Министерство цифрового развития, связи и массовых коммуникаций Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Сибирский государственный университет телекоммуникаций и информатики»

СибГУТИ

09.03.01 "Информатика и вычислительная техника"

профиль "Программное обеспечение средств вычислительной техники и автоматизированных систем"

ОТЧЁТ

по курсовой работе по дисциплине

«ВЫЧИСЛИТЕЛЬНАЯ МАТЕМАТИКА»

Выполнил:

студент гр. ИП-114 \_\_\_\_\_\_\_\_\_\_\_\_\_/Яворский Д. И./

«27» мая 2023 г.

Преподаватель

ассистент каф. ПМиК \_\_\_\_\_\_\_\_\_\_\_\_\_/Рубан А. А./

«27» мая 2023 г. Оценка\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Новосибирск, 2023г.

**Оглавление:**

[1. **Задание на курсовую работу** 2](#_Toc136456965)

[2. **Теория** 2](#_Toc136456966)

[3. **Листинг** 4](#_Toc136456967)

# **Задание на курсовую работу**

Реализовать программу для решения дифференциальных уравнений, заданных в неявном виде (старшая производная выражена неявным образом), с краевыми условиями (решение краевой задачи).

# **Теория**

1. ДУ задано в неявном виде:

Так как ДУ не может быть решено относительно старшей производной, то тогда на каждом шаге решаем нелинейное уравнение относительно 𝑦( ) (все остальные неизвестные 𝑦, 𝑦 , 𝑦 , … , 𝑦( ) к этому моменту уже известны). Решать уравнение относительно старшей производной любым методом (Хорд, МПД, Ньютона).

1. Краевая задача:

Все известные методы решения ДУ и СДУ используют начальные условия, заданные в одной точке, так называемую задачу Коши, но для ДУ высших порядков часто бывает необходимо решить не задачу Коши, а так называемую краевую задачу, т.е. начальные условия, которые заданы в разных точках.

Рассмотрим простейшую краевую задачу для ДУ 2го порядка:

𝑦 = 𝑓(𝑥, 𝑦, 𝑦 )

𝑦(𝑎) = 𝑦

𝑦(𝑏) = 𝑦

для решения этой задачи мы будем подбирать 𝑦′(𝑎) с тем, чтобы 𝑦(𝑏) = 𝑦

1. Метод стрельб:

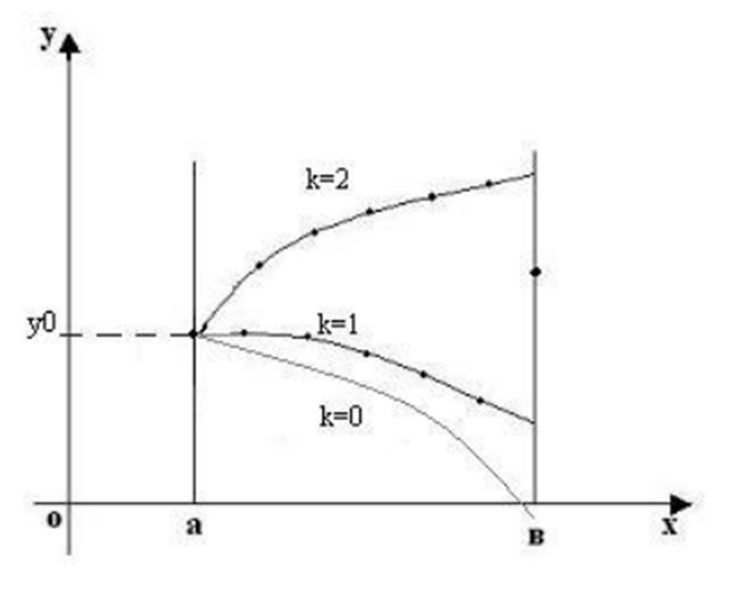
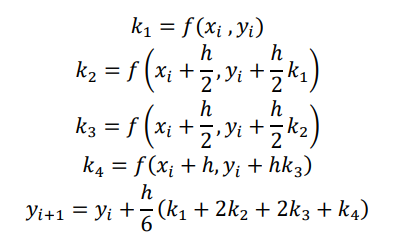
После пристрелки и определения интервала [a, b], где идёт смена знака, запускаем МПД или МХ. На практике это выглядит так, как будто мы решаем уравнение 𝑞(𝑘) = 𝑦 , где 𝑞(𝑘) возвращает решение задачи Коши в точке b при заданном k.

Рис. 1. Метод стрельб

1. Метод Рунге-Кутта 4го порядка:

Наиболее применяемым методом решения ДУ и СДУ является метод РунгеКутта 4го порядка.

Формулы метода Рунге-Кутта 4го порядка:



В векторной форме данной формулы, величины y, f, k заменяют на Y, F, K.

# **Листинг**

#include <iostream>

#include <vector>

#include <math.h>

using namespace std;

double funk(double x, double y, double yi) {

return (exp(x) + yi);

}

double u\_cal(double u, int n) {

double temp = u;

for (int i = 1; i < n; i++) temp \*= (u + i);//q\*(q+1)\*...\*(q+m-1)

return temp;

}

double u\_cal2(double u, int n) { //1 formula

double temp = u;

for (int i = 1; i < n; i++) temp \*= (u - i);//q\*(q-1)\*...\*(q-n+1)

return temp;

}

double fact(int n) {

double f = 1.;

for (int i = 2; i <= n; i++) f \*= i;//формула для факториала

return f;

}

double inter(int n, double x, vector<pair<double, pair<double, double>>> XnY) {

double\*\* y = new double\* [n];

for (int i = 0; i < n; i++) {

y[i] = new double[n];

}

for (int i = 0; i < n; i++) y[i][0] = XnY.at(i).second.second;

for (int i = 1; i < n; i++) {

for (int j = 0; j < n - i; j++) y[j][i] = y[j + 1][i - 1] - y[j][i - 1];

}

double sum = y[0][0];

double u = (x - XnY.at(0).first) / (XnY.at(1).first - XnY.at(0).first);//вычисление q

for (int i = 1; i < n; i++) {

sum += (u\_cal2(u, i) \* y[0][i]) / fact(i); //основная формула

}

return sum;

}

vector<pair<double, pair<double, double>>> Runge\_Kutt\_4(double a, double b, double y0, double yb, double e, double \*h1) {

vector<pair<double, pair<double, double>>> XnY;

XnY.push\_back(make\_pair(a, make\_pair(y0, yb)));

int index = 0;

double h = \*h1;

double y;

double yl;

double yI;

while (powf(h, 4) > e) {

h /= 2;

}

\*h1 = h;

double\*\* K = new double\* [4];

for (int i = 0; i < 4; i++) {

K[i] = new double[2];

}

for (auto i = a + h; i < b ; i += h) {

K[0][0] = XnY.at(index).second.second;

K[0][1] = funk(XnY.at(index).first, XnY.at(index).second.first, XnY.at(index).second.second);

y = XnY.at(index).second.first + K[0][0] \* h / 2;

yI = XnY.at(index).second.second + K[0][1] \* h / 2;

K[1][0] = yI;

K[1][1] = funk(XnY.at(index).first + h / 2, y, yI);

y = XnY.at(index).second.first + K[1][0] \* h / 2;

yI = XnY.at(index).second.second + K[1][1] \* h / 2;

K[2][0] = yI;

K[2][1] = funk(XnY.at(index).first + h / 2, y, yI);

y = XnY.at(index).second.first + K[2][0] \* h;

yI = XnY.at(index).second.second + K[2][1] \* h;

K[3][0] = yI;

K[3][1] = funk(XnY.at(index).first + h, y, yI);

y = XnY.at(index).second.first + h \* (K[0][0] + 2 \* K[1][0] + 2 \* K[2][0] + K[3][0]) / 6;

yI = XnY.at(index).second.second + h \* (K[0][1] + 2 \* K[1][1] + 2 \* K[2][1] + K[3][1]) / 6;

XnY.push\_back(make\_pair(i, make\_pair(y, yI)));

index++;

}

while (abs(XnY.back().second.first - yb) > e) {

yl = XnY.at(0).second.second;

yl -= (XnY.back().second.first - yb) / 2;

XnY.clear();

XnY.push\_back(make\_pair(a, make\_pair(y0, yl)));

index = 0;

for (auto i = a + h; i < b ; i += h) {

K[0][0] = XnY.at(index).second.second;

K[0][1] = funk(XnY.at(index).first, XnY.at(index).second.first, XnY.at(index).second.second);

y = XnY.at(index).second.first + K[0][0] \* h / 2;

yI = XnY.at(index).second.second + K[0][1] \* h / 2;

K[1][0] = yI;

K[1][1] = funk(XnY.at(index).first + h / 2, y, yI);

y = XnY.at(index).second.first + K[1][0] \* h / 2;

yI = XnY.at(index).second.second + K[1][1] \* h / 2;

K[2][0] = yI;

K[2][1] = funk(XnY.at(index).first + h / 2, y, yI);

y = XnY.at(index).second.first + K[2][0] \* h;

yI = XnY.at(index).second.second + K[2][1] \* h;

K[3][0] = yI;

K[3][1] = funk(XnY.at(index).first + h, y, yI);

y = XnY.at(index).second.first + h \* (K[0][0] + 2 \* K[1][0] + 2 \* K[2][0] + K[3][0]) / 6;

yI = XnY.at(index).second.second + h \* (K[0][1] + 2 \* K[1][1] + 2 \* K[2][1] + K[3][1]) / 6;

XnY.push\_back(make\_pair(i, make\_pair(y, yI)));

index++;

}

}

cout << endl;

cout << " | " << XnY.back().second.first<<"|"<<endl;

return XnY;

}

int main()

{

vector<pair<double, pair<double, double>>> answerF;

double x0 = 0;

double xN = 1;

double y0 = 0;

double h = 0.1;

double e = 0.00000001;

answerF = Runge\_Kutt\_4(x0, xN, y0, 2.7181828, e, &h);

cout.precision(9);

cout << " | Rungey-Kutt 4 poryadka |" << endl;

for (int i = 0; i < (int)answerF.size(); i++) {

cout << "x: " << answerF.at(i).first << " | y: " << answerF.at(i).second.first << " | y': " << answerF.at(i).second.second << endl;

}

cout << endl;

cout << "step: " << h;

//cout << "| result of interpolation y':" << inter(answerF.size(), 0.5, answerF);

answerF.clear();

return 0;

}