**Федеральное государственное бюджетное образовательное учреждение «Сибирский государственный университет телекоммуникаций и информатики»**

Кафедра ПМ и К

**КУРСОВАЯ РАБОТА**

По дисциплине «Вычислительная математика»

Вариант 7

Выполнил:

студент гр. ИВ-621

Евтушенко Д.А

Проверил:

Чирихин К. С.

Новосибирск 2018

**Оглавление**

Постановка задачи................................................................................................3

Описание алгоритма программы.........................................................................4

Результат работы программы..............................................................................5

Листинг .................................................................................................................8

**Постановка задачи**

Вариант №7

Решить краевую задачу методом Рунге-Кутта IIпорядка с усреднением по времени.

Построить графики функции y(x) и интерполяционного многочлена P5(x) (интерполяция по точкам x=0; 0.2;0.4; 0.6; 0.8; 1.0). Найти интеграл

**Описание алгоритма программы**

1. Находим первую производную *методом стрельбы*.
2. Решаем краевую задачу Коши методом *Рунге-Кутта II* порядка с усреднением по времени.
3. Интерполяция (полиномом Лагранжа)
4. Численное интегрирование (формула Симпсона)

**Метод стрельбы** - [численный метод](https://ru.wikipedia.org/wiki/%D0%A7%D0%B8%D1%81%D0%BB%D0%B5%D0%BD%D0%BD%D1%8B%D0%B9_%D0%BC%D0%B5%D1%82%D0%BE%D0%B4), заключающийся в сведении [краевой задачи](https://ru.wikipedia.org/wiki/%D0%9A%D1%80%D0%B0%D0%B5%D0%B2%D0%B0%D1%8F_%D0%B7%D0%B0%D0%B4%D0%B0%D1%87%D0%B0) к некоторой [задаче Коши](https://ru.wikipedia.org/wiki/%D0%97%D0%B0%D0%B4%D0%B0%D1%87%D0%B0_%D0%9A%D0%BE%D1%88%D0%B8) для той же системы [дифференциальных уравнений](https://ru.wikipedia.org/wiki/%D0%94%D0%B8%D1%84%D1%84%D0%B5%D1%80%D0%B5%D0%BD%D1%86%D0%B8%D0%B0%D0%BB%D1%8C%D0%BD%D1%8B%D0%B5_%D1%83%D1%80%D0%B0%D0%B2%D0%BD%D0%B5%D0%BD%D0%B8%D1%8F).

Для пристрелки выбираем два параметра: a = y(0),b -произвольное число.

Если по a перелет, а по b недолет, останавливаем корректировку, на данном этапе отрезок найден.

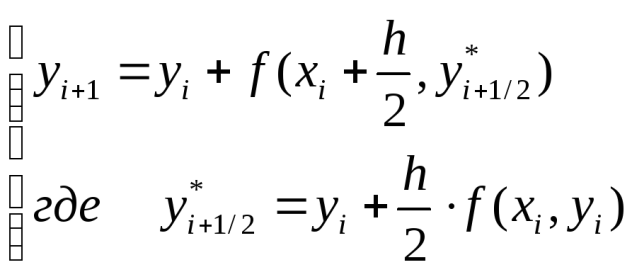
После пристрелки и определения интервала [a,b], где идёт смена знака, запускаем МПД или МХ.На практике это выглядит так, как будто мы

решаем уравнение q(k), где возвращаетq(k) решение задачи Коши в точке b при

заданном k.

**Задача Коши**

Формула Рунге-Кутта 2го порядка с усреднением по времени.



Так как ДУ не может быть решено относительно старшей производной, то тогда на каждом шаге решаем нелинейное уравнение относительно y(n)(все остальные неизвестные y,y’,y”,…, y(n-1)-к этому моменту уже известны).

Решать уравнение относительно старшей производной любым методом(Хорд, МПД, Ньютона).

**Интерполяция (полиномом Лагранжа)**

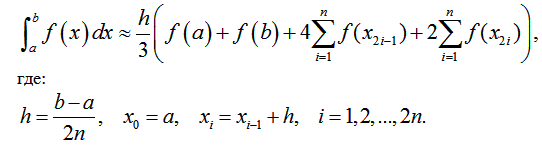
[Лагранж](https://ru.wikipedia.org/wiki/%D0%9B%D0%B0%D0%B3%D1%80%D0%B0%D0%BD%D0%B6,_%D0%96%D0%BE%D0%B7%D0%B5%D1%84_%D0%9B%D1%83%D0%B8) предложил способ вычисления таких многочленов:

где базисные полиномы определяются по формуле:

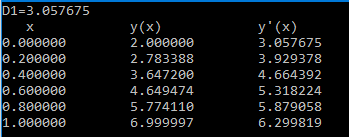
{\displaystyle l\_{i}(x)=\prod \_{j=0,j\neq i}^{n}{\frac {x-x\_{j}}{x\_{i}-x\_{j}}}={\frac {x-x\_{0}}{x\_{i}-x\_{0}}}\cdots {\frac {x-x\_{i-1}}{x\_{i}-x\_{i-1}}}\cdot {\frac {x-x\_{i+1}}{x\_{i}-x\_{i+1}}}\cdots {\frac {x-x\_{n}}{x\_{i}-x\_{n}}}}

**Численное интегрирование**

Интеграл находим по формуле Симпсона:



**Результат работы программы**



**Графики**

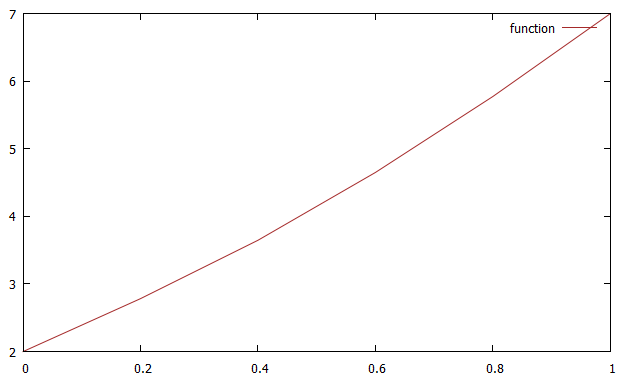


Рис. 1. График функции

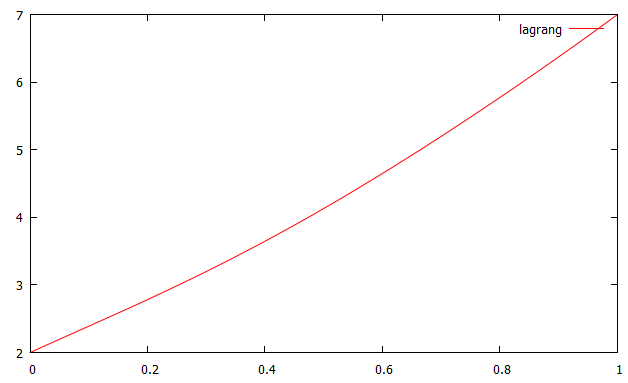


Рис. 2. График интерполяции

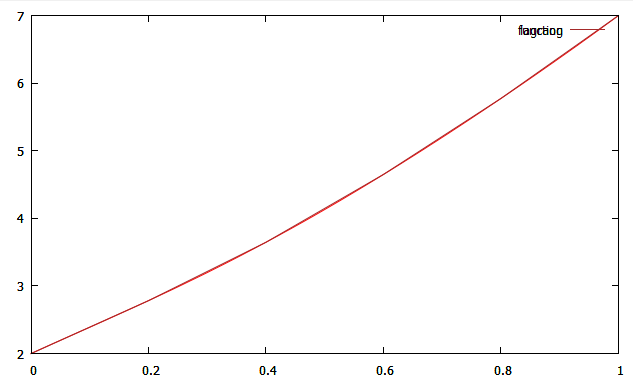


Рис.3 Графики функции и интерполяции

Интеграл

= 3.71178

**Заключение**

В рамках курсовой работы была решена краевая задача, результаты которой удовлетворяют заданным граничным условиям в концах интервала. Проведена интерполяция полиномом Лагранжа, построен график сеточной функции, который иллюстрирует решение дифференциального уравнения. По формуле Симпсона вычислено приближенное значение интеграла для заданной подынтегральной функции.

Листинг

#include <iostream>

#include <fstream>

#include <vector>

#include <cmath>

using namespace std;

doubleoutD1;*//перваяпроизводная*

doubleeps = 1e-6;//точность для МПД

double diff(double x, double y, double D1, double D2)*//Диф. урав*

{

returnpow(D2, 3) - 2 \* sin(x) \* D2 - cos(x) + 9 \* log(x) \* D1 + y/((x + 3)\*(x + 3));

}

*//нахождение значения второй производной*

doublefunc(doublex, doubley, doubleD1)

{

if (x == 0) {

x = 1e-10;

}

double a = 1, b = 0;

doublefa = 0, fb = 0;

do {

fa = diff(x, y, D1, a--);

fb = diff(x, y, D1, b++);

} while (fa \* fb> 0);

double c = 0;

while(fabs(b - a) >eps) {

c = (a + b) / 2;

if(diff(x, y, D1, a) \* diff(x, y, D1, c) < 0)

b = c;

else if (diff(x, y, D1, c) \* diff(x, y, D1, b) < 0)

a = c;

}

return (a + b) / 2;

}

*// Метод Рунге-Кутты II-го порядка с усреднением по времени*

doubleRunge(double a, double b, double h, double y, double D1)

{

double x = a;

doubley\_t = 0, D1\_t = 0;

while (x < b) {

y\_t = y + (h/2) \* D1;

D1\_t = D1 + (h/2) \* func(x, y, D1);

y += h \* D1\_t;

D1 += h \* func(x + h/2, y\_t, D1\_t);

x += h;

}

h = b - x;

y\_t = y + (h/2) \* D1;

D1\_t = D1 + (h/2) \* func(b, y, D1);

y += h \* D1\_t;

outD1 = D1 + h \* func(b + h/2, y\_t, D1\_t);

return y;

}

//*методстрельбы*

doubleshoot\_method(double x0, double x1, double y0, double y1, double h)

{

double al = 1, bt = 0;

doublefa = 0, fb = 0;

do {

fa = Runge(x0, x1, h, y0, al) - y1;

fb = Runge(x0, x1, h, y0, bt) - y1;

al -= h;

bt += h;

} while (fa \* fb> 0);

double g = 0;

while(fabs(bt - al) >eps) {

g = (al + bt) / 2;

if((Runge(x0, x1, h, y0, al) - y1) \* (Runge(x0, x1, h, y0, g) - y1) < 0) {

bt = g;

} else if ((Runge(x0, x1, h, y0, g) - y1) \* (Runge(x0, x1, h, y0, bt) - y1) < 0) {

al = g;

}

}

return (al + bt) / 2;

}

doublelagr(double Xl, int n, pair<double, double> \* v) {

int k, j;

double L=0, P;

for (k = 0; k < n; k++) {

P = 1;

for (j = 0; j < n; j++) {

if (k != j) {

P \*= ((Xl - v[j].first)/(v[k].first - v[j].first));

}

}

L += v[k].second \* P;

}

returnL;

}

//*формулаСимпсона(нахождениеинтеграла)*

doublesimpsons\_rule(vector <pair<double, double>> v, double h)

{

double sum = 0;

double a = v.front().first;

double b = v.back().first;

for(inti = 1; a + h \* i<= b - h; i++)

sum += (i % 2 ? 4 : 2) \* v[i].second;

return (v.front().second + sum + v.back().second) \* h / 3;

}

*//функция двойного пересчета*

double double\_counting\_runge(double a, double b, double h, double y0, double D1)

{

//DOUBLE COUNTING RUNGE

int norm = ((b-a)/h) + 1;

int exit = 1;

int k =0, k1=0, count = 0;

double h\_t = h;

int cnt\_et = 1;

while (exit) {

k = ((b-a)/h\_t) + 1;

printf("K=%d\n", k);

double per[k];

printf(" x\t\ty(x)\t\ty'(x)\n");

double b\_tmp = a;

for(int i = 0; b\_tmp <= b; i++, b\_tmp+=h\_t) {

per[i] = Runge(a, b\_tmp, h\_t, y0, D1);

printf("%d\t%lf\t%lf b = %lf\n", i, per[i], outD1, b\_tmp);

}

h\_t /=2;

k1 = ((b-a)/h\_t) + 1;

printf("K1=%d\n", k1);

double cer[k1];

printf(" x\t\ty(x)\t\ty'(x)\n");

b\_tmp = a;

printf("A=%lf B=%lf\n", a, b);

for(int i = 0; b\_tmp <= b; i++, b\_tmp+=h\_t) {

cer[i] = Runge(a, b\_tmp, h\_t, y0, D1);

printf("%d\t%lf\t%lf b = %lf\n", i, cer[i], outD1, b\_tmp);

}

int j = 0;

printf("COMPARE PAIRS\n");

for(int i = 0; i < k; i = i + cnt\_et) {

printf("%d|%d %lf - %lf = %lf > %lf\n", i, i\*2, per[i], cer[i\*2], fabs(per[i] - cer[i\*2]), eps);

count++;

if (fabs(per[i] - cer[i\*2]) > eps) {

printf("SHOOT\n");

cnt\_et = cnt\_et\*2;

count = 0;

break;

}

if (count == norm) {

exit=0;

break;

}

}

}

printf(" \n");

// printf("\nOUT\n");

k1 = ((b-a)/h\_t) + 1;

// printf("K1=%d\n", k1);

double cer[k1];

double ot[k1];

printf(" x\t\ty(x)\t\ty'(x)\n");

double b\_tmp = a;

for(int i = 0; b\_tmp < b; i++, b\_tmp+=h\_t) {

cer[i] = Runge(a, b\_tmp, h\_t, y0, D1);

ot[i] = outD1;

//printf("%d\t%lf\t%lf b = %lf\n", i, cer[i], outD1, b\_tmp);

}

int j = 0;

for(int i = 0; i < k1; i = i + cnt\_et\*2) {

Y[j] = cer[i];

OUTD[j] = ot[i];

// printf("%d\t%lf\t%lf \n", i, cer[i], ot[i]);

j++;

}

}

int main()

{

vector<pair<double, double>> v;

vector<pair<double, double>> v1;

*//входные данные*

doublea = 0, b = 1, h = 0.2, x0 = 0, y0 = 2, x1 = 1, y1 = 7;

for(double m = a; m <= b; m += h)//заполняемвектора

v.push\_back(make\_pair(m, 0));

double\_counting\_runge(a, b, h, y0,D1);

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

printf("%lf\t%lf\t%lf \n", v[i].first, Y[i], OUTD[i]);

}

for(size\_t i = 0; i < v.size(); i++)

v[i].second = Y[i];

ofstream in("in.csv", ios\_base::out);

for(size\_t i = 0; i < v.size(); i++)

in << v[i].first << ", " << Y[i] << "\n";

ofstream out("out.csv", ios\_base::out);

for(double m = a; m < b + 0.01; m += 0.01)

out << m << ", " << lagr(m, v.size(), v.data()) << "\n";

double x = x0, y = 0;

while(x <= x1) {

y = Runge(x0, x, h, y0, D1);

v1.push\_back(make\_pair(x, outD1));

x+=h;

}

double I = simpsons\_rule(v1, 0.2);

cout << "\n I = " << I << endl;

return 0;

}

**Литература**

1. Электронный конспект лекций по дисциплине «Вычислительная математика» преподаватель: Рубан А.А.
2. Wikipedia –[электронный ресурс]: https://ru.wikipedia.org
3. WolframMathworld. [электронный ресурс]: http://mathworld.wolfram.com/