МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАВЧАЛЬНО-НАУКОВИЙ КОМПЛЕКС

«ІНСТИТУТ ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ»

НАЦІОНАЛЬНОГО ТЕХНІЧНОГО УНІВЕРСИТЕТУ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

КАФЕДРА МАТЕМАТИЧНИХ МЕТОДІВ СИСТЕМНОГО АНАЛІЗУ

Лабораторна робота №8

з курсу «Чисельні методи»

Виконав: студент 3 курсу

групи КА-23

Деундяк О.В.

Прийняла: Кузнєцова Н. В.

Київ – 2014р.

**Умова**

Знайти розв’язок крайової задачі:  
  
**E:\VS\ChM\Lab8\prob.png**E:\VS\ChM\Lab8\prob1.png

**Текст програми**

**main.cpp**

// Ay = y'' - y' + 2y / x

// a1y = y(1, 1) - 0.5y'(1, 1)

// a2y = y'(1, 4)

// -0.282, 1.981, 2.065, -0.577, -2.029

// y(x) = ax^2 +bx + c + (dx+e)^-1

#include <functional>

#include <fstream>

#include <iostream>

#include "FDM.h"

using namespace std;

const double a = -0.282, b = 1.981, c = 2.065, d = -0.577, e = -2.029;

void test()

{

func y = [](double x)->double {return a\*x\*x + b\*x + c + 1 / (d\*x + e); };

func yd = [](double x)->double {return 2 \* a\*x + b - d / pow(d\*x + e, 2); };

func f = [](double x)->double {return 2 \* a - b

+ 2 \* d\*d / pow(d\*x + e, 3)

- 2 \* a\*x + d / pow(d\*x + e, 2)

+ 2 / x\*(a\*x\*x + b\*x + c + 1 / (d\*x + e)); };

func p = [](double x)->double {return -1; }, q = [](double x)->double {return 2 / x; };

double kda = -0.5, ka = 1;

double kdb = 1, kb = 0;

double xa = 1.1, f1 = ka\*y(xa) + kda\*yd(xa);

double xb = 1.4, f2 = kb\*y(xb) + kdb\*yd(xb);

ofstream err("error.txt");

for (size\_t N = 10; N <= 10000; N \*= 2)

{

vector<double> res = FDM::Result(p, q, f, xa, kda, ka, f1, xb, kdb, kb, f2, N);

double x = xa, h = (xb - xa) / N;

double error = abs(y(x) - res[0]);

for (double yr : res)

{

double t = abs(y(x) - yr);

if (t > error)

error = t;

x += h;

}

err << h << '\t' << error << endl;

}

}

int main()

{

test();

cin.get();

}

**FDM.h**

#pragma once

#include <vector>

#include <functional>

typedef std::function<double(double)> func;

class FDM

{

public:

static std::vector<double> Result(func p, func q, func f,

double a, double kda, double ka, double f1,

double b, double kdb, double kb, double f2,

size\_t N);

};

**FDM.cpp**

#include "FDM.h"

#include "TMA.h"

#include <iostream>

std::vector<double> FDM::Result(func p, func q, func f,

double a, double kda, double ka, double f1,

double b, double kdb, double kb, double f2,

size\_t N)

{

TMA tma(N + 1);

double h = (b - a) / N, x = a + h;

tma.Iterate(0., -kda / h + ka, kda / h, f1);

for (size\_t i = 1; i < N; i++, x += h)

tma.Iterate(1 / (h\*h) - p(x) / (2 \* h), -2 / (h\*h) + q(x), 1 / (h\*h) + p(x) / (2 \* h), f(x));

tma.Iterate(-kdb/h, kdb/h + kb, 0, f2);

return tma.Result();

}

**TMA.h**

#pragma once

#include <vector>

class TMA

{

std::vector<double> c, d, r;

size\_t i; // number of iterations

size\_t n; // size of system

public:

TMA(size\_t n) : i(0), c(n), d(n), r(n)

{

this->n = n;

}

std::vector<double> Result();

void Iterate(double a, double b, double c, double d);

};

**TMA.cpp**

#include "TMA.h"

void TMA::Iterate(double A, double B, double C, double D)

{

if (i == n)

return;

if (i == 0)

{

c[i] = C / B;

d[i] = D / B;

}

else

{

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

d[i] = (D - A\*d[i - 1]) / (B - A\*c[i - 1]);

}

i++;

}

std::vector<double> TMA::Result()

{

if (i != n)

return std::vector<double>();

r[n - 1] = d[n - 1];;

for (size\_t i = n - 1; i > 0; i--)

r[i - 1] = d[i - 1] - c[i - 1] \* r[i];

return r;

};

**Результати роботи програми**

h=0.015

x=1.1 y=3.52746 yr=3.52238 err=0.00508632

x=1.115 y=3.54902 yr=3.54385 err=0.00517151

x=1.13 y=3.57045 yr=3.56519 err=0.00525589

x=1.145 y=3.59174 yr=3.5864 err=0.00533943

x=1.16 y=3.6129 yr=3.60748 err=0.00542211

x=1.175 y=3.63392 yr=3.62842 err=0.00550393

x=1.19 y=3.65481 yr=3.64923 err=0.00558486

x=1.205 y=3.67557 yr=3.6699 err=0.00566489

x=1.22 y=3.69618 yr=3.69044 err=0.00574399

x=1.235 y=3.71667 yr=3.71085 err=0.00582215

x=1.25 y=3.73702 yr=3.73112 err=0.00589936

x=1.265 y=3.75724 yr=3.75126 err=0.00597559

x=1.28 y=3.77732 yr=3.77127 err=0.00605083

x=1.295 y=3.79727 yr=3.79115 err=0.00612507

x=1.31 y=3.81709 yr=3.81089 err=0.00619829

x=1.325 y=3.83677 yr=3.8305 err=0.00627046

x=1.34 y=3.85632 yr=3.84997 err=0.00634158

x=1.355 y=3.87573 yr=3.86932 err=0.00641163

x=1.37 y=3.89501 yr=3.88853 err=0.00648059

x=1.385 y=3.91416 yr=3.90761 err=0.00654845

x=1.4 y=3.93317 yr=3.92655 err=0.0066152

**Висновок**

Графік залежності помилки від кроку:

Графік помилки лінійний, оскільки обраний метод має точність O(h).