Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное

образовательное учреждение высшего образования

Санкт-Петербургский университет аэрокосмического приборостроения

ГУАП

КАФЕДРА № 2

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Кандидат тех. Наук, доцент |  |  |  | С.Л Козенко |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ПРАКТИЧЕСКОЙ РАБОТЕ №3  **“Интерполяция”** |
| по дисциплине: Вычислительная математика |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. | 4134к |  |  |  | Костяков НА |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург

2023

Цели работы: а) освоение методов интерполяции функций; б) совершенствование навыков по алгоритмизации и программированию вычислительных задач

# Постановка задачи:

Составить схему алгоритма и программу на языке C/C++ решения задачи по теме «Интерполяция» в соответствии с индивидуальным заданием (варианты заданий приведены ниже в таблице).



# Математическая часть

# Интерполяция. Введение. Общая постановка задачи

При решении различных практических задач результаты исследований оформляются в виде таблиц, отображающих зависимость одной или нескольких измеряемых величин от одного определяющего параметра (аргумента). Такого рода таблицы представлены обычно в виде двух или более строк (столбцов) и используются для формирования математических моделей.

Таблично заданные в математических моделях функции обычно записываются в таблицы вида:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***Х*** | ***Х0*** | ***Х1*** | ***…*** | ***Хn*** |  |
| ***Y1(X)*** | ***Y(Х0)*** | ***Y(Х1)*** | ***…*** | ***Y(Хn)*** |  |
| ***…*** | ***…*** | ***…*** | ***…*** | ***…*** | **( 1 )** |
| ***Ym(X)*** | ***Y(Х0)*** | ***Y(Х1)*** | ***…*** | ***Y(Хn)*** |  |

Ограниченность информации, представленной такими таблицами, в ряде случаев требует получить значения функций ***Yj(X) (j=1,2,…,m)*** в точках ***Х***, не совпадающих с узловыми точками таблицы ***Хi(i=0,1,2,…,n)***. В таких случаях необходимо определить некоторое аналитическое выражение ***φj(Х)*** для вычисления приближенных значений исследуемой функции ***Yj(X)*** в произвольно задаваемых точках ***Х***. Функция ***φj(Х)*** используемая для определения приближенных значений функции ***Yj(X)*** называется аппроксимирующей функцией (от латинского ***approximo*** - приближаюсь). Близость аппроксимирующей функции ***φj(Х)*** к аппроксимируемой функции ***Yj(X)*** обеспечивается выбором соответствующего алгоритма аппроксимации.

Все дальнейшие рассмотрения и выводы мы будем делать для таблиц, содержащих исходные данные одной исследуемой функции (т. е. для таблиц с ***m=1***). Пар1.1

## Постановка задачи интерполяции

Наиболее часто для определения функции ***φ(Х)*** используется постановка, называемая постановкой задачи интерполяции.

В этой классической постановке задачи интерполяции требуется определить приближенную аналитическую функцию ***φ(Х)***, значения которой в узловых точках ***Хi*** совпадают со значениями ***Y(Хi)***исходной таблицы, т.е. условий



Построенная таким образом аппроксимирующая функция ***φ(Х)*** позволяет получить достаточно близкое приближение к интерполируемой функции ***Y(X)*** в пределах интервала значений аргумента [***Х0; Хn***], определяемого таблицей. При задании значений аргумента ***Х***, не принадлежащих этому интервалу, задача интерполяции преобразуется в задачу *экстраполяции.* В этих случаях точность значений, получаемых при вычислении значений функции ***φ(Х),*** зависит от расстояния значения аргумента ***Х*** от ***Х0***, если ***Х*** < ***Х0*** *,* или от ***Хn*** , если ***Х*** > ***Хn****.*

При математическом моделировании интерполирующая функция может быть использована для вычисления приближенных значений исследуемой функции в промежуточных точках подынтервалов [***Хi****;* ***Хi+1***]. Такая процедура называется *уплотнением таблицы*.

Алгоритм интерполяции определяется способом вычисления значений функции ***φ(Х).*** Наиболее простым и очевидным вариантом реализации интерполирующей функции является замена исследуемой функции ***Y(Х)*** на интервале [***Хi; Хi+1***] отрезком прямой, соединяющим точки ***Yi , Yi+1***. Этот метод называется методом линейной интерполяции.

# Интерполяционная формула Лагранжа

Предложенный Лагранжем алгоритм построения интерполирующих функций по таблицам (1) предусматривает построение интерполяционного многочлена ***Ln(x)*** в виде

где ***li(x)*** - многочлен степени ***n***, для которого выполняются условия

Очевидно, что выполнение для (10) условий (11) определяет выполнение условий (2) постановки задачи интерполяции.

Многочлены ***li(x)*** записываются следующим образом

***li(x)= qi(x-x0)(x-x1)·…·(x-xi-1)(x-xi+1) … (x-xn).***

Здесь ***q***j - константа, значение которой определяется с учётом (12) как

Отметим, что ни один множитель в знаменателе формулы (14) не равен нулю. Вычислив значения констант ***сi***, можно использовать их для вычисления значений интерполируемой функции в заданных точках.

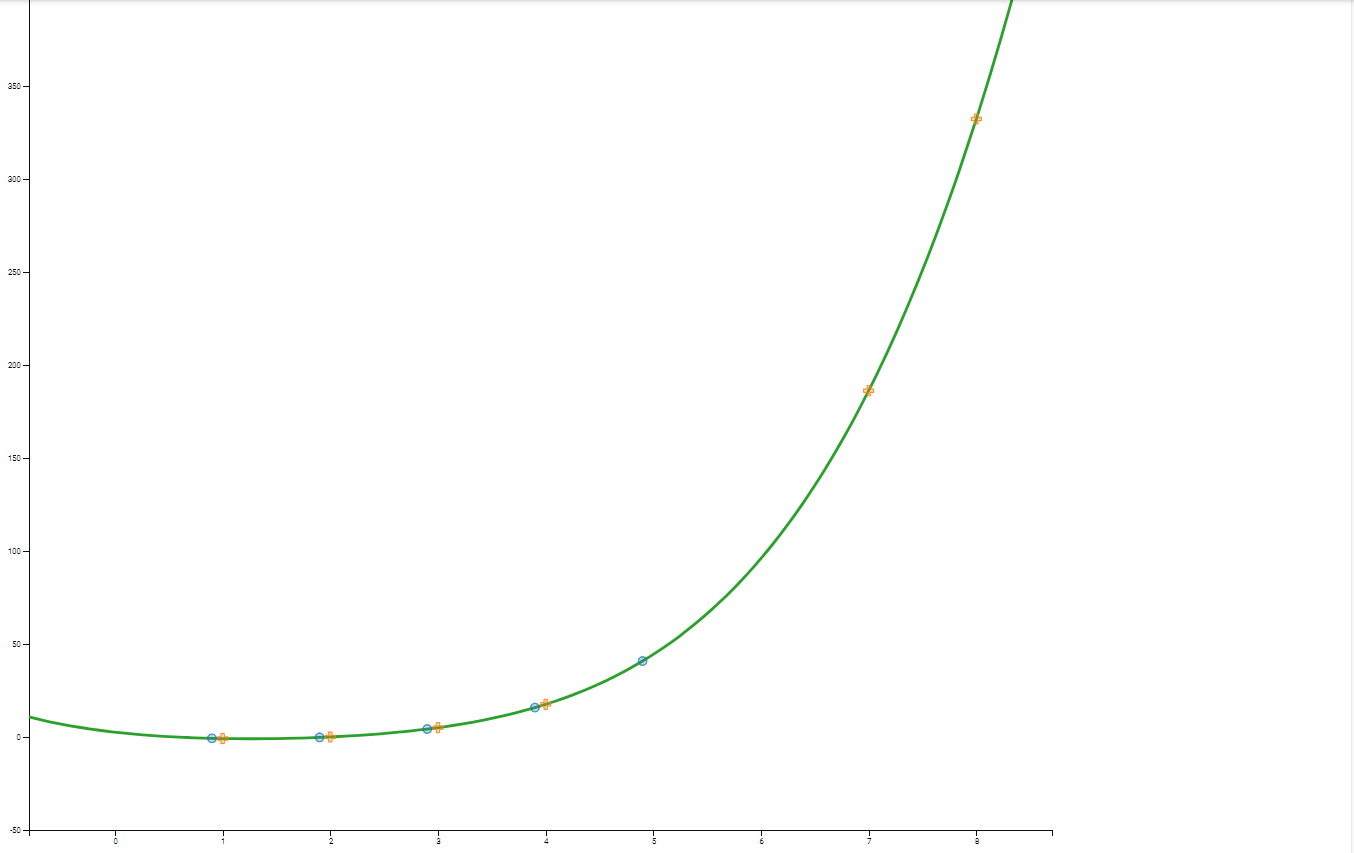
Формула интерполяционного многочлена Лагранжа (11) с учётом формул (13) и (14) может быть записана в виде

# Аналитические расчеты

Для вычисления я воспользовался калькулятором <https://planetcalc.ru/8692/> и получил следующие результаты для точек 0,3 1,4 2,4 3,4 4,6 6



**И график**

****

**Также в excel**

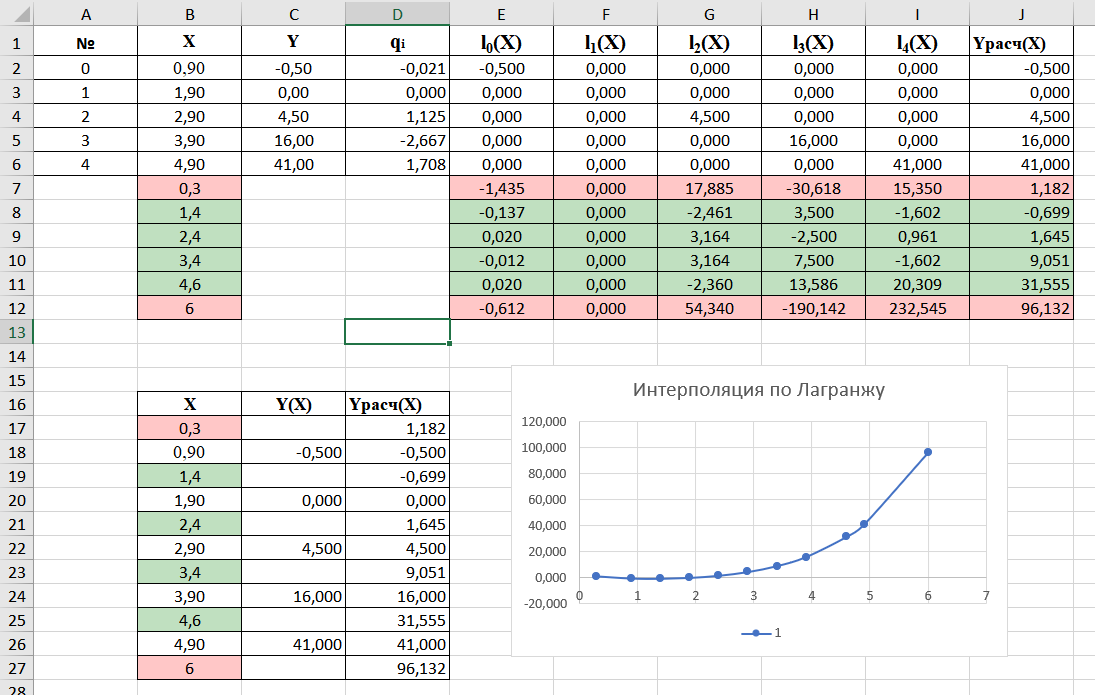
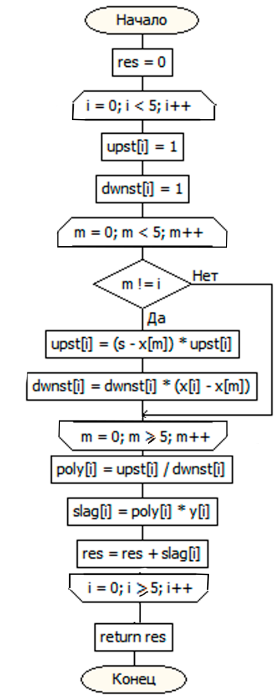
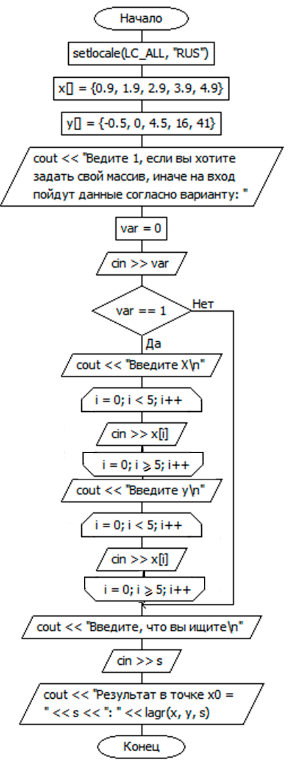
****

Схема программы **double lagr()**

****

**Int main()**

****

# Листинг программы

#include <iostream>

using namespace std;

double lagr(double x[5], double y[5], double s) {

double res = 0;

double upst[5];

double dwnst[5];

double poly[5];

double slag[5];

for (int i = 0; i < 5; i++) {

upst[i] = 1;

dwnst[i] = 1;

for (int m = 0; m < 5; m++) {

if (m!=i)

{

upst[i] = (s - x[m]) \* upst[i];

dwnst[i] = dwnst[i] \* (x[i] - x[m]);

}

}

poly[i] = upst[i] / dwnst[i];

slag[i] = poly[i] \* y[i];

res = res + slag[i];

}

return res;

}

int main()

{

setlocale(LC\_ALL, "RUS");

double x[] = { 0.9, 1.9, 2.9, 3.9, 4.9 };

double y[] = { -0.5, 0, 4.5, 16, 41};

cout << "Ведите 1, если вы хотите задать свой массив, иначе на вход пойдут данные согласно варианту: ";

int var = 0;

cin >> var;

if (var==1) {

cout << "Введите X\n";

for (int i = 0; i < 5; i++) {

cin >> x[i];

}

cout << "Введите y\n";

for (int i = 0; i < 5; i++) {

cin >> x[i];

}

}

cout << "Введите, что вы ищите\n";

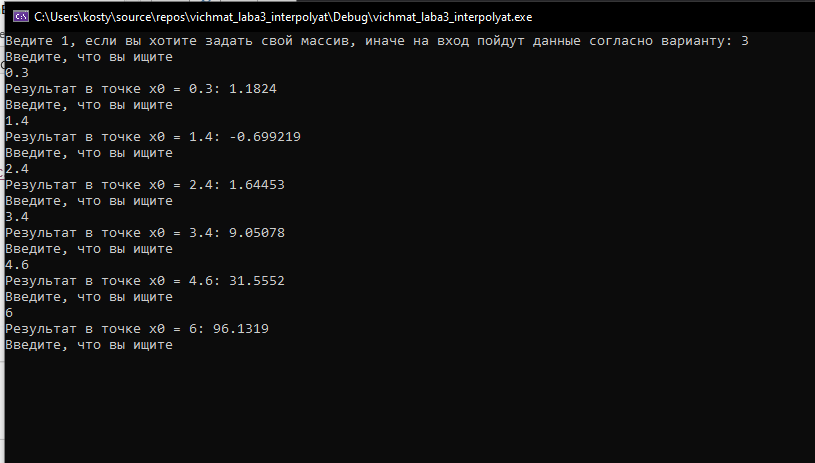
double s;

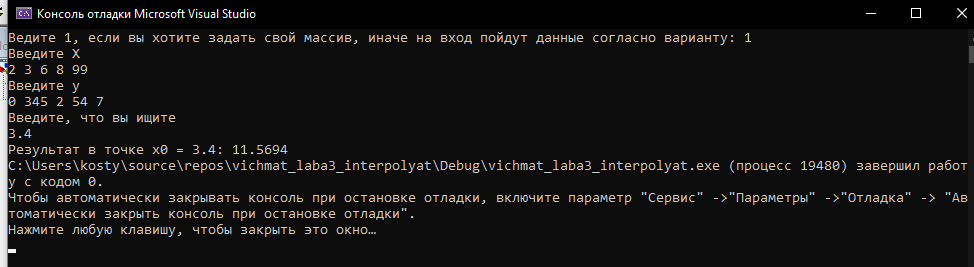
cin >> s;

cout << "Результат в точке x0 = " << s << ": " << lagr(x, y, s);

}

# Результаты работы



****

# Вывод

Я освоил навыки программирования интерполяционных функций