Министерство образования и науки Российской Федерации

Севастопольский государственный университет

Кафедра ИС

Отчет

по лабораторной работе № 1

«Исследование погрешностей

приближенных вычислений»

Выполнил

ст. гр. И12д

Серегин А.В.

Проверил:

асс. Дрозин А. Ю.

Севастополь

2015

1.Цель работы

Выполнение лабораторной имеет целью формирования навыков практических расчетов при вычислениях на ЭВМ. В данной работе необходимо изучить правила округления приближенных чисел, методы расчета абсолютных и относительных погрешностей результатов вычислений, принципы построения вычислительных алгоритмов по критерию минимальной погрешности вычислений.

2. Варианты заданий

Вариант 6:

Задание 1: Вычислить ошибку, абсолютную и относительную погрешности. A = 4.26845000.

Задание 2: Вычислить относительную погрешность и число верных знаков в числе х. x= 0.2589637; Δx=6 \* 10­­-5.

Задание 3: Вычислить абсолютную погрешность и число верных знаков числа у. y= 58.237545; δy=1.4\*10-4.

Задание 4: Вычислить абсолютную и относительную погрешность фукнции U. U = (a-b)(c+d); a=85.12; b=48,4; c=0.5072; d=0,12.

3. Расчетные данные.

На рисунках ниже показаны результаты расчетных данных выполненных программой и граф вычислительного процесса.

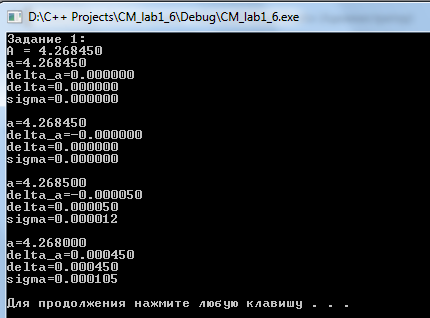


Рисунок 1 – Задание 1.

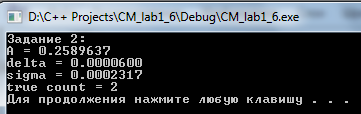


Рисунок 2 – Задание 2.

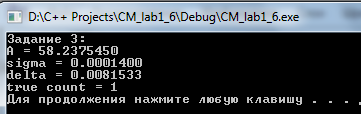


Рисунок 3 – Задание 3.

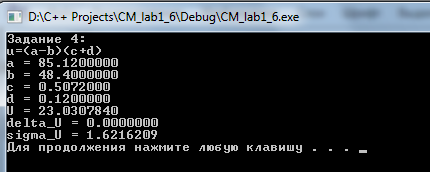


Рисунок 4 – Задание 4.

4.Код программы.

Main.cpp

#include <iostream>

#include <iomanip>

#include "CalcError.h"

using namespace std;

namespace CM

{

void Ex1()

{

system("cls");

double A = 4.26845000;

cout.setf(ios::fixed);

cout << "Задание 1:" << endl;

cout << "A = " << A << endl;

for (int i(6); i > 2; i--)

{

double a = CalcError::Round(A, i);

double d = CalcError::delta(A, a);

double da = CalcError::delta\_a(A, a);

double s = CalcError::sigma(A, d);

cout << "a=" << a << endl;

cout << "delta\_a=" << da << endl;

cout << "delta=" << d << endl;

cout << "sigma=" << s << endl;

cout << endl;

}

system("pause");

}

void Ex2()

{

system("cls");

double A = 0.2589637;

double d = 0.00006;

double s = CalcError::sigma(A, d);

cout.setf(ios::fixed);

cout.precision(7);

cout << "Задание 2:" << endl;

cout << "A = " << A << endl;

cout << "delta = " << d << endl;

cout << "sigma = " << s << endl;

cout << "true count = " << CalcError::TrueSimbolsNumber\_A\_delta(A, d) <<

endl;

system("pause");

}

void Ex3()

{

system("cls");

double A = 58.237545;

double s = 0.00014;

double d = CalcError::sigma\_to\_delta(A, s);

cout.setf(ios::fixed);

cout << "Задание 3:" << endl;

cout << "A = " << A << endl;

cout << "sigma = " << s << endl;

cout << "delta = " << d << endl;

cout << "true count = " << CalcError::TrueSimbolsNumber\_A\_sigma(A, s) <<

endl;

system("pause");

}

void Ex4()

{

system("cls");

double A = 85.12, a = CalcError::Round(A, 5);

double B = 48.4, b = CalcError::Round(B, 5);

double C = 0.5072, c = CalcError::Round(C, 5);

double D = 0.12, d = CalcError::Round(D, 5);

double U = (A - B)\*(C + D);

double d\_f1 = CalcError::delta\_difference(CalcError::delta\_a(A, a),

CalcError::delta\_a(B, b));

double d\_f2 = CalcError::delta\_difference(CalcError::delta\_a(C, c),

CalcError::delta\_a(D, d));

double dU = CalcError::delta\_multiplication(A - B, d\_f1, C + D, d\_f2);

double s\_f1 = CalcError::sigma\_summ(A, a, B, b);

double s\_f2 = CalcError::sigma\_difference(C, c, D, d);

double sU = CalcError::sigma\_multiplication(A-B, s\_f1, C+D, s\_f2);

cout.setf(ios::fixed);

cout << "Задание 4:" << endl;

cout << "u=(a-b)(c+d)" << endl;

cout << "a = " << A << endl;

cout << "b = " << B << endl;

cout << "c = " << C << endl;

cout << "d = " << D << endl;

cout << "U = " << U << endl;

cout << "delta\_U = " << dU << endl;

cout << "sigma\_U = " << sU << endl;

system("pause");

}

}

int main()

{

setlocale(LC\_ALL, "Russian");

CM::Ex1();

CM::Ex2();

CM::Ex3();

CM::Ex4();

return 0;

}

CalcError.h

#pragma once

namespace CM

{

class CalcError

{

public:

static double delta\_a(double, double);

static double delta(double);

static double delta(double, double);

static double sigma(double, double);

static double sigma\_to\_delta(double, double);

static int TrueSimbolsNumber\_A\_delta(double, double);

static int TrueSimbolsNumber\_A\_sigma(double, double);

static double Round(double, int);

static double delta\_summ(double, double);

static double delta\_difference(double, double);

static double delta\_multiplication(double, double, double, double);

static double delta\_division(double, double, double, double);

static double sigma\_summ(double, double, double, double);

static double sigma\_difference(double, double, double, double);

static double sigma\_multiplication(double, double, double, double);

static double sigma\_division(double, double, double, double);

};

}

CalcError.cpp

#include "CalcError.h"

#include <math.h>

#include <string>

namespace CM

{

double CalcError::Round(double a, const int n)

{

a \*= pow(10.0, n);

a = round(a+0.0000000001);

return a\*pow(10.0, -n);

}

double CalcError::delta(double A, double a)

{

return abs(A-a);

}

double CalcError::delta(double da)

{

return abs(da);

}

double CalcError::sigma\_to\_delta(double A, double s)

{

return s\*abs(A);

}

double CalcError::delta\_a(double A, double a)

{

return A - a;

}

double CalcError::sigma(double A, double d)

{

return (d/abs(A));

}

int CalcError::TrueSimbolsNumber\_A\_delta(double a, double d)

{

int true\_count = 0;

bool f = false;

std::string A = std::to\_string(a);

std::string t1 = std::to\_string(a+d);

std::string t2 = std::to\_string(a-d);

for (int i(0); i < A.length(); i++)

{

if (A[i] == t1[i] && A[i] == t2[i] && f) true\_count++;

if ((A[i] != t1[i] || A[i] != t2[i]) && f) break;

if (A[i] == '.') f=true;

}

return true\_count;

}

int CalcError::TrueSimbolsNumber\_A\_sigma(double a, double s)

{

double d = s\*abs(a);

return TrueSimbolsNumber\_A\_delta(a, d);

}

double CalcError::delta\_summ(double da, double db)

{

return da + db;

}

double CalcError::delta\_difference(double da, double db)

{

return da - db;

}

double CalcError::delta\_multiplication(double a, double da, double b, double db)

{

return b\*da + a\*db;

}

double CalcError::delta\_division(double a, double da, double b, double db)

{

return da / b - a\*db / (b\*b);

}

double CalcError::sigma\_summ(double x, double dx, double y, double dy)

{

return x\*dx / (x + y) / x + y\*dy / (x + y) / y;

}

double CalcError::sigma\_difference(double x, double dx, double y, double dy)

{

return x\*dx / (x - y) / x - y\*dy / (x - y) / y;

}

double CalcError::sigma\_multiplication(double x, double dx, double y, double dy)

{

return dx / x + dy / y;

}

double CalcError::sigma\_division(double x, double dx, double y, double dy)

{

return dx / x - dy / y;

}

}

Выводы

В ходе лабораторной работы мы научились правильно округлять приближенные числа, находить количество верных знаков после запятой, абсолютную и относительные погрешности вычислений.