Университет ИТМО

Факультет ФПИ и КТ

Лабораторная работа №5

“ Интерполяция функции”

По вычислительной математике

Вариант 8

Выполнил: Рогачев М.

Группа: P32082

Проверил: Машина Е.

Санкт-Петербург

2023

Цель:

решить задачу интерполяции, найти значения функции при заданных значениях аргумента, отличных от узловых точек.

Задание:

Вычислительная реализация задачи:

1. Выбрать из табл. 1 заданную по варианту таблицу $y = f(x)$ (таблица 1.1 – таблица 1.5);
2. Построить таблицу конечных разностей для заданной таблицы. Таблицу отразить в отчете;
3. Вычислить значения функции для аргумента X_1 (см. табл.1), используя первую или вторую интерполяционную формулу Ньютона. Обратить внимание какой конкретно формулой необходимо воспользоваться;
4. Вычислить значения функции для аргумента X_2 (см. табл. 1), используя первую или вторую интерполяционную формулу Гаусса. Обратить внимание какой конкретно формулой необходимо воспользоваться;
5. Подробные вычисления привести в отчете.

Вычислительная часть лабораторной работы №5.

X	y	X_1	X_2
1,25	1,2438	1,852	1,652
1,10	0,2234		
1,40	2,2644		
1,55	3,2984		
1,70	4,3222		
1,85	5,3516		
2,0	6,3867		

N	N	y_i	Δy_i	$\Delta^2 y_i$	$\Delta^3 y_i$	$\Delta^4 y_i$	$\Delta^5 y_i$	$\Delta^6 y_i$
-3	0	0,2234	1,0204	0,0002	0,0132	-0,0368	0,0762	-0,1313
-2	1	1,2438	1,0206	0,0134	-0,0236	-0,0394	-0,0551	
-1	2	2,2644	1,034	-0,0102	0,0158	-0,0157		
0	3	3,2984	1,0238	0,0056	0,0001			
1	4	4,3222	1,0294	0,0057				
2	5	5,3516	1,0351					
3	6	6,3867						

Вычисление X_1. Вторая интерполяционная формула Ньютона

$$t = \frac{x - x_n}{h}$$

$$N_6(x_1) = y_6 + t\Delta y_5 + \frac{t(t+1)}{2!}\Delta^2 y_4 + \frac{t(t+1)(t+2)}{3!}\Delta^3 y_3 + \frac{t(t+1)(t+2)(t+3)}{4!}\Delta^4 y_2 + \frac{t(t+1)(t+2)(t+3)(t+4)}{5!}\Delta^5 y_1 + \frac{t(t+1)(t+2)(t+3)(t+4)(t+5)}{6!}\Delta^6 y_0 =$$

$$= 5{,}3655$$

Вычисление X_2. Первая интерполяционная формула Гаусса

$$t = \frac{x - x_0}{h}$$

$$P_6(1{,}652) = y_0 + t\Delta y_0 + \frac{t(t-1)}{2!}\Delta^2 y_{-1} + \frac{(t+1)t(t-1)}{3!}\Delta^3 y_{-1} + \frac{(t+1)t(t-1)(t-2)}{4!}\Delta^4 y_{-2} + \frac{(t+2)(t+1)t(t-1)(t-2)}{5!}\Delta^5 y_{-2} + \frac{(t+2)(t+1)t(t-1)(t-2)(t-3)}{6!}\Delta^6 y_{-3} = -612{,}129$$

Программная реализация задачи:

Пример работы:

```
Это пятая лабораторная работа по Вычислительной математике
Введите:
0 - для выхода
1 - для ввода данных из консоли
2 - для ввода данных из файла
3 - выбор функции
3
Введите номер функции:
0 - sin(x)
1 - x^2 + 5 * x + 3
2 - x
0
Введите a:
2
Введите b:
4
Введите n:
5
Введите x дя которого надо найти значение: 3
Таблица конечных разностей:
0,9093  -0,5846 -0,3333 0,1400  0,0136
0,6755  -0,8512 -0,1653 0,1617
0,3350  -0,9834 0,0288
-0,0584 -0,9604
-0,4425
Значение по Лагранжу: 0.14123215464300737
Значение по Ньютону: 0.14123215464300734
```

Листинг программы:

```
import java.util.ArrayList;

public class Methods {
    public static double lagrange(ArrayList<Point> points, double solve) {
        double ans = 0.0;

        for (int i = 0; i < points.size(); i++) {
            double basis = 1.0;

            for (int j = 0; j < points.size(); j++) {
                if (i != j) {
                    basis *= (solve - points.get(j).getX()) /
(points.get(i).getX() - points.get(j).getX());
                }
            }

            ans += points.get(i).getY() * basis;
        }

        return ans;
    }
```

```
    public static double[][] finiteDifferences(ArrayList<Point> points) {
        int n = points.size();
        double[][] table = new double[n][n];

        for (int i = 0; i < n; i++) {
            table[i][0] = points.get(i).getY();
        }

        for (int j = 1; j < n; j++) {
            for (int i = 0; i < n - j; i++) {
                table[i][j] = (table[i+1][j-1] - table[i][j-1]) /
(points.get(i + j).getX() - points.get(i).getX());
            }
        }

        return table;
    }

    public static double newtonPolynomial(ArrayList<Point> points, double
solve) {
        double[][] table = finiteDifferences(points);
        double[] d = new double[table.length];

        for (int i = 0; i < table.length; i++) {
            d[i] = table[0][i];
        }

        double result = d[0];
        double product = 1;

        for (int i = 1; i < points.size(); i++) {
            product *= (solve - points.get(i - 1).getX());
            result += d[i] * product;
        }

        return result;
    }
}
```

Вывод:

в ходе лабораторной работы я научился реализовывать в программе методы для интерполяции функций по точкам, такие как многочлен Лагранжа и Ньютона.