + 2\tau K_2\right), \\ y_{n+1} = \frac{\tau}{6}(K_1 + 4K_2 + K_3) + y_n; \end{cases}$$

явный четырёхэтапный метод Рунге-Кутты четвёртого порядка аппроксимации:

$$\begin{cases} K_1 = f(t_n, y_n), \\ K_2 = f\left(t_n + \frac{\tau}{2}, y_n + \frac{\tau}{2}K_1\right), \\ K_3 = f\left(t_n + \frac{\tau}{2}, y_n + \frac{\tau}{2}K_2\right), \\ K_4 = f\left(t_n + \tau, y_n + \tau K_3\right), \\ y_{n+1} = \frac{\tau}{6}(K_1 + 2K_2 + 2K_3 + K_4) + y_n. \end{cases}$$

Будет продемонстрирована реализация на языке программирования C++ сформулированных выше методов Рунге-Кутты для решения задачи Коши (1) в случае, когда $f(t, u) = u + t, a = 0, b = 1, N = 10$, а также будет рассмотрена визуализация следа точного решения $u(t) = 2e^t - t - 1$ и визуализация разностного решение задачи Коши (1) в одном окне. Однако эти данные можно несложно быстро поменять и решить задачу Коши (1) с другими данными $f(t, u), a, b, N$.

Результаты исследования и их обсуждение

Программный код для решения задачи Коши

$$\begin{cases} \frac{du}{dt} = u + t, t \in [0; 1], \\ u(0) = 1, \end{cases}$$

если мелкость разбиения $N = 0,1$:

```
#include <SFML/Graphics.hpp>
#include <cmath>
#include <iostream>
using namespace sf;
using namespace std;
//ширина окна для визуализации решения
const int W = 752;
//высота окна для визуализации решения
const int H = 802;
//следующие данные можно изменить, если необходимо решить другую задачу Коши (1)
const int N = 10;
const double a = 0;
const double b = 1;
const double y_0 = 1; //начальное условие u(0)=1
//объявление функции f(t,u)
double f(double t, double u) {
    return t + u;
}
//объявление функции - точного решения задачи Коши (1)
double ExactSolution(double x) {
    return 2 * exp(x) - x - 1;
}
int main() {
    setlocale(LC_ALL, "Russian");
```

//tau - мелкость разбиения, T - массив значений равномерной разностной сетки, Y - массив значений разностного решения (1)

```
double tau = (b - a) / N;
```

```
double T[N + 1] = {};
```

```
for (int i = 1; i <= N; i++) T[i] = a + i * tau;
```

```
double Y[N + 1] = {};
```

```
Y[0] = y_0;
```

```
//z - условие проверки корректности введённых данных пользователем программы
```

```
double z = 0;
```

```
while (z == 0) {
```

```
    cout << "Метод Рунге-Кутты какого порядка аппроксимации Вы хотите  
использовать для приближённого решения задачи Коши?" << endl << "Нажмите:" << endl <<  
"2, если второго порядка аппроксимации;" << endl << "3, если третьего порядка  
аппроксимации;" << endl << "4, если четвёртого порядка аппроксимации" << endl;
```

```
    int p;
```

```
    cin >> p;
```

```
    if (p == 2)
```

```
        for (int i = 0; i < N; i++) {
```

```
            double k1 = f(T[i], Y[i]);
```

```
            double k2 = f(T[i] + tau / 2, Y[i] + tau / 2 * k1);
```

```
            Y[i + 1] = Y[i] + tau * k2;
```

```
            z = 1;
```

```
        }
```

```
    else if (p == 3)
```

```
        for (int i = 0; i < N; i++) {
```

```
            double k1 = f(T[i], Y[i]);
```

```
            double k2 = f(T[i] + tau / 2, Y[i] + tau / 2 * k1);
```

```
            double k3 = f(T[i] + tau, Y[i] - tau * k1 + 2 * tau * k2);
```

```
            Y[i + 1] = Y[i] + tau / 6 * (k1 + 4 * k2 + k3);
```

```
            z = 1;
```

```
        }
```

```
    else if (p == 4)
```

```
        for (int i = 0; i < N; i++) {
```

```
            double k1 = f(T[i], Y[i]);
```

```
            double k2 = f(T[i] + tau / 2, Y[i] + tau / 2 * k1);
```

```

        double k3 = f(T[i] + tau / 2, Y[i] + tau / 2 * k2);
        double k4 = f(T[i] + tau, Y[i] + tau * k3);
        Y[i + 1] = Y[i] + tau / 6 * (k1 + 2 * k2 + 2 * k3 + k4);
        z = 1;
    }
    else {
        cout << "Такого варианта нет, повторите попытку" << endl;
    }
}
//вывод разностного решения и следа точного решения (1)
cout << "t    y_n (разностное решение)    u_n (след точного решения)" << endl;
for (int i = 0; i <= N; i++) cout << T[i] << "    " << Y[i] << "    " <<
ExactSolution(a+tau*i)<< endl;
//следующий фрагмент программы программы написан для визуализации разностного
решения и следа точного решения (1), большая часть кода заимствована в [3]
RenderWindow window(VideoMode(W, H), "Exact (black line) and inexact (blue line)
solutions of a differential equation");
//центр координатной плоскости
int x0 = W / 2;
int y0 = H / 2;
//ось ox
RectangleShape OsX(Vector2f(W, 1));
OsX.setFill(Color::Black);
OsX.setPosition(0, y0);
//ось oy
RectangleShape OsY(Vector2f(1, H));
OsY.setFill(Color::Black);
OsY.setPosition(x0, 0);
//стрелки на осях
RectangleShape strel[4];
for (int i = 0; i < 4; i++) {
    strel[i].setSize(Vector2f(1, 25));
    strel[i].setFillColor(Color::Black);

    if (i < 2)

```

```

        strel[i].setPosition(x0, 0);
    else
        strel[i].setPosition(W, y0);
}
strel[0].setRotation(25);
strel[1].setRotation(-25);
strel[2].setRotation(60);
strel[3].setRotation(-250);
//выполнение действий, когда открыто окно
while (window.isOpen())
{
    Event event;
    //закрытие окна - нажать на крестик
    while (window.pollEvent(event))
    {
        if (event.type == Event::Closed)
            window.close();
    }
    //вид окна: белый экран
    window.clear(Color::White);
    //изображение в окне стрелок
    window.draw(OsX);
    window.draw(OsY);
    for (int i = 0; i < 4; i++)
        window.draw(strel[i]);
    //масштабирование графика по оси ox
    int scalex = 100;
    //масштабирование графика по оси oy
    int scaley = 100;
    //визуализация разностного решения задачи Коши (1)
    //синие точки радиуса 2 для визуализации разностного решения (1)
    CircleShape point1(2.f);
    point1.setFill(Color::Blue);
    for (int i = 0; i <= N; i++) {
        float x = T[i];

```

```

float y = Y[i]; // наш график
//меняем ориентацию системы координат, т.к. по опциям окна она не
является "стандартной"
float x1 = x0 + x * scalex;
float y1 = y0 - y * scaley;
//выводим на экран разностное решение задачи Коши(1)
point1.setPosition(x1, y1);
window.draw(point1);
}
//визуализация следа точного решения задачи Коши (1)
//чёрные точки радиуса 1 для визуализации следа точного решения (1)
CircleShape point2(1.f);
point2.setFill(Color::Black);
float c = 1000;
int mass = (b - a) * c + 1;
for (int i = 0; i < mass; i++) {
    float x = a + i / c;
    float y = ExactSolution(x);
    float x1 = x0 + x * scalex;
    float y1 = y0 - y * scaley;
    point2.setPosition(x1, y1);
    window.draw(point2);
}
window.display();
}
return 0;
}

```

В результате выполнения программы, если пользователь сначала ошибётся и на вопрос о выборе метода Рунге-Кутты нажмёт 0, а затем исправится и выберет третий порядок аппроксимации будет выведено два окна (рис. 1 и рис. 2).

```

C:\Users\Admin\source\repos\Project1\x64\Debug\Project1.exe
Метод Рунге-Кутты какого порядка аппроксимации Вы хотите использовать для приближённого решения задачи Коши?
Нажмите:
2, если второго порядка аппроксимации;
3, если третьего порядка аппроксимации;
4, если четвёртого порядка аппроксимации
0
Такого варианта нет, повторите попытку
Метод Рунге-Кутты какого порядка аппроксимации Вы хотите использовать для приближённого решения задачи Коши?
Нажмите:
2, если второго порядка аппроксимации;
3, если третьего порядка аппроксимации;
4, если четвёртого порядка аппроксимации
3
t      u_p (разностное решение)      u_n (след точного решения)
0      1      1
0.1    1.11033      1.11034
0.2    1.24279      1.24281
0.3    1.39969      1.39972
0.4    1.5836      1.58365
0.5    1.79738      1.79744
0.6    2.04415      2.04424
0.7    2.3274      2.32751
0.8    2.65094      2.65108
0.9    3.01904      3.01921
1      3.43635      3.43656

```

Рис. 1.

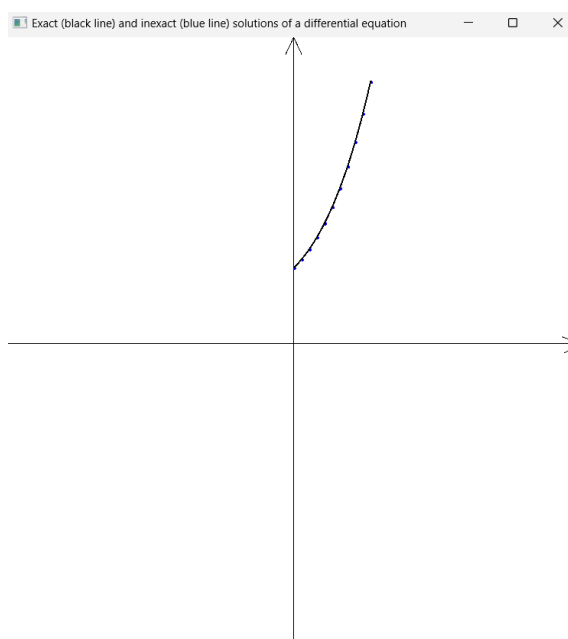


Рис. 2.

Заключение

Таким образом, в работе продемонстрирована программа, реализованная на языке программирования C++ для решения задачи Коши (1). Данный результат можно использовать для сравнения точности методов Рунге-Кутты второго, третьего и четвертого порядков аппроксимации опытным путём, а также применять в различных областях техники, науки, производства.

Список литературы

1. Амосов А. А., Дубинский Ю. А., Копченова Н. В. Вычислительные методы для инженеров: Учеб. пособие. – М.: Высш. шк., 1994. – 544 с.: ил. [Электронный ресурс]. URL: <https://djuv.online/file/PPFsrza9PEZHq?ysclid=lr9z29dvp6314913486> (дата обращения: 12.01.2024).

2. Мышенков В.И., Мышенков Е.В. Численные методы. Ч. 2. Численное решение обыкновенных дифференциальных уравнений: Учебное пособие для студентов специальности 073000. – М.:МГУЛ, 2005. – 109 с.: ил. [Электронный ресурс]. URL: https://mf.bmstu.ru/UserFiles/File/KF/k6/books/math/uchebniki/Mishenkov_2.pdf (дата обращения: 12.01.2024).
3. Код для визуализации графика функции. [Электронный ресурс]. URL: <https://gist.github.com/Andrey-byte/c0a7189f2d422d00cdf762ed434d8fd7> (дата обращения: 12.01.2024).
4. Зенков, А.В. Численные методы: учеб. пособие. — Екатеринбург: Изд-во Урал. ун-та, 2016.— 124 с. [Электронный ресурс]. URL: https://elar.urfu.ru/bitstream/10995/40678/1/978-5-7996-1781-3_2016.pdf?ysclid=lr9zeq8vkn645122653 (дата обращения: 12.01.2024).
5. Т. А. Павловская. С/С++. Программирование на языке высокого уровня. — СПб.: Питер, 2003. —461 с: ил. [Электронный ресурс]. URL: <https://studylib.ru/doc/2356765/t.a.-pavlovskaya?ysclid=lr9z9eu9rx64307149> (дата обращения: 12.01.2024).