**Министерство науки и высшего образования РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра ИС**

отчет

**по лабораторной работе №3**

**по дисциплине «Конструирование программ»**

Тема: Интерполирование и экстраполирование данных. Интерполяционный многочлен Ньютона

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

Санкт-Петербург

2020

**Цель работы.**

Заработать навыки работы с интерполяционным многочленом Ньютона с конечными разностями для интерполяции вперед. В ней используются только конечные разности, расположенные в верхней косой строке таблицы конечных разностей. Использовать разности нижней косой строки, и получить многочлен Ньютона с конечными разностями для интерполяции назад.

**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. **Интерполяционный многочлен Ньютона с конечными разностями**

Если интерполируемая функция задана на таблице с постоянным шагом , то можно использовать связь между конечными и разделенными разностями:  В этом случае многочлен Ньютона можно записать несколько в ином виде:

Пусть 



Преобразуем разделенные разности в конечные:  тогда





 и так далее.

Тогда многочлен Ньютона можно переписать в следующем виде:

 (2.10.1)

Эту формулу называют интерполяционным многочленом Ньютона с конечными разностями для интерполяции вперед. В ней используются только конечные разности, расположенные в верхней косой строке таблицы конечных разностей. Если использовать разности нижней косой строки, то аналогично получим многочлен Ньютона с конечными разностями для интерполяции назад:



**Экспериментальные результаты.**

**Задание № 1**

Построить интерполяционный многочлен Ньютона по неравноотстоящей сетке узлов и найти приближенное значение интерполируемой функции при значении аргумента .

**Дано:** Вариант 11

|  |  |  |  |
| --- | --- | --- | --- |
| № варианта |  |  |  |
| 11 | 0.552 | 0.21 | 5.4739 |

**Задание № 2.**

Вычислить приближенное значение функции  по интерполяцион-ной формуле Ньютона для интерполяции вперед или назад.

**Дано:**

|  |  |  |  |
| --- | --- | --- | --- |
| № варианта |  |  |  |
| 11 | 0.4675 | 4.85 | 18.7686 |

**Обработка результатов эксперимента.**

**Задание № 1. решение:**

Запишем получаемые конечные разности в виде таблицы, однако, в целях экономии места, не будем писать разности между строк  и . Тогда получим

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |
| 0.35 | 2.73951 | -7.311833 | 14.243057 | -15.227431 | -102.232694 | 825.705728 |
| 0.41 | 2.30080 | -5.602667 | 11.806667 | -36.696296 | 137.221967 |  |
| 0.47 | 1.96464 | -4.422000 | 6.302222 | -5.135244 |  |  |
| 0.51 | 1.78776 | -3.854800 | 5.429231 |  |  |  |
| 0.56 | 1.59502 | -3.149000 |  |  |  |  |
| 0.64 | 1.34310 |  |  |  |  |  |

Далее по формуле (2.7.1) вычисляем

Подставляя теперь значение вместо аргумента в найденную формулу, вычислим приближенное значение функции

**Задание № 2. решение:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 0.45 | 0.46 | 0.47 | 0.48 | 0.49 | 0.50 |
|  | 20.1946 | 19.6133 | 18.9425 | 18.1746 | 17.3010 | 16.3123 |
|  | 0.51 | 0.52 | 0.53 | 0.54 | 0.55 | 0.56 |
|  | 15.1984 | 13.9484 | 12.5508 | 10.9937 | 9.2647 | 7.3510 |

Составим таблицу конечных разностей. Ограничимся при этом разностями четвертого порядка, так как они практически постоянны. Для вычисления значения функции при  воспользуемся формулой Ньютона для интерполяции вперед. При этом разности из таблицы лучше всего брать стоящие по той диагонали, которая ближе всего расположена к . В таблице разностей эта диагональ подчеркнута.

Тогда и

с точностью

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| 0.45 | 20.1946 |  |  |  |  |
|  |  | -0.5813 |  |  |  |
| 0.46 | 19.6133 |  | -0.0895 |  |  |
|  |  | -0.6708 |  | -0.0076 |  |
| 0.47 | 18.9425 |  | -0.0971 |  | -0.0010 |
|  |  | -0.7679 |  | -0.0086 |  |
| 0.48 | 18.1746 |  | -0.1057 |  | -0.0008 |
|  |  | -0.8736 |  | -0.0094 |  |
| 0.49 | 17.3010 |  | -0.1151 |  | -0.0007 |
|  |  | -0.9887 |  | -0.0101 |  |
| 0.50 | 16.3123 |  | -0.1252 |  | -0.0008 |
|  |  | -1.1139 |  | -0.0109 |  |
| 0.51 | 15.1984 |  | -0.1361 |  | -0.0006 |
|  |  | -1.2500 |  | -0.0115 |  |
| 0.52 | 13.9484 |  | -0.1476 |  | -0.0004 |
|  |  | -1.3976 |  | -0.0119 |  |
| 0.53 | 12.5508 |  | -0.1595 |  | -0.0005 |
|  |  | -1.5571 |  | -0.0124 |  |
| 0.54 | 10.9937 |  | -0.1719 |  | -0.0004 |
|  |  | -1.7290 |  | -0.0128 |  |
| 0.55 | 9.2647 |  | -0.1847 |  |  |
|  |  | -1.9137 |  |  |  |
| 0.56 | 7.3510 |  |  |  |  |

Аналогично, для Воспользуемся формулой для интерполяции назад:

с той же точностью, что и в предыдущей формуле (

**Выводы.**

В ходе выполнения данной лабораторной была выпполнена работа с интерполяционным многочленом Ньютона с конечными разностями для интерполяции вперед. В ней используются только конечные разности, расположенные в верхней косой строке таблицы конечных разностей. Использовалои разности нижней косой строки, и получили многочлен Ньютона с конечными разностями для интерполяции назад.

**ПРИЛОЖЕНИЕ А**

**Задание №1**

#include <iostream>

#include <conio.h>

using namespace std;

int main()

{

double a[9][8], x, b[8];

do

{

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

{

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

cin >> a[0][i];

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

cin >> a[1][i];

}

for (int i = 2; i < 9; ++i)

for (int j = 0; j < 9 - i; ++j)

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

for (int z = 7; z >= 0; --z)

{

b[z] = a[z + 1][0];

if (z == 7) continue;

for (int i = 0; i < z + 1; ++i)

b[z] -= a[z+2][0] \* a[0][i];

if (z == 6) continue;

for (int i = 0; i < z + 2; ++i)

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

b[z] += a[z+3][0] \* a[0][i] \* a[0][j];

if (z == 5) continue;

for (int i = 0; i < z + 3; ++i)

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

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

b[z] -= a[z + 4][0] \* a[0][i] \* a[0][j] \* a[0][k];

if (z == 4) continue;

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

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

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

for (int l = 0; l < k; ++l)

b[z] += a[z + 5][0] \* a[0][i] \* a[0][j] \* a[0][k] \* a[0][l];

if (z == 3) continue;

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

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

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

for (int l = 0; l < k; ++l)

for (int s = 0; s < l; ++s) b[z] -= a[z + 6][0] \* a[0][i] \* a[0][j] \* a[0][k] \* a[0][l] \* a[0][s];

if (z == 2) continue;

for (int i = 0; i < z + 6; ++i)

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

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

for (int l = 0; l < k; ++l)

for (int s = 0; s < l; ++s)

for (int u = 0; u < s; ++u)

b[z] += a[z + 7][0] \* a[0][i] \* a[0][j] \* a[0][k] \* a[0][l] \* a[0][s] \* a[0][u];

}

b[0] -= a[8][0] \* a[0][0] \* a[0][1] \* a[0][2] \* a[0][3] \* a[0][4] \* a[0][5] \* a[0][6];

for (int i = 7; i >= 1; --i)

cout << b[i] << "x^" << i << "+";

cout << b[0] << endl;

do

{

cout << "x = ";

cin >> x;

double temp = 1, result = 0;

result += b[0];

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

{

temp \*= x;

result += temp\*b[i];

}

cout << "y(" << x << ") = " << result << endl;

x = \_getch();

} while (x != 27);

x = \_getch();

} while (x != 27);

return 0;

}

**Задание №2.**

#include <iostream>

#include <conio.h>

using namespace std;

int fact(int n)

{

if (n == 1) return 1;

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

}

int main()

{

double a[6][10], x, q, h, result, temp;

int k;

do

{

cout << "x = ";

cin >> x;

for (int i = 0; i < 10; ++i)

{

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

cin >> a[0][i];

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

cin >> a[1][i];

}

h = a[0][1] - a[0][0];

for (int j = 2; j < 6; ++j)

for (int i = 0; i < 9; ++i)

a[j][i] = a[j - 1][i + 1] - a[j - 1][i];

if (x > a[0][4])

{

for (k = 5; k < 10; ++k)

if (a[0][k] > x) break;

q = (x - a[0][k]) / h;

result = a[1][k];

temp = 1;

for (int i = 1; i < 5; ++i)

{

temp \*= q + i - 1;

result += a[i + 1][k - i] / fact(i)\*temp;

}

}

else

{

for (k = 4; k > 0; --k)

if (a[0][k] < x) break;

q = (x - a[0][k]) / h;

result = a[1][k];

temp = 1;

for (int i = 1; i < 5; ++i)

{

temp \*= q - i + 1;

result += a[i+1][k-1] / fact(i)\*temp;

}

}

cout << "y(" << x << ") = " << result << endl;

x = \_getch();

} while (x != 27);

}

Протокол

1. Задаем исходные данные в программу.
2. Создаем двухмерный массив для хранения сетки узлов.
3. Определяем члены по неравноотстоящей сетке узлов.
4. Построим интерполяционный многочлен Ньютона.
5. Найдем приближенное значение интерполируемой функции при заданном значении аргумента.
6. Вычисляем приближенное значение функции  по интерполяцион-ной формуле Ньютона для интерполяции вперед.
7. Вычисляем приближенное значение функции  по интерполяцион-ной формуле Ньютона для интерполяции назад.