**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра математического обеспечения и применения ЭВМ**

отчет

**по практическому заданию №9**

**по дисциплине «Вычислительная математика»**

Тема: **Интерполяционные формулы для равноотстоящих узлов**

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

Санкт-Петербург

2017

**Цель работы.**

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

**Основные теоретические положения.**

Если значения функции  заданы в точках   с постоянным положительным шагом, то часто используется интерполяционный многочлен Ньютона для интерполяции вперед

, (1.10)

где , а конечные разности , носящие названия нисходящих разностей, находят из соотношений

,

.

Интерполяционный многочлен (1.10) удобно использовать при работе в начале таблицы значений функции и для экстраполяции левее точки .

Интерполяционный многочлен с узлами  где , имеет вид

 (1.11)

и называется интерполяционным многочленом Ньютона для интерполяции назад. Его удобно использовать при интерполяции в конце таблицы и для экстраполяции правее точки . Входящие в выражение (1.11) значения конечных восходящих разностей находят из соотношений

,

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

.

Если при заданном  в таблице значений функции  с шагом  имеется достаточное число узлов с каждой стороны от , то целесообразно узлы интерполяции  выбрать так, чтобы точка  оказалась как можно ближе к середине минимального отрезка, содержащего узлы. При этом обычно в качестве  берется ближайший к  узел, затем за  принимается ближайший к  узел, расположенный с противоположной от  стороны, чем . Следующие узлы назначаются поочередно с разных сторон от  и должны быть расположены как можно ближе к . Одной из возможных схем интерполяции в этом случае является схема Стирлинга с интерполяционным многочленом вида



В этом выражении учитывается, что дано нечетное число  значений функции,  где . Обычно эту формулу целесообразно использовать при .

**Постановка задачи.**

Вычислить значение функции, заданной табл. 1, в точках x1 = 0.8110, x2 = 0.3770 и x3 = 1.5480, используя интерполяционную формулу Стирлинга.

Таблица 1 – Список узлов интерполирования

|  |  |
| --- | --- |
| X[ 0] = 0.2680 | Y[ 0] = -0.1950 |
| X[ 1] = 0.4290 | Y[ 1] = -0.0480 |
| X[ 2] = 0.5910 | Y[ 2] = 0.0110 |
| X[ 3] = 0.7520 | Y[ 3] = 0.0100 |
| X[ 4] = 0.9130 | Y[ 4] = -0.0280 |
| X[ 5] = 1.0740 | Y[ 5] = -0.0770 |
| X[ 6] = 1.2360 | Y[ 6] = -0.1130 |
| X[ 7] = 1.3970 | Y[ 7] = -0.1090 |
| X[ 8] = 1.5580 | Y[ 8] = -0.0410 |
| X[ 9] = 1.7190 | Y[ 9] = 0.1140 |
| X[10] = 1.8810 | Y[10] = 0.3870 |

**Выполнение работы.**

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

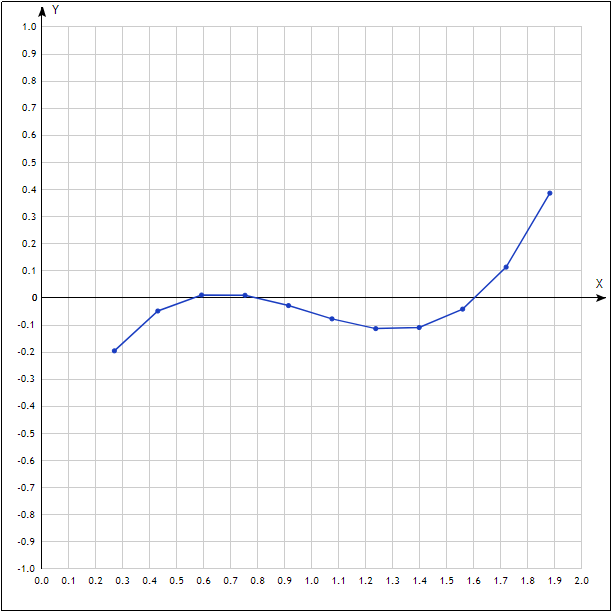


Рисунок 1 – Предварительный график функции

Cоставим подпрограммы-функции к которым будет обращаться функция интерполирования double delta(double\* y, int deg, int low\_index, int index) и int Fact(int n).

Cоставим программу-функцию для вычисления неизвестной функции по узлам интерполяции double Stirling(double\* y,int index, double q).

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.

Протестируем программу и запишем полученные результаты в табл. 2.

Таблица 2 – Результаты работы программы

|  |  |  |  |
| --- | --- | --- | --- |
| Точка | x | q | Результат |
| 1 | 0.811 | 0.364198 | 0.0151118 |
| 2 | 0.377 | 0.320988 | -0.175501 |
| 3 | 1.548 | 0.0617284 | -0.0455152 |

**Выводы.**

В результате выполнения работы была написана программа, позволяющая вычислять значения неизвестной функции, если известен ряд ее значений в равноотстоящих узлах интерполяции.

В условиях данной задачи нельзя оценить точность вычисления значений функции в какой бы то ни было точке, так как её вид неизвестен, но аналитико-графическая оценка позволяет сделать вывод о правильности полученных данных.

Приложение А

Код программы

#include <math.h>

#include <stdio.h>

#include <fstream>

#define ELEMS 11

using namespace std;

double Stirling(double\* y, int d, double q);

double delta(double\* y, int r, int k, int delta);

int main() {

FILE\* input = fopen("C:/Users/zo\_Om/Documents/codeblocks/vm\_9/input.txt", "r");

double x[ELEMS];

double y[ELEMS];

double points[3] = {0.8110, 0.3770, 1.5480};

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

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

fclose(input);

ofstream output("C:/Users/zo\_Om/Documents/codeblocks/vm\_9/output.txt");

double minInterval, q, point, result;

int index;

printf("Using the Stirling interpolation formula.\n\n");

for (int j = 0; j < 3; j++) {

minInterval = 10.0;

point = points[j];

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

if (fabs(x[i] - point) < minInterval) {

index = i;

minInterval = fabs(x[i] - point);

}

q = fabs(point - x[index]) / (x[2] - x[1]);

result=Stirling(y, index, q);

output<<"x = "<<point<<endl;

output<<"q = "<<q<<" y0 = "<<y[index]<<endl;

output<<"Result is "<<result<<"\n"<<endl;

}

printf("Results are in the file\n");

}

int Fact(int n) {

if (n <= 1) return 1;

else return n \* Fact(n - 1);

}

double delta(double\* y, int deg, int low\_index, int index) {

if (deg == 1)

return y[index+low\_index+1] - y[index+low\_index];

else

return delta(y, deg-1, low\_index+1, index) - delta(y, deg-1, low\_index, index);

}

double Stirling(double\* y,int index, double q) {

double q\_coef;

double result = y[index];

for (int i = 1; i < 6; i++){

if (i==1) q\_coef = q;

else q\_coef \*= pow(q,2) - pow(i-1, 2);

result -= q\_coef/Fact(2\*i-1) \* (delta(y, 2\*i-1, -i, index) + delta(y, 2\*i-1, -(i-1), index))/2 + q\*q\_coef\*delta(y, 2\*i, -i, index)/Fact(2\*i);

}

return result;

}