**Федеральное государственное бюджетное образовательное учреждение**

**высшего профессионального образования**

**«Санкт-Петербургский государственный политехнический университет»**

—

**Институт информационных технологий и управления**

**Кафедра «Распределенные вычисления и компьютерные сети»**

**Отчет по лабораторной работе №1**

**На тему: «Интерполяция и квадратурные формулы».   
Вариант 23.**

**по дисциплине «Вычислительная математика»**

Выполнил

студент гр. 23507/1 В.Б. Борисов

Руководитель

доц. Т.В. Леонтьева   
  
  
 ≪\_\_\_≫ \_\_\_\_\_\_\_\_\_\_ 201\_ г.

Санкт-Петербург

2015

Постановка задачи

Для c h = 0,25 вычислить значения , используя для вычисления интеграла программу **QUANC8**. По полученным точкам построить сплайн-функцию и полином Лагранжа 8-й степени. Сравнить значения обеих аппроксимаций в точках (k= 0,….7).

Описание решения

Руководствуясь условию задания, мы вначале используем подпрограмму QUANC8 для того чтобы вычислить значения функции. Далее используем сплайн-функцию, которая была построена с помощью стандартной программы SPLINE, а затем считаем значения её в точке с помощью стандартной программы SEVAL, параллельно с этим вычисляем полином Лагранжа 8-й степени, который был написан собственноручно.

При выполнении данной работы использовался язык C++, ОС Windows 8.1. Также было проведено тестирование работы программ.

Вывод результатов работы программы:

Результаты тестирования программы были выведены на экран (Результаты работы программы смотри ниже).

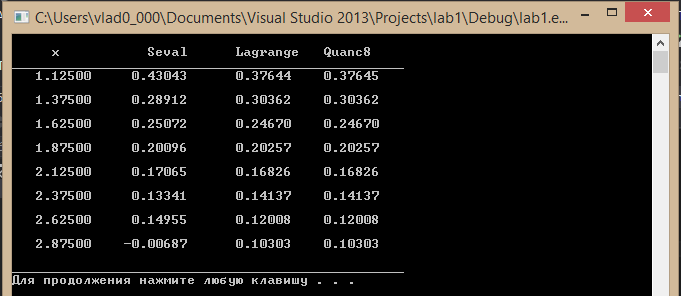


Рис.1

Анализ полученных результатов   
  
 Для заданной функции было протестировано следующее программное обеспечение: QUANC8, SPLINE, SEVAL, а также подпрограмма для построения полинома Лагранжа.  
  
 Результаты работы программы показали, что наиболее точные значения получены при помощи использования полинома Лагранжа. SPLINE-функция имеет большие погрешности на краях

промежутка. Следовательно, для небольшого числа точка следует использовать полином Лагранжа, когда как для большего их числа более подходящим окажется SPLINE.

Исходный текст программы  
  
**FILES CPP:**  
  
1) main.cpp

#include <math.h>

#include "cmath.h"

#include <iostream>

#define ndim 9

#define size 8

double x;

double Quanc8[ndim] = { 0 };

double Xk[ndim] = { 1 };

double ResultLagrange[size] = { 0 };

double Seval[ndim];

// Полином Лагранжа

double Lagrange(double x){

double Qm = 0, wkx[ndim] = {1, 1, 1, 1, 1, 1, 1, 1, 1}, wkk[ndim] = { 1, 1, 1, 1, 1, 1, 1, 1, 1 };

int i, j;

for (i = 0; i<ndim; i++)

for (j = 0; j<ndim; j++)

if (i != j)

wkx[i] \*= (x - Xk[j]);

for (i = 0; i<ndim; i++)

for (j = 0; j<ndim; j++)

if (i != j)

wkk[i] \*= Xk[i] - Xk[j];

for (i = 0; i<ndim; i++)

if (wkk[i] != 0)

Qm += (wkx[i] / wkk[i])\*Quanc8[i];

return Qm;

}

// Результат Лагранжа 8-й степени для xk = 1,125 + 0,25k.

void ResultLagrangeInit(){

x = 1.125;

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

ResultLagrange[i] = Lagrange(x + i\*0.25);

}

}

// Вычисление интеграла с помощью QUANC8 с шагом 0,25

void QUANC8(double xx, int k)

{

double f(double z);

double a, b, abserr, relerr, result, errest, posn;

int nfe, flag, i;

x = xx;

a = 0.0;

b = 1;

relerr = 1.0e-10;

abserr = 0.0;

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

quanc8(f, a, b, abserr, relerr, &result, &errest,

&nfe, &posn, &flag);

Quanc8[i] = result;

Xk[i] = x;

x = x + 0.25;

}

}

// Функция

double f(double z)

{

return 1 / (exp(pow(z, 2)) + pow(x, 2));

}

// Функция Spline для xk = 1,125 + 0,25k вычисления с шагом 0,25

void Spline(){

double b[ndim], c[ndim], d[ndim];

int n, i, last, flag;

n = 9;

spline(n, 1, 1, 1.0, exp(1.0), Xk, Quanc8, b, c, d, &flag);

if (flag == 0)

{

last = 0;

x = 1.125;

for (i = 0; i<size; i++){

Seval[i] = seval(n, x + i\*0.25, Xk, Quanc8, b, c, d, &last);

}

}

}

int main(){

QUANC8(1, 9);

ResultLagrangeInit();

Spline();

QUANC8(1.125, 8);

x = 1.125;

printf("\n x Seval Lagrange Quanc8\n");

printf("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

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

printf("%10.5f %10.5f %10.5f %10.5f\n\n", x + i\*0.25, Seval[i], ResultLagrange[i], Quanc8[i]);

printf("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

system("pause");

}

QUANC8.cpp  
  
SPLINEnSEVAL.cpp

**HEADER FILE:**

Cmath.h