**Министерство науки и высшего образования РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра ИС**

отчет

**по лабораторной работе №2**

**по дисциплине «Конструирование программ»**

Тема: Интерполирование и экстраполирование данных. Интерполяционный многочлен Лагранжа

|  |  |  |
| --- | --- | --- |
| Студент гр. 8363 |  | Нерсисян А.С. |
| Преподаватель |  | Копыльцов А.В. |

Санкт-Петербург

2020

**Цель работы.**

Заработать навыки использования интерполяционных многочленов не-высоких степеней, восстанавливаемый без дополнительных условий многочлен Лагранжа имеющий степень, на единицу меньшую числа точек интерполяционной таблицы. По найденному многочлену найти приближенные значения функции для любых значений аргумента, лежащих между узлами заданной сетки.

**Основные теоретические положения.**

* 1. **Задача приближения функций**

Вычисление значений функции  - задача, с которой постоянно приходиться сталкиваться на практике. Часто бывает, что вычисление  затруднительно, например:

1) функция  задана таблично , а вычисление необходимо проводить в точках , не совпадающих с табличными;

2) вычисление функции  дорого;

3) для вычисления  необходим эксперимент.

В таких условиях целесообразно заменить  некоторой близкой к ней функцией , которая вычисляется быстро и надежно, а погрешность приближения  достаточно мала. При этом полезно при выборе функции  использовать любую дополнительную информацию о функции , о ее гладкости, четности, периодичности, монотонности и так далее. Это дает возможность осознанно выбрать класс  аппроксимирующих функций.

Широко используются функции вида , представляющие собой линейные комбинации некоторых базисных функций . Функция  называется обобщенным многочленом степени .

* 1. **Интерполяция обобщенными многочленами**

Если ставится требование совпадения функции  с функцией  в некоторых фиксированных точках, то это приводит к задаче интерполяции.

Построить функцию , удовлетворяющую условиям , - узлы интерполяции. Очевидно, что выбор  неоднозначен, так как по заданной таблице можно построить бесконечно много интерполирующих функций. Рассмотрим обобщенный многочлен , удовлетворяющий условию  Эта формула, представленная в виде , очевидно, эквивалентна следующей системе линейных алгебраических уравнений:



Для определения  необходимо решить систем относительно . На практике это делается чрезвычайно редко. Как правило, система плохо обусловлена. В большинстве приложений используются специальные явные формулы для записи  и вычисление  не нужно.

* 1. **Полиномиальная интерполяция. Многочлен Лагранжа**

Если в качестве базисной взять систему степенных функций, то есть , то получаем задачу полиномиальной интерполяции:



Теорема 2.1. Существует единственный интерполяционный многочлен степени , удовлетворяющий условиям.

В качестве искомого многочлена возьмем многочлен степени  вида



Таким образом, система функций, по которой строится интерполяционный многочлен, есть



Для нахождения  надо найти набор коэффициентов  . Не будем составлять и решать систему линейных уравнений вида      …  , найдем коэффициенты иным способом.

Пусть , с учетом  получим



Аналогично, полагая  и учитывая, что  будем иметь



Если , то  Тогда сам многочлен  будет иметь вид



Этаформула называется интерполяционной формулой Лагранжа. Приведем ее в сокращенной записи:



Очевидно,  представляет собой многочлен степени , удовлетворяющий условию



Таким образом, степень многочлена  равна , при  в формуле (2.3.4) обращаются в нуль все слагаемые, кроме слагаемого с номером , равного .

Выпишем отдельно многочлены Лагранжа первой и второй степени, ибо именно они чаще всего используются на практике.



* 1. **Погрешность интерполяции**

*Теорема 2.2.* Пусть функция  дифференцируема  раз на отрезке , содержащем узлы интерполяции  Тогда для погрешности интерполяции в точке  справедливо равенство  в котором 

Последнюю формулу несколько модернизируют. Так как положение   
точки  неизвестно, то  заменяют на  Тогда



* 1. **Конечные разности и их свойства**

Пусть функция  задана таблично  - шаг таблицы,  - узлы таблицы.

Величина  называется конечной разностью первого порядка функции  в точке  с шагом .

Конечная разность порядка  функции  в точке  есть  Таким образом, конечная разность второго порядка есть Аналогичным образом могут быть определены конечные разности произвольного порядка.

* 1. **Разделенные разности и их свойства**

Пусть функция  задана на таблице  значений аргумента с произвольным шагом, причем точки таблицы занумерованы также в произвольном порядке.

Величины  называются разделенными разностями первого порядка функции  в узлах Аналогично определяются разделенные разности более высокого порядка:  - разделенная разность второго порядка в узлах  Разделенной разностью -го порядка называется число



Разделенные разности обладают рядом замечательных свойств, изложенных в следующих теоремах.

Теорема 2.5. Разделенная разность  является симметричной функцией своих аргументов  (то есть ее свойства не меняются при любой их перестановке).

**Экспериментальные результаты.**

**Задание № 1**

Восстановить многочлен Лагранжа, удовлетворяющий приведенным исходным данным.

**Дано:** Вариант 11

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 0 | 1 | 2 | 3 | 4 |
|  | -3 | -2 | -1 | 0 | 4 |
|  | 6 | 0 | 2 | 0 | 6 |

**Задание № 2.**

Используя схему Эйткена, вычислить приближенное значение функции , заданной таблично при данном значении аргумента .

**Дано:**

|  |  |  |  |
| --- | --- | --- | --- |
| № варианта |  |  |  |
| 11 | 1.785 | 1.20 | 11.4725 |

**Обработка результатов эксперимента.**

**Задание № 1. решение:**

Здесь заданы лишь пять узлов сетки. Следовательно, можно восстановить многочлен Лагранжа четвертого порядка:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 0 | 1 | 2 | 3 | 4 |
|  | -3 | -2 | -1 | 0 | 4 |
|  | 6 | 0 | 2 | 0 | 6 |

В начале вычислим коэффициенты

Тогда

Многочлен Лагранжа имеет вид:

**Задание № 2. решение:**

Вычислим , расположив данные по схеме Эйткена в таблице:

Составим таблицу и заполним по формулам ее столбцы, начиная с четвертого:

Следующий столбец таблицы заполняется аналогично:

На этом вычисления можно прекратить, так как совпадают до третьего знака, следовательно, с точностью Итоговая таблица с результатами вычислений приведена ниже.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |
| 0 |  |  |  |  |  |  |  |
| 1 |  |  |  |  |  |  |  |
| 2 |  |  |  |  |  |  |  |
| 3 |  |  |  |  |  |  |  |
| 4 |  |  |  |  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| 0 | 1.785 |  | 1206.3658 |  |  |
|  |  |  |  |  |  |
| 1 | 1.896 |  | 1425.5874 |  |  |
|  |  |  |  |  |  |
| 2 | 1.906 |  | 1834.8976 |  |  |
|  |  |  |  |  |  |
| 3 | 1.945 |  | 2225.1438 |  |  |
|  |  |  |  |  |  |
| 4 | 1.984 |  | 2789.7543 |  |  |

**Выводы.**

В результате выполнения данной лабораторной работы без дополнительных условий был восстановлен многочлен Лагранжа имеющий степень, на единицу меньшую числа точек интерполяционной таблицы. Был использован восстанавливаемый интерполяционный многочленов невысоких степеней, для нахождения приближенных значений функции для любых значений аргумента, лежащих между узлами заданной сетки.

**ПРИЛОЖЕНИЕ А**

**Задание №1**

#include <iostream>

#include <conio.h>

using namespace std;

int main()

{

int x[4], y[4], x\_x[4][4];

double a[4], b[4], temp;

do

{

for (int i = 0; i < 4; ++i)

{

cout << "x" << i << " = ";

cin >> x[i];

cout << "y" << i << " = ";

cin >> y[i];

}

for (int i = 0; i < 4; ++i)

for (int j = 0; j < 4; ++j)

x\_x[i][j] = x[i] - x[j];

for (int i = 0; i < 4; ++i)

{

a[i] = y[i];

for (int j = 0; j < 4; ++j)

if (i != j) a[i] /= x\_x[i][j];

}

b[3] = 0;

for (int i = 0; i < 4; ++i) b[3] += a[i];

b[2] = 0;

for (int i = 0; i < 4; ++i)

for (int j = 0; j < 4; ++j)

if (i != j) b[2] -= x[j] \* a[i];

b[1] = 0;

for (int i = 0; i < 4; ++i)

for (int j = 0; j < 4; ++j)

if (i != j) for (int k = 0; k < 4; ++k)

if ((i != k) && (j != k)) b[1] += x[j] \* x[k] \* a[i];

b[1] /= 2;

b[0] = 0;

for (int i = 0; i < 4; ++i)

{

temp = a[i];

for (int j = 0; j < 4; ++j)

if (i != j) temp \*= x[j];

b[0] -= temp;

}

cout << b[3] << "x^3+" << b[2] << "x^2+" << b[1] << "x+" << b[0] << endl;

temp = \_getch();

} while (temp != 27);

return 0;

}

**Задание №2**

#include <iostream>

#include <conio.h>

using namespace std;

int main()

{

const double h = 0.01;

double a[10][8],x;

bool br;

do

{

cout << "x = ";

cin >> x;

for (int i = 0; i < 8; ++i)

{

cout << "x" << i << " = ";

cin >> a[0][i];

cout << "y" << i << " = ";

cin >> a[1][i];

a[2][i] = a[0][i] - x;

}

for (int i = 1; i < 8; ++i)

a[3][i] = (a[1][i - 1] \* a[2][i] - a[1][i] \* a[2][i - 1]) / (a[0][i] - a[0][i - 1]);

for (int j = 4; j < 10; ++j)

{

for (int i = j - 2; i < 8; ++i)

{

a[j][i] = (a[j-1][i - j + 2] \* a[2][i] - a[j-1][i] \* a[2][i - j + 2]) / (a[0][i] - a[0][i - j + 2]);

if (((a[j][i] - a[j - 1][i]) < h) || ((a[j][i] - a[j - 1][i - 1]) < h))

{

cout << "y(" << x << ") = " << round(a[j][i] / h)\*h << endl;

br = true;

break; }

}

if (br) break; }

x = \_getch();

} while (x != 27);

return 0;

}

Протокол

1. Задали исходные данные.
2. Определили наибольший возможный порядок многочлена для восстановления многочлена Лагранжа соответствующего порядка.
3. Вычислили коеффиценты соответствующих членов.
4. Привели подобные и восстановили многочлен Лагранжа.
5. В программе прописали алгоритм схемы Эйткена.
6. Вычислили приближенное значение функции , заданной таблично при данном значении аргумента.
7. Произвели округление результатов и вывели ответ.