**Федеральное государственное бюджетное образовательное учреждение высшего образования**

**"Уфимский государственный авиационный технический университет"**

**Кафедра** Высокопроизводительных вычислительных технологий и систем

**Дисциплина:** Теория разностных схем

**Отчет по лабораторной работе №1**

**Тема:** «**РЕШЕНИЕ НАЧАЛЬНЫХ И КРАЕВЫХ ЗАДАЧ**

**ДЛЯ ОБЫКНОВЕННЫХ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ**»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Группа МКН-317 | Фамилия И.О. | Подпись | Дата | Оценка |
| Студент | Хромых К.М.  Чучкалол Д,А. |  |  |  |
| Принял | Гайнетдинова А.А |  |  |  |

**Уфа 2023**

**Цель:** получить навык численного решения задач для обыкновенных дифференциальных уравнений с использованием различных методов на примере задачи Коши для системы обыкновенных дифференциальных уравнений первого порядка и начально-краевой задачи для обыкновенного дифференциального уравнения второго порядка.

**Задание 1.**

Рассматривается задача Коши для системы уравнений движения материальной точки в потенциальном поле :

|  |  |
| --- | --- |
|  | (1) |

1. Написать вычислительную программу на языке программирования C++ решения задачи (1) методом Эйлера с постоянным шагом.
2. Исследовать зависимость решения при больших временах от величины шага временной сетки. Построить графики решений для различных значений шага.
3. Выполнить сравнение полученных решений с численным решением в каком-либо математическом пакете, полученным с помощью метода высокого порядка точности (например, Рунге-Кутта 4–5). Построить графики разности решений.

Разностная схема метода Эйлера:

Ниже представлены графики решения с соответствующими шагами, промежуток был выбран . Изображение выглядит как текст, диаграмма, График, Шрифт

Автоматически созданное описание

Рисунок 1 графики решения с соответствующими шагами

Для сравнения расчётов метода Эйлера с методом более высокого порядка был выбран метод РК4 с математического пакета scipy.

Изображение выглядит как График, диаграмма, линия, текст

Автоматически созданное описание

Рисунок 2 разность решении РК4 и Эйлера

**Задание 2.**

1. Написать вычислительную программу на языке программирования C++ решения задачи (1) по явной двухшаговой схеме Адамса с постоянным шагом.
2. Исследовать зависимость решения при больших временах от величины шага временной сетки. Построить графики решений для различных значений шага.
3. Выполнить сравнение полученных решений с решением по методу Эйлера (задача 1) и численным решением в каком-либо математическом пакете, полученным с помощью метода высокого порядка точности (например, Рунге-Кутта 4–5). Построить графики разности решений.

Разностная схема метода Адамса:

Так как метод двухшаговый нужно найти и . Для их поиска был выполнен метод Эйлера.

Изображение выглядит как текст, диаграмма, График, линия

Автоматически созданное описание

Рисунок 3 график решении

Изображение выглядит как График, диаграмма, линия, текст

Автоматически созданное описание

Рисунок 4 разность решении между РК4(scipy) и метода Адамса

Изображение выглядит как График, диаграмма, текст, линия

Автоматически созданное описание

Рисунок 5 график разности решении между методами Эйлера и Адамса

**Задание 3.**

1. Написать вычислительную программу на языке программирования C++ решения задачи (1) методом Рунге-Кутта 4-го порядка.
2. Исследовать зависимость решения при больших временах от величины шага временной сетки. Построить графики решений для различных значений шага.
3. Выполнить сравнение полученных решений с численным решением в каком-либо математическом пакете, полученным с помощью метода высокого порядка точности (например, Рунге-Кутта 4–5). Построить графики разности решений.

Разностная схема:

Изображение выглядит как текст, диаграмма, График, линия

Автоматически созданное описание

Рисунок 6 график решении метода РК4

Изображение выглядит как График, диаграмма, линия, текст

Автоматически созданное описание

Рисунок 7 график разности решении между методами РК4 и РК4(scipy)

**Задание 4.**

1. Написать вычислительную программу на языке программирования C++ решения задачи (2) конечно-разностным методом с решением получающейся СЛАУ методом прогонки.
2. Исследовать зависимость решения от величины шага сетки. Построить графики решений для различных значений шага.
3. Выполнить сравнение полученных решений с численным решением в каком-либо математическом пакете. Построить графики разности решений.

|  |  |
| --- | --- |
|  | (2) |

Изображение выглядит как текст, диаграмма, График, линия

Автоматически созданное описание

Рисунок 8 график решения с разными шагами

Изображение выглядит как линия, График, диаграмма

Автоматически созданное описание

Рисунок 9 разность решения с scipy

**Листинг**

#include <iostream>

#include <vector>

#include<string>

#include <fstream>

using namespace std;

double U(double x) {

return -(4 / 9.1) \* (x \* x \* sinh(x) + 2 \* x \* cosh(x));

}

double func(double x) {

return 6 \* x \* x + 1;

};

vector<double> TripleDiag(vector<double> A, vector<double> B, vector<double> C, vector<double> D) {

vector<double> P(A.size());

vector<double> Q(A.size());

vector<double> X(A.size());

int N = A.size();

P[0] = C[0] / B[0];

Q[0] = D[0] / B[0];

for (int i = 1; i < N; ++i)

{

if (i < N - 1)

P[i] = C[i] / (B[i] - A[i] \* P[i - 1]);

Q[i] = (D[i] - A[i] \* Q[i - 1]) / (B[i] - A[i] \* P[i - 1]);

}

// backward

X[N - 1] = Q[N - 1];

for (int i = N - 2; i >= 0; --i)

{

X[i] = Q[i] - P[i] \* X[i + 1];

}

return X;

}

double f1(double x, double v, double u)

{

return 4 \* v + 6 \* x \* x + 1;

}

double f2(double x, double v, double u)

{

return v;

}

double rk4(double p)

{

int n = 1000;

double k1, k2, k3, k4;

double q1, q2, q3, q4;

double h = 1. / n;

double v\_0, u\_n, v\_n, u\_0;

u\_0 = v\_0 = 0;

u\_n = p;

v\_n = u\_n - 3.5625;

double x = 0;

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

{

u\_0 = u\_n;

v\_0 = v\_n;

x += i \* h;

k1 = f1(x, v\_0, u\_0);

q1 = f2(x, v\_0, u\_0);

k2 = f1(x + h / 3, v\_0 + h \* k1 / 3, u\_0 + h \* q1 / 3);

q2 = f2(x + h / 3, v\_0 + h \* k1 / 3, u\_0 + h \* q1 / 3);

k3 = f1(x + 2 \* h / 3, v\_0 - h \* k1 / 3, u\_0 - h \* q1 / 3 + h \* q2);

q3 = f2(x + 2 \* h / 3, v\_0 - h \* k1 / 3, u\_0 - h \* q1 / 3 + h \* q2);

k4 = f1(x + h, v\_0 + h \* k1 - h \* k2 + h \* k3, u\_0 + h \* q1 - h \* q2 + h \* q3);

q4 = f2(x + h, v\_0 + h \* k1 - h \* k2 + h \* k3, u\_0 + h \* q1 - h \* q2 + h \* q3);

v\_n += h \* (k1 + 3 \* k2 + 3 \* k3 + k4) / 8;

u\_n += h \* (q1 + 3 \* q2 + 3 \* q3 + q4) / 8;

}

return (u\_n + v\_n + 3.5625);

}

double find\_p()

{

double p1 = 0;

double p2 = -1.;

double p = (p2 - p1) / 2.;

while (rk4(p1) > 1e-10)

{

p = (p2 + p1) / 2.;

if (rk4(p1) \* rk4(p) < 0)

{

p2 = p;

}

else if (rk4(p) \* rk4(p2) < 0)

{

p1 = p;

}

}

return p1;

}

void z1\_diff\_eq() {

double v\_next, x\_next, v\_prev = 0.09, x\_prev = 0;

double h;

vector<double> n{ 10,100,1000,10000 };

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

ofstream file\_to\_cout\_V("z1\_v" + to\_string(i) + ".txt");

ofstream file\_to\_cout\_X("z1\_x" + to\_string(i) + ".txt");

v\_prev = 0.09, x\_prev = 0;

v\_next = 0, x\_next = 0;

h = 10. / (n[i] - 1);

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

x\_next = v\_prev \* h + x\_prev;

v\_next = U(x\_next) \* h + v\_prev;

file\_to\_cout\_V << v\_prev << endl;

file\_to\_cout\_X << x\_prev << endl;

v\_prev = v\_next;

x\_prev = x\_next;

}

file\_to\_cout\_V.close();

file\_to\_cout\_X.close();

}

}

void z2\_diff\_eq() {

double v\_next, x\_next, v\_prev\_prev = 0.09, x\_prev\_prev = 0, x\_prev, v\_prev;

double h;

vector<double> n{ 10,100,1000,10000 };

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

ofstream file\_to\_cout\_V("z2\_v" + to\_string(i) + ".txt");

ofstream file\_to\_cout\_X("z2\_x" + to\_string(i) + ".txt");

v\_prev\_prev = 0.09, x\_prev\_prev = 0;

h = 10. / (n[i] - 1);

x\_prev = v\_prev\_prev \* h + x\_prev\_prev;

v\_prev = U(x\_prev) \* h + v\_prev\_prev;

file\_to\_cout\_V << v\_prev\_prev << endl;

file\_to\_cout\_X << x\_prev\_prev << endl;

file\_to\_cout\_V << v\_prev << endl;

file\_to\_cout\_X << x\_prev << endl;

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

x\_next = (3 \* v\_prev - v\_prev\_prev) \* h / 2 + x\_prev;

v\_next = (3 \* U(x\_next) - U(x\_prev)) \* h / 2 + v\_prev;

file\_to\_cout\_V << v\_next << endl;

file\_to\_cout\_X << x\_next << endl;

v\_prev\_prev = v\_prev;

v\_prev = v\_next;

x\_prev = x\_next;

}

file\_to\_cout\_V.close();

file\_to\_cout\_X.close();

}

}

void z3\_diff\_eq() {

double kx1, kx2, kx3, kx4, kv1, kv2, kv3, kv4;

double h;

double v\_prev, v\_next, x\_prev, x\_next;

vector<double> n{ 10,100,1000,10000 };

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

ofstream file\_to\_cout\_V("z3\_v" + to\_string(i) + ".txt");

ofstream file\_to\_cout\_X("z3\_x" + to\_string(i) + ".txt");

v\_prev = 0.09, x\_prev = 0;

v\_next = 0, x\_next = 0;

h = 10. / (n[i] - 1);

file\_to\_cout\_V << v\_prev << endl;

file\_to\_cout\_X << x\_prev << endl;

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

kx1 = v\_prev;

kx2 = v\_prev + h / 2. \* kx1;

kx3 = v\_prev + h / 2. \* kx2;

kx4 = v\_prev + h \* kx3;

kv1 = U(x\_prev);

kv2 = U(x\_prev + h / 2. \* kv1);

kv3 = U(x\_prev + h / 2. \* kv2);

kv4 = U(x\_prev + h \* kv3);

x\_next = (kx1 + 2 \* kx2 + 2 \* kx3 + kx4) \* h / 6 + x\_prev;

v\_next = (kv1 + 2 \* kv2 + 2 \* kv3 + kv4) \* h / 6 + v\_prev;

file\_to\_cout\_V << v\_prev << endl;

file\_to\_cout\_X << x\_prev << endl;

v\_prev = v\_next;

x\_prev = x\_next;

}

file\_to\_cout\_V.close();

file\_to\_cout\_X.close();

}

}

void z4\_diff\_eq() {

double h;

double b = 1, a = 0;

vector<double> n{ 10,100,1000,10000 };

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

ofstream file\_to\_cout\_V("z4\_u" + to\_string(i) + ".txt");

vector<double> A(n[i]), B(n[i]), C(n[i]), F(n[i]);

h = (b - a) / n[i];

A[0] = 0; // edge

B[0] = 1; // in edge indexes

C[0] = 0; // in edge indexes

F[0] = 0;

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

A[j] = 1 / h / h + 2 / h;

B[j] = -2 / h / h;

C[j] = 1 / h / h - 2 / h;

F[j] = func(a + j \* h);

}

A[n[i] - 1] = -1 / h;

B[n[i] - 1] = 1 + 1 / h;

C[n[i] - 1] = 0; //edge of sys

F[n[i] - 1] = -3.5625;

vector<double> ANS = TripleDiag(A, B, C, F);

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

file\_to\_cout\_V << ANS[i] << endl;

}

}

}

void z5\_diff\_eq()

{

//u''-4u'=6x^2+1

//u(a)=0

//u(b)+v(b)=-3.5625

ofstream X\_5("X5\_out.txt");

ofstream V\_5("V5\_out.txt");

ofstream U\_5("U5\_out.txt");

int n = 1000;

double p = find\_p();

double k1, k2, k3, k4;

double q1, q2, q3, q4;

double h = 1. / n;

double v\_0, u\_n, v\_n, u\_0;

u\_0 = v\_0 = 0;

u\_n = p;

v\_n = u\_n - 3.5625;

double x = 0;

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

{

X\_5 << x << endl;

U\_5 << u\_n << endl;

V\_5 << v\_n << endl;

k1 = f1(x, v\_n, u\_n);

q1 = f2(x, v\_n, u\_n);

k2 = f1(x + h / 3, v\_n + h \* k1 / 3, u\_n + h \* q1 / 3);

q2 = f2(x + h / 3, v\_n + h \* k1 / 3, u\_n + h \* q1 / 3);

k3 = f1(x + 2 \* h / 3, v\_n - h \* k1 / 3, u\_n - h \* q1 / 3 + h \* q2);

q3 = f2(x + 2 \* h / 3, v\_n - h \* k1 / 3, u\_n - h \* q1 / 3 + h \* q2);

k4 = f1(x + h, v\_n + h \* k1 - h \* k2 + h \* k3, u\_n + h \* q1 - h \* q2 + h \* q3);

q4 = f2(x + h, v\_n + h \* k1 - h \* k2 + h \* k3, u\_n + h \* q1 - h \* q2 + h \* q3);

v\_n += h \* (k1 + 3 \* k2 + 3 \* k3 + k4) / 8;

u\_n += h \* (q1 + 3 \* q2 + 3 \* q3 + q4) / 8;

x += i \* h;

}

V\_5.close();

X\_5.close();

U\_5.close();

}

int main() {

int ZADANIE = 0;

cout << "Ex: ";

cin >> ZADANIE;

switch (ZADANIE)

{

case 1:

z1\_diff\_eq();

break;

case 2:

z2\_diff\_eq();

break;

case 3:

z3\_diff\_eq();

break;

case 4:

z4\_diff\_eq();

break;

case 5:

z5\_diff\_eq();

break;

}

}