Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

**(ФГБОУ ВО «КубГУ»)**

**Факультет компьютерных технологий и прикладной математики**

**Кафедра вычислительных технологий**

**Отчёт**

**по дополнительной задаче**

**Дисциплина: Алгоритмы вычислительной математики**

**Тема: «рЕШЕНИЕ СИСТЕМ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ»**

Работу выполнил \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Д.Е Гиренко

Направление подготовки 02.03.02 Фундаментальная информатика и

информационные технологии

Направленность (профиль) Математическое и программное обеспечение

компьютерных технологий

Преподаватель

ассистент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В. И. Шиян

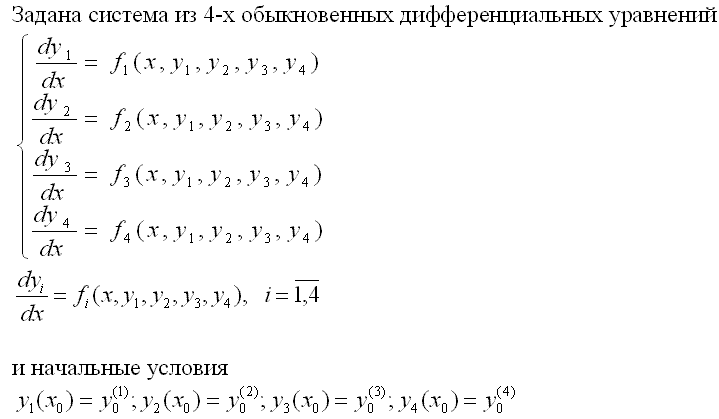
**1 Постановка задачи**

Дано:

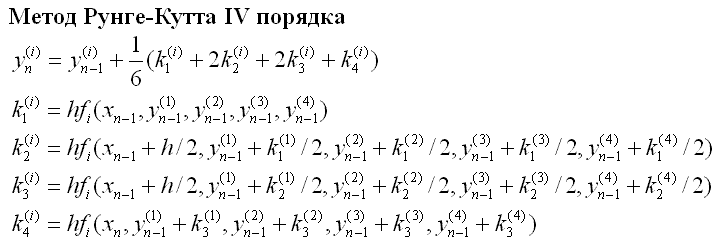
1. , … , – система дифференциальных уравнений в каноническом виде.
2. – начальное условие
3. – конец отрезка изображения. – заданная точность.

Задача: решить систему дифференциальных уравнений методом Рунге-Кутта 4-го порядка и построить графики решения, используя кубические сплайны.

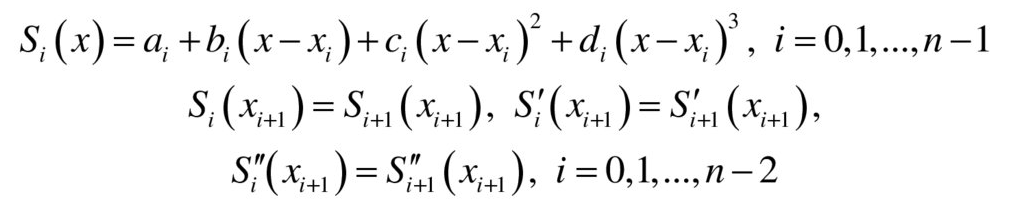
Пусть система задана формулами и условиями (1):

 (1)

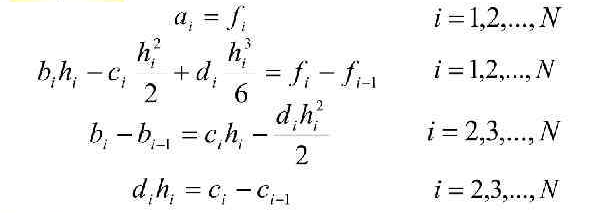
Выбираем достаточный для заданной точности шаг и итерационно считаем последующие точки функции по формулам (2):

 (2)

Получив множество точек, используем метод интерполяции кубическими сплайнами (3), чтобы получить график.

 (3)

С помощью метода прогонки найдём коэффициенты сплайнов по формулам (4):

 (6)

Выведем график с помощью полученной функции.

**2 Реализация на языке С++**

Код реализован на языке С++ с использованием стандартных библиотек.

#include <vector>

#include <iostream>

#include <fstream>

#include <string>

#include <math.h>

#include <sstream>

#include <Windows.h>

typedef long double ll;

using namespace std;

//String To Znach

struct listik

{

listik\* n = NULL, \* b = NULL;

string inf;

};

void ladd(listik \* l, string s)

{

listik\* r = new listik;

r->inf = s;

l->n = r;

r->b = l;

}

listik\* lnext(listik \* l, ll x)

{

listik\* p = l;

for (; x > 0; x--)

{

p = p->n;

}

return p;

}

bool isznak(char i)

{

vector<char> oper = { '+','-','/','\*','^' };

for (auto u : oper)

if (u == i)

return true;

return false;

}

ll foundback(listik \* vir)

{

listik\* p = vir;

ll n = 0;

ll k = 1;

while (k)

{

if (p->inf == "(")

k++;

else if (p->inf == ")")

k--;

n++;

p = p->n;

}

return n - 1;

}

string podchetak(listik \* vir, ll & kol)

{

listik\* p = vir;

ll i = 0;

while (i < kol - 1)

{

if (p->inf == "ln")

{

ll skobkol = foundback(p->n->n);

stringstream stoll(podchetak(p->n->n, skobkol));

ll itg;

stoll >> itg;

itg = log(itg);

string bac = to\_string(itg);

p->inf = bac;

p->n = lnext(p, skobkol + 3);

if (p->n != NULL) p->n->b = p;

kol -= (skobkol + 2);

}

else if (p->inf == "sin")

{

ll skobkol = foundback(p->n->n);

stringstream stoll(podchetak(p->n->n, skobkol));

ll itg;

stoll >> itg;

itg = sin(itg);

string bac = to\_string(itg);

p->inf = bac;

p->n = lnext(p, skobkol + 3);

if (p->n != NULL) p->n->b = p;

kol -= (skobkol + 2);

}

else if (p->inf == "cos")

{

ll skobkol = foundback(p->n->n);

stringstream stoll(podchetak(p->n->n, skobkol));

ll itg;

stoll >> itg;

itg = cos(itg);

string bac = to\_string(itg);

p->inf = bac;

p->n = lnext(p, skobkol + 3);

if (p->n != NULL) p->n->b = p;

kol -= (skobkol + 2);

}

else if (p->inf == "tan")

{

ll skobkol = foundback(p->n->n);

stringstream stoll(podchetak(p->n->n, skobkol));

ll itg;

stoll >> itg;

itg = tan(itg);

string bac = to\_string(itg);

p->inf = bac;

p->n = lnext(p, skobkol + 3);

if (p->n != NULL) p->n->b = p;

kol -= (skobkol + 2);

}

else if (p->inf == "sqrt")

{

ll skobkol = foundback(p->n->n);

stringstream stoll(podchetak(p->n->n, skobkol));

ll itg;

stoll >> itg;

itg = sqrt(itg);

string bac = to\_string(itg);

p->inf = bac;

p->n = lnext(p, skobkol + 3);

if (p->n != NULL) p->n->b = p;

kol -= (skobkol + 2);

}

else if (p->inf == "asin")

{

ll skobkol = foundback(p->n->n);

stringstream stoll(podchetak(p->n->n, skobkol));

ll itg;

stoll >> itg;

itg = asin(itg);

string bac = to\_string(itg);

p->inf = bac;

p->n = lnext(p, skobkol + 3);

if (p->n != NULL) p->n->b = p;

kol -= (skobkol + 2);

}

else if (p->inf == "acos")

{

ll skobkol = foundback(p->n->n);

stringstream stoll(podchetak(p->n->n, skobkol));

ll itg;

stoll >> itg;

itg = acos(itg);

string bac = to\_string(itg);

p->inf = bac;

p->n = lnext(p, skobkol + 3);

if (p->n != NULL) p->n->b = p;

kol -= (skobkol + 2);

}

else if (p->inf == "atan")

{

ll skobkol = foundback(p->n->n);

stringstream stoll(podchetak(p->n->n, skobkol));

ll itg;

stoll >> itg;

itg = atan(itg);

string bac = to\_string(itg);

p->inf = bac;

p->n = lnext(p, skobkol + 3);

if (p->n != NULL) p->n->b = p;

kol -= (skobkol + 2);

}

if (p->n != NULL)

p = p->n;

i++;

}

p = vir;

i = 0;

while (i < kol - 1)

{

if (p->inf == "(")

{

ll skobkol = foundback(p->n);

string bak = podchetak(p->n, skobkol);

p->inf = bak;

p->n = lnext(p, skobkol + 2);

if (p->n != NULL) p->n->b = p;

kol -= (skobkol + 1);

}

if (p->n != NULL)

p = p->n;

i++;

}

p = vir;

i = 0;

while (i < kol - 1)

{

if (p->inf == "^")

{

stringstream stoll1(p->b->inf);

stringstream stoll2(p->n->inf);

ll s1, s2;

stoll1 >> s1;

stoll2 >> s2;

p = p->b;

p->inf = to\_string(pow(s1, s2));

p->n = lnext(p, 3);

if (p->n != NULL)

p->n->b = p;

kol -= 2;

i--;

}

if (p->n != NULL)

p = p->n;

i++;

}

p = vir;

i = 0;

while (i < kol - 1)

{

if (p->inf == "\*")

{

stringstream stoll1(p->b->inf);

stringstream stoll2(p->n->inf);

ll s1, s2;

stoll1 >> s1;

stoll2 >> s2;

p = p->b;

p->inf = to\_string(s1 \* s2);

p->n = lnext(p, 3);

if (p->n != NULL) p->n->b = p;

kol -= 2;

i--;

}

else if (p->inf == "/")

{

stringstream stoll1(p->b->inf);

stringstream stoll2(p->n->inf);

ll s1, s2;

stoll1 >> s1;

stoll2 >> s2;

p = p->b;

p->inf = to\_string(s1 / s2);

p->n = lnext(p, 3);

if (p->n != NULL) p->n->b = p;

kol -= 2;

i--;

}

if (p->n != NULL)

p = p->n;

i++;

}

p = vir;

i = 0;

while (i < kol - 1)

{

if (p->inf == "+")

{

stringstream stoll1(p->b->inf);

stringstream stoll2(p->n->inf);

ll s1, s2;

stoll1 >> s1;

stoll2 >> s2;

p = p->b;

p->inf = to\_string(s1 + s2);

p->n = lnext(p, 3);

if (p->n != NULL) p->n->b = p;

kol -= 2;

i--;

}

else if (p->inf == "-")

{

stringstream stoll1(p->b->inf);

stringstream stoll2(p->n->inf);

ll s1, s2;

stoll1 >> s1;

stoll2 >> s2;

p = p->b;

p->inf = to\_string(s1 - s2);

p->n = lnext(p, 3);

if (p->n != NULL) p->n->b = p;

kol -= 2;

i--;

}

if (p->n != NULL)

p = p->n;

i++;

}

return(vir->inf);

// ищем скобки, если нашли, вызываем от них подсчёт,

// если скобок нет, то ищем все возведения в степень слева направо, считаем их

// потом ищем все \* и \, потом + и -

}

bool chetak(string s, ll & x) // проверяет правильность выражения и считает его

{

ll skob = 0;

for (auto u : s)

{

if (u == '(')

skob++;

else if (u == ')')

{

skob--;

if (skob < 0)

return false;

}

}

if (skob != 0)

return false;

string ss = " ";

for (auto u : s)

{

if (u != ' ')

ss += {u};

}

ss += " ";

listik\* delet = new listik;

listik\* p = delet;

bool f = true;

ll kol = 0;

bool otr = true;

bool isotr = false;

for (ll i = 1; i < ss.size() - 1; i++, f = !f)

{

if (f)

{

// число

if (ss[i] >= '0' && ss[i] <= '9')

{

bool toch = true;

string nowchis = { ss[i] };

while (true)

{

if (toch)

{

if (!((ss[i + 1] >= '0' && ss[i + 1] <= '9') || ss[i + 1] == '.'))

{

if (!isotr)

ladd(p, nowchis);

else

ladd(p, "-" + nowchis);

isotr = false;

p = p->n;

kol++;

break;

}

if (ss[i + 1] == '.')

{

nowchis += '.';

toch = false;

if (!(ss[i + 1 + 1] >= '0' && ss[i + 1 + 1] <= '9'))

return false;

}

else

{

nowchis += {ss[i + 1]};

}

}

else

{

if (!(ss[i + 1] >= '0' && ss[i + 1] <= '9'))

{

if (!isotr)

ladd(p, nowchis);

else

ladd(p, "-" + nowchis);

isotr = false;

p = p->n;

kol++;

break;

}

else

{

nowchis += {ss[i + 1]};

}

}

i++;

}

otr = false;

}

else if (ss[i] == 'e')

{

if (!isotr)

ladd(p, to\_string(exp(1)));

else

ladd(p, to\_string(-exp(1)));

p = p->n;

kol++;

otr = false;

isotr = false;

}

else if (ss[i] == 'p')

{

if (ss[i + 1] == 'i')

{

if (!isotr)

ladd(p, to\_string(asin(1) \* 2));

else

ladd(p, to\_string(-asin(1) \* 2));

p = p->n;

kol++;

i++;

otr = false;

isotr = false;

}

else

return false;

}

else if (ss[i] == 'l')

{

if (ss[i + 1] == 'n')

{

if (ss[i + 2] == '(')

{

ladd(p, "ln");

p = p->n;

ladd(p, "(");

p = p->n;

kol++;

kol++;

i += 2;

f = !f;

otr = true;

}

else return false;

}

else return false;

}

else if (ss[i] == 'c')

{

if (ss[i + 1] == 'o')

{

if (ss[i + 2] == 's')

{

if (ss[i + 3] == '(')

{

ladd(p, "cos");

p = p->n;

ladd(p, "(");

p = p->n;

kol++;

kol++;

i += 3;

f = !f;

otr = true;

}

else return false;

}

else return false;

}

else return false;

}

else if (ss[i] == 't')

{

if (ss[i + 1] == 'a')

{

if (ss[i + 2] == 'n')

{

if (ss[i + 3] == '(')

{

ladd(p, "tan");

p = p->n;

ladd(p, "(");

p = p->n;

kol++;

kol++;

i += 3;

f = !f;

otr = true;

}

else return false;

}

else return false;

}

else return false;

}

else if (ss[i] == 's')

{

if (ss[i + 1] == 'i')

{

if (ss[i + 2] == 'n')

{

if (ss[i + 3] == '(')

{

ladd(p, "sin");

p = p->n;

ladd(p, "(");

p = p->n;

kol++;

kol++;

i += 3;

f = !f;

otr = true;

}

else return false;

}

else return false;

}

else if (ss[i + 1] == 'q')

{

if (ss[i + 2] == 'r')

{

if (ss[i + 3] == 't')

{

if (ss[i + 4] == '(')

{

ladd(p, "sqrt");

p = p->n;

ladd(p, "(");

p = p->n;

kol++;

kol++;

i += 4;

f = !f;

otr = true;

}

else return false;

}

else return false;

}

else return false;

}

else return false;

}

else if (ss[i] == 'a')

{

if (ss[i + 1] == 'c')

{

if (ss[i + 2] == 'o')

{

if (ss[i + 3] == 's')

{

if (ss[i + 4] == '(')

{

ladd(p, "acos");

p = p->n;

ladd(p, "(");

p = p->n;

kol++;

kol++;

i += 4;

f = !f;

otr = true;

}

else return false;

}

else return false;

}

else return false;

}

else if (ss[i + 1] == 't')

{

if (ss[i + 2] == 'a')

{

if (ss[i + 3] == 'n')

{

if (ss[i + 4] == '(')

{

ladd(p, "atan");

p = p->n;

ladd(p, "(");

p = p->n;

kol++;

kol++;

i += 4;

f = !f;

otr = true;

}

else return false;

}

else return false;

}

else return false;

}

else if (ss[i + 1] == 's')

{

if (ss[i + 2] == 'i')

{

if (ss[i + 3] == 'n')

{

if (ss[i + 4] == '(')

{

ladd(p, "asin");

p = p->n;

ladd(p, "(");

p = p->n;

kol++;

kol++;

i += 4;

f = !f;

otr = true;

}

else return false;

}

else return false;

}

else return false;

}

else return false;

}

else if (ss[i] == '(')

{

ladd(p, "(");

p = p->n;

kol++;

f = !f;

otr = true;

}

else if (ss[i] == '-' && otr)

{

f = !f;

isotr = true;

otr = false;

}

else

{

return false;

}

}

else

{

otr = false;

// znak

if (isznak(ss[i]))

{

ladd(p, { ss[i] });

p = p->n;

kol++;

if (ss[i + 1] == ' ')

return false;

}

// skoba

else if (ss[i] == ')')

{

ladd(p, { ss[i] });

p = p->n;

kol++;

f = !f;

}

//err

else

{

return false;

}

}

}

if (isotr)

return false;

delet = delet->n;

delet->b = NULL;

string itogo = podchetak(delet, kol);

stringstream sss(itogo);

sss >> x;

return true;

}

//

// Func to Znach

ll useFunc(string f, vector<ll> &args)

{

for (long i = args.size()-1; i >=0; i--)

{

for (long j = 0; j<f.size();j++)

if (f[j] == 'x')

{

string tmp;

for (long k = 1; k <= to\_string(i).size() && j + k < f.size(); k++)

tmp += f[j + k];

if (tmp == to\_string(i))

if (args[i] < 0)

f = f.replace(j, to\_string(i).size()+1, "("+to\_string(args[i])+")");

else

f = f.replace(j, to\_string(i).size(), to\_string(args[i]));

}

}

ll ans = 0;

if (!chetak(f, ans))

{

cout << "Неправильный формат ввода! Проверьте правильность функций.";

return -1;

}

return ans;

}

//

// Points to Splain

vector<ll> progonka(vector<ll> f, vector<ll> h, int n)

{

//vvod

vector <ll> b;

vector <ll> d;

vector <ll> rez(n);

for (ll i = 0; i < n - 2; i++)

{

b.push\_back(2 \* (h[i + 1] + h[i + 2]));

}

for (ll i = 0; i < n - 2; i++)

{

d.push\_back(6 \* ((f[i + 2] - f[i + 1]) / h[i + 2] - (f[i + 1] - f[i]) / h[i + 1]));

}

//koef chet

vector <ll> y(n - 2);

vector <ll> u(n - 2); // альфи

vector <ll> o(n - 2); // бета

y[0] = b[0]; u[0] = h[2] / y[0]; o[0] = d[0] / y[0];

for (ll i = 1; i < n - 3; i++)

{

y[i] = b[i] + h[i + 1] \* u[i - 1];

u[i] = h[i + 2] / y[i];

o[i] = (d[i] - h[i + 1] \* o[i - 1]) / y[i];

}

y[n - 3] = b[n - 3] + h[n - 2] \* u[n - 4];

o[n - 3] = (d[n - 3] - h[n - 2] \* o[n - 4]) / y[n - 3];

// otvet

rez[0] = 0; rez[n - 1] = 0;

rez[n - 2] = o[n - 3];

for (ll i = n - 4; i != -1; i--)

{

rez[i + 1] = u[i] \* rez[i + 2] + o[i];

}

//end

vector<ll> nevyaz;

nevyaz.push\_back(rez[1] \* b[0] - rez[2] \* h[2] - d[0]);

for (ll i = 2; i < n - 4; i++)

nevyaz.push\_back(rez[i] \* h[2] + rez[i + 1] \* b[2] - rez[i + 2] \* h[2] - d[i]);

nevyaz.push\_back(rez[n - 3] \* h[2] + rez[n - 2] \* b[2] - d[n - 3]);

return rez;

}

ll splain(ll x, vector<vector<ll>>& koef, vector<ll>& p, ll i)

{

ll tmp = (x - p[i]);

ll tmp2 = pow(tmp, 2);

ll tmp3 = pow(tmp, 3);

ll rez = koef[0][i] + koef[1][i] \* tmp + koef[2][i] \* pow(tmp, 2) / 2 + koef[3][i] \* pow(tmp, 3) / 6;

return rez;

}

//

int main()

{

SetConsoleCP(1251); // Подключаем ввод русских букв

SetConsoleOutputCP(1251);

int n = 0; double e = 1; //размерность, погрешность

double xo = 0, xm = 1; // начальная и конечная точки

vector<string> func = \*(new vector<string>()); // список функций в виде строк

vector<ll> nowFuncZnach = \*(new vector<ll>()); // список начальных условий

vector<ll> xPoints = \*(new vector<ll>()); //список иксов

vector<vector<ll>> yPoints = \*(new vector<vector<ll>>()); // список игриков

try

{

ifstream sr("func.txt");

sr >> n;

sr >> e; //две строки для размерности и погрешности

func.clear();

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

{

string tmp; sr >> tmp;

func.push\_back(tmp); // n строк записи функций, переменные обозначаются как xi

// , где i - число от 1 и до n;

// Аргумент функций обозначается как х0;

}

sr >> xo; // единая начальная точка

sr >> xm; // единая конечная точка

nowFuncZnach.clear();

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

{

double tmp; sr >> tmp;

nowFuncZnach.push\_back(tmp); // значения в этой точке

}

sr.close();

}

catch(double) { cout << "Неправильный формат ввода! Проверьте файл"; return 0; }

ll h, x = xo;

long kol = (long)((xm - xo) / e ) + 1;

if (kol < 5)

kol=6;

h = (xm - xo) / kol;

xPoints.push\_back(xo);

yPoints.push\_back(nowFuncZnach);

while (x < xm)

{

vector<ll> k1;

vector<ll> k2;

vector<ll> k3;

vector<ll> k4;

vector<ll> args;

vector<ll> nextZnach;

args.push\_back(0);

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

args.push\_back(0);

args[0] = x;

for (long j = 1; j < n + 1; j++)

args[j] = nowFuncZnach[j - 1];

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

k1.push\_back( h \* useFunc(func[i], args));

args[0] += h / 2;

for (long j = 1; j < n + 1; j++)

args[j] = nowFuncZnach[j - 1] + k1[j - 1] / 2;

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

k2.push\_back(h \* useFunc(func[i], args));

for (long j = 1; j < n + 1; j++)

args[j] = nowFuncZnach[j - 1] + k2[j - 1] / 2;

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

k3.push\_back(h \* useFunc(func[i], args));

args[0] = x;

for (long j = 1; j < n + 1; j++)

args[j] = nowFuncZnach[j - 1] + k3[j - 1];

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

{

k4.push\_back(h \* useFunc(func[i], args));

nextZnach.push\_back(nowFuncZnach[i] + (k1[i] + 2 \* k2[i] + 2 \* k3[i] + k4[i]) / 6.0);

}

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

{

nowFuncZnach[i] = nextZnach[i];

}

x += h;

xPoints.push\_back(x);

yPoints.push\_back(nextZnach);

}

//

ll multi = 100;

//

Sleep(2000);

m:

n = xPoints.size();

for (long l = 0; l < yPoints[0].size(); l++)

{

HDC hDC = GetDC(GetConsoleWindow());

vector<ll> hVector;

hVector.push\_back(-1);

for (ll i = 1; i < n; i++)

{

hVector.push\_back(xPoints[i] - xPoints[i - 1]);

}

vector<ll> aa;

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

{

aa.push\_back(yPoints[i][l]);

}

vector<ll> c;

c = progonka(aa, hVector, xPoints.size());

vector<ll> d;

d.push\_back(-1);

for (ll i = 1; i < n; i++)

{

d.push\_back((c[i] - c[i - 1]) / hVector[i]);

}

vector<ll> bb;

bb.push\_back(-1);

for (ll i = 1; i < n; i++)

{

bb.push\_back((aa[i] - aa[i - 1]) / hVector[i] + hVector[i] / 2 \* c[i] - hVector[i] \* hVector[i] / 6 \* d[i]); //- (2\*c[i]+c[i-1])\*h[i]/3);

}

vector<vector<ll>> koef;

koef.push\_back(aa);

koef.push\_back(bb);

koef.push\_back(c);

koef.push\_back(d);

HPEN Pen;

if (l == 0)

{

HPEN Sen = CreatePen(PS\_SOLID, 2, RGB(255, 255, 255));

SelectObject(hDC, Sen);

MoveToEx(hDC, 0, 850, NULL);

LineTo(hDC, 2000, 850);

MoveToEx(hDC, 1000, 0, NULL);

LineTo(hDC, 1000, 1700);

}

for (int i = l / 3; i > 0; i--)

cout << "Тёмно";

if (l % 3 == 0)

{

Pen = CreatePen(PS\_SOLID, 2, RGB((int)((yPoints[0].size() - l) \* 255 / yPoints[0].size()), 0, 0));

cout << "Красный:" << func[l]<<endl;

}

else if (l % 3 == 1)

{

Pen = CreatePen(PS\_SOLID, 2, RGB(0, (int)((yPoints[0].size() - l) \* 255 / yPoints[0].size()), 0));

cout << "Зелёный:" << func[l] << endl;;

}

else

{

Pen = CreatePen(PS\_SOLID, 2, RGB(0, 0, (int)((yPoints[0].size() - l) \* 255 / yPoints[0].size())));

cout << "Синий:" << func[l] << endl;;

}

SelectObject(hDC, Pen);

ll i = 1;

MoveToEx(hDC, multi \* xo + 1000, -multi \* splain(xo, koef, xPoints, i) + 850, NULL);

ll step = (xm - xo) / (multi \* multi);

for (x = xo; x <= xm; x += step) // O(100,85) - center

{

ll elka = -multi \* splain(x, koef, xPoints, i) + 850;

//MoveToEx(hDC, 100 \* x + 1000, elka, NULL);//10 - scale

while (i != n-1 && x >= xPoints[i])

i++;

LineTo(hDC, multi \* x + 1000, elka);

}

hVector.clear();

aa.clear();

bb.clear();

d.clear();

c.clear();

koef.clear();

}

cin >> multi;

system("cls");

goto m;

}

На рисунке 1 приведены графики решения системы:

Изображение выглядит как текст, вода, легкий, другой

Автоматически созданное описание

Рисунок 1 – График решения системы