**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра математического обеспечения и применения ЭВМ**

отчет

**по практическому заданию №8**

**по дисциплине «Вычислительная математика»**

Тема: **Интерполяционные формулы для неравноотстоящих узлов**

|  |  |  |
| --- | --- | --- |
| Студент гр. 6304 |  | Виноградов К.А. |
| Преподаватель |  | Лисс А.Р. |

Санкт-Петербург

2017

**Цель работы.**

Изучение интерполирования функции с помощью интерполяционных формул для неравноотстоящих узлов, разработка программы на одном из языков программирования, вычисляющей значение в точке x функции, заданной таблицей, используя интерполяционную схему Эйткена либо интерполяционную формулу Ньютона.

**Основные теоретические положения.**

Пусть известны значения некоторой функции  в n+1 различных точках , которые обозначим следующим образом: .

Указанные значения могут быть получены путем экспериментальных измерений или найдены с помощью достаточно сложных вычислений. В задаче интерполяции функции , как было сказано ранее, решается проблема приближенного восстановления значения функции в произвольной точке x. Для этого строится алгебраический многочлен  степени n, который в точках  принимает заданные значения, т. е.

. (1.4)

Следует заметить, что если точка x расположена вне минимального отрезка, содержащего все узлы интерполяции , то замену функции  на  также называют экстраполяцией.

В общем случае доказано, что существует единственный интерполяционный многочлен n-й степени, удовлетворяющий условиям (1.4),

, (1.5)

где

. (1.6)

Интерполяционный многочлен, представленный в виде (1.5), называется интерполяционным многочленом Лагранжа, а функции (1.6) - лагранжевыми коэффициентами [1]-[4].

Для оценки погрешности интерполяции (в частности, и экстраполяции) в текущей точке  ( - отрезок, содержащий все узлы интерполяции и точку) можно использовать соотношение

, (1.7) где ;- (n+1)-я производная интерполируемой функции в некоторой точке ; .

Оценить максимальную погрешность интерполяции на всем отрезке  можно с помощью соотношения

. (1.8)

Использование оценок погрешностей (1.7) и (1.8) предполагает ограниченность (n+1)-й производной интерполируемой функции на отрезке , т. е. .

На практике вместо общей формы записи (1.5) часто используются другие формы записи интерполяционного многочлена, более удобные для применения в конкретных ситуациях [5], [10], [12].

Интерполяционный многочлен Ньютона для неравноотстоящих узлов интерполяции имеет вид

…

…, (1.9) где  - разделенная разность k-го порядка.

Вычисление разделенных разностей производится по соотношениям

,

...................................................

.

При использовании интерполяционного многочлена Ньютона (1.9) изменение степени n требует только добавить или отбросить соответствующее число стандартных слагаемых, что удобно на практике. В то же время, непосредственное использование интерполяционного многочлена Лагранжа (1.5) требует строить его заново при изменении n.

В том случае, если требуется найти лишь численное значение интерполяционного многочлена , а не его представление, может быть использована итерационно-интерполяционная схема Эйткена [6], [12].

Пусть  - интерполяционный многочлен, определяемый парами , , , ... так, что .

Интерполяционные многочлены возрастающих степеней получают последовательно следующим образом:

,

,

...…..............................................

,

......................................................

.

......................................................

Этот процесс можно закончить, когда у значений двух интерполяционных многочленов последовательных степеней совпадает требуемое количество знаков.

**Постановка задачи.**

Вычислить значение функции, заданной табл. 1, в точке x = 4.8392, используя схему Эйткена, с помощью ЭВМ.

Таблица 1 – Список узлов интерполирования

|  |  |
| --- | --- |
| X[0] = 0.3152 | Y[0] = -3.0976 |
| X[1] = 0.9536 | Y[1] = -0.0994 |
| X[2] = 1.1080 | Y[2] = 0.1823 |
| X[3] = 2.4728 | Y[3] = -0.3671 |
| X[4] = 4.1792 | Y[4] = 8.1696 |
| X[5] = 4.5616 | Y[5] = 14.2471 |
| X[6] = 4.6304 | Y[6] = 15.5694 |
| X[7] = 4.6704 | Y[7] = 16.3723 |
| X[8] = 5.2480 | Y[8] = 31.0168 |
| X[9] = 5.5680 | Y[9] = 41.8549 |
| X[10] = 6.3568 | Y[10] = 78.3427 |

**Выполнение работы.**

Cоставим программу-функцию для вычисления неизвестной функции по узлам интерполяции double Aitken(double\* x, double\* y, int n, int m).

Составим головную программу, содержащую обращение к файлу с входными данными input.txt, обращение к вычислительным процедурам и осуществляющую печать результатов в файл output.txt.

Все программы представлены в приложении А.

Расставим на координатной оси все точки, соответствующие таблице, и соединим их. Получим ломанную – эскиз интерполируемой функции. График представлен на рис. 1.

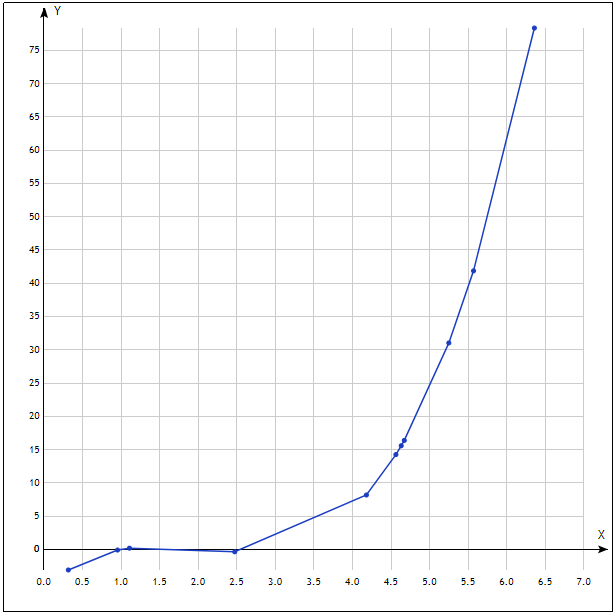


Рисунок 1 – Предварительный график функции

Проведем вычисления функции с помощью программы по точкам x от 0 до 7 с шагом 0.05, а также в заданной точке. График представлен на рис. 2.

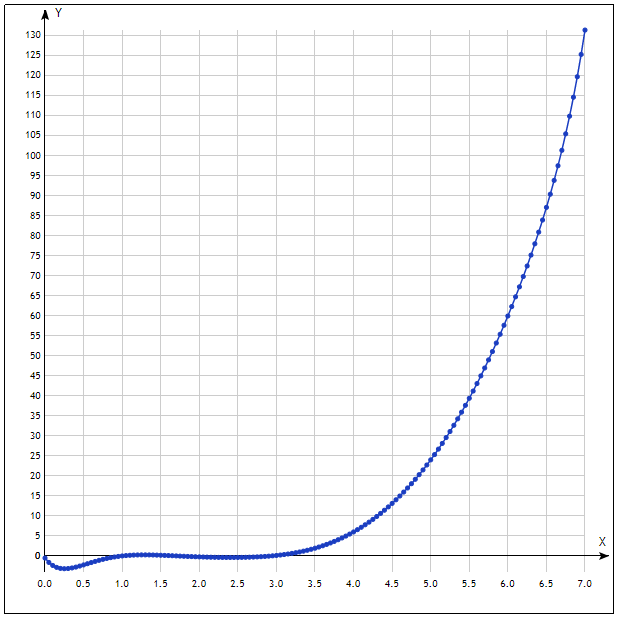


Рисунок 2 – График интерполированной функции

Значение y в заданной точке x равняется 20.0468.

**Выводы.**

В результате выполнения работы была написана программа, вычисляющая значения неизвестной функции, если известны только узлы интерполяции. В условиях данной задачи сложно оценить точность вычисления значения функции, так как вид функции неизвестен. Аналитико-графическая оценка позволяет сделать вывод о правильности полученных данных.

Приложение А

Код программы

#include <stdlib.h>

#include <stdio.h>

#include <math.h>

#include <fstream>

#define ELEMS 11

using namespace std;

double Aitken(double\* x, double\* y, int n, int m, double point);

int main()

{

double point = 0;

double point2=4.8392;

FILE\* input = fopen("input.txt", "r");

double x[ELEMS];

double y[ELEMS];

double result, result2;

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

fscanf(input, "%lf %lf", &x[i], &y[i]);

fclose(input);

ofstream output("output.txt");

while(point<=7)

{

for (int i = 1; i < ELEMS; i++) {

for (int j = 0; j < ELEMS - i; j++)

result = Aitken(x, y, i, j + i, point);

}

output<<"("<<point<<";"<<result<<")";

point+=0.05;

}

for (int i = 1; i < ELEMS; i++) {

for (int j = 0; j < ELEMS - i; j++)

result2 = Aitken(x, y, i, j + i, point2);

}

output<<"\nResult is y = "<<result2<<" when x = "<<point2;

printf("Result is y = %.4lf when x = %lf\n", result2, point2);

return 0;

}

double Aitken(double\* x, double\* y, int n, int m, double point){

double s;

if (n == 0)

s = y[m];

else

s = ((point - x[m - n]) \* Aitken(x, y, n - 1, m, point) - (point - x[m]) \* Aitken(x, y, n - 1, m - 1, point)) / (x[m] - x[m - n]);

return s;

}