Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет ИТМО»

Факультет программной инженерии и компьютерной техники

Отчет по лабораторной работе №5

«Интерполяция функции»

По дисциплине «Вычислительная математика»

Вариант 8

Выполнила: Иванова Мария Максимовна

Группа: P3208

Преподаватель: Машина Екатерина Алексеевна

Санкт-Петербург

~ 2024 ~

Рабочие формулы:

$$Ln(x) = \sum_{i=0}^{n} y_i \frac{(x-x_0)(x-x_1)\dots(x-x_{i-1})(x-x_{i+1})\dots(x-x_n)}{(x_i-x_0)(x_i-x_1)\dots(x_i-x_{i-1})(x_i-x_{i+1})\dots(x_i-x_n)}$$

$$\boldsymbol{N_n(x) = f(x_0) + \sum_{k=1}^{n} f(x_0, x_1, \dots, x_k) \prod_{j=0}^{k-1} (x - x_j)}$$

$$\begin{aligned} P_n(x) = {} & y_0 + t\Delta y_0 + \frac{t(t-1)}{2!}\Delta^2 y_{-1} + \frac{(t+1)t(t-1)}{3!}\Delta^3 y_{-1} \\ & + \frac{(t+1)t(t-1)(t-2)}{4!}\Delta^4 y_{-2} \\ & + \frac{(t+2)(t+1)t(t-1)(t-2)}{5!}\Delta^5 y_{-2} \dots \\ & + \frac{(t+n-1)\dots(t-n+1)}{(2n-1)!}\Delta^{2n-1} y_{-(n-1)} \\ & + \frac{(t+n-1)\dots(t-n)}{(2n)!}\Delta^{2n} y_{-n} \end{aligned}$$

$$\begin{aligned} P_n(x) = {} & y_0 + t\Delta y_{-1} + \frac{t(t+1)}{2!}\Delta^2 y_{-1} + \frac{(t+1)t(t-1)}{3!}\Delta^3 y_{-2} \\ & + \frac{(t+2)(t+1)t(t-1)}{4!}\Delta^4 y_{-2} + \cdots \\ & + \frac{(t+n-1)\dots(t-n+1)}{(2n-1)!}\Delta^{2n-1} y_{-n} \\ & + \frac{(\boldsymbol{t}+\boldsymbol{n})(t+n-1)\dots(t-n+\boldsymbol{1})}{(2n)!}\Delta^{2n} y_{-n} \end{aligned}$$

Вычислительная часть:

ВычМат Лаб5 Вычислительная часть Вариант 8

x	1,10	1,25	1,40	1,55	1,70	1,85	2,00
y	0,2234	1,2438	2,2644	3,2984	4,3222	5,3516	6,3867

Таблица конечных разностей:

x_i	y_i	Δy_i	$\Delta^2 y_i$	$\Delta^3 y_i$	$\Delta^4 y_i$	$\Delta^5 y_i$	$\Delta^6 y_i$
$x_0=1,10$	0,2234	1,0204	0,0002	0,0132	-0,0368	0,0762	-0,1313
$x_1=1,25$	1,2438	1,0206	0,0134	-0,0236	0,0394	-0,0551	–
$x_2=1,40$	2,2644	1,034	-0,0102	0,0158	-0,0157	–	–
$x_3=1,55$	3,2984	1,0238	0,0056	0,0001	–	–	–
$x_4=1,70$	4,3222	1,0294	0,0057	–	–	–	–
$x_5=1,85$	5,3516	1,0351	–	–	–	–	–
$x_6=2,00$	6,3867	–	–	–	–	–	–

$$\Delta^k y_i = \Delta^{k-1} y_{i+1} - \Delta^{k-1} y_i$$

$X_1 = 1,852$

Т.к. $X_1 > x_3 \Rightarrow$ используем вторую интерполяционную формулу Ньютона

$h = 1,25 - 1,10 = 0,15$

$$t = \frac{x - x_n}{h} = \frac{1,852 - 2}{0,15} = -0,99$$

$$N_n(x) = y_n + t\Delta y_{n-1} + \frac{t(t+1)}{2!}\Delta^2 y_{n-2} + \ldots + \frac{t(t+1)\ldots(t+n-1)}{n!}\Delta^n y_0$$

$$N_6(1,852) = 6,3876 + (-0,99)\cdot 1,0351 + \frac{-0,99\cdot 0,01}{2!}\cdot 0,0057 +$$

$$+ \frac{-0,99\cdot 0,01\cdot 1,01}{3!}\cdot 0,0001 + \frac{-0,99\cdot 0,01\cdot 1,01\cdot 2,01}{4!}\cdot(-0,0157) +$$

$$+ \frac{-0,99\cdot 0,01\cdot 1,01\cdot 2,01\cdot 3,01}{5!}\cdot(-0,0551) + \frac{-0,99\cdot 0,01\cdot 1,01\cdot 2,01\cdot 3,01\cdot 4,01}{6!}\cdot$$

$$\cdot(-0,1313) = \underline{5,3627}$$

$X_2 = 1{,}652$

$a = x_3 = 1{,}55$

$X_2 > a \Rightarrow$ первая интерполяционная формула Гаусса

$$t = \frac{x-a}{h} = \frac{1{,}652 - 1{,}55}{0{,}15} = 0{,}68$$

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

$$P_6(1{,}652) = 3{,}2984 + 0{,}68 \cdot 1{,}0238 + \frac{0{,}68 \cdot (-0{,}32)}{2!} \cdot (-0{,}0102) + \frac{1{,}68 \cdot 0{,}68 \cdot (-0{,}32)}{3!} \cdot 0{,}0158 + \frac{1{,}68 \cdot 0{,}68 \cdot (-0{,}32) \cdot (-1{,}32)}{4!} \cdot 0{,}039 + \frac{2{,}68 \cdot 1{,}68 \cdot 0{,}68 \cdot (-0{,}32) \cdot (-1{,}32)}{5!} \cdot (-0{,}0551) + \frac{2{,}68 \cdot 1{,}68 \cdot 0{,}68 \cdot (-0{,}32) \cdot (-1{,}32) \cdot (-2{,}32)}{6!} \cdot (-0{,}1313) =$$

$$= 3{,}9954$$

Листинг программы:

```
public class Lagrange {

    public double solve(double [] x, double [] y, int amount, double t ){
        double result = 0;
        for (int i = 0; i < amount; i++) {
            double temp = y[i];
```

```java
            for (int j = 0; j < amount; j++) {
                if (j != i) {
                    temp *= (t - x[j]) / (x[i] - x[j]);
                }
            }
            result += temp;
        }
        return result;
    }
}
```

```java
package org.example;

public class Newton {

    public double solve(double[] x, double[] y, int amount, double t) {
        double[][] dividedDifferences = new double[amount][amount];

        for (int i = 0; i < amount; i++) {
            dividedDifferences[i][0] = y[i];
        }

        for (int j = 1; j < amount; j++) {
            for (int i = 0; i < amount - j; i++) {
                dividedDifferences[i][j] = (dividedDifferences[i + 1][j - 1]
- dividedDifferences[i][j - 1]) / (x[i + j] - x[i]);
            }
        }



        double result = dividedDifferences[0][0];
        double term = 1.0;

        for (int i = 1; i < amount; i++) {
            term *= (t - x[i - 1]);
            result += dividedDifferences[0][i] * term;
        }

        return result;
    }
}
```

```java
public class Gauss {

    public double solve(double[] x, double[] y, int amount, double t) {
        int mid_index = amount / 2;
        double h = x[1] - x[0];
        double[][] centralDifferencesTable = computeDifferences(y);
        double f1 = firstInterpolationGaussForm(amount, x, y, mid_index, h,
centralDifferencesTable, t);
        double f2 = secondInterpolationGaussForm(amount, x, y, mid_index, h,
centralDifferencesTable, t);
        return (t > x[mid_index]) ? f1 : f2;
    }

    public double firstInterpolationGaussForm(int n, double[] x, double[] y,
int mid_index, double h, double[][] centralDifferencesTable, double value) {
        double t = (value - x[mid_index]) / h;
        double result = y[mid_index];
        double term = 1.0;
        double tProduct = t;
```

```
        for (int i = 1; i < n; i++) {
            term *= tProduct / i;
            if (i % 2 == 0) {
                tProduct *= (t - i / 2);
            } else {
                tProduct *= (t + (i - 1) / 2);
            }
            int index = mid_index - (i / 2);
            if (index < 0 || index >= n) continue;  // Avoid out-of-bounds
            result += term * centralDifferencesTable[index][i];
        }

        return result;
    }

    public double secondInterpolationGaussForm(int n, double[] x, double[] y,
int mid_index, double h, double[][] centralDifferencesTable, double value) {
        double t = (value - x[mid_index]) / h;
        double result = y[mid_index];
        double term = 1.0;
        double tProduct = t;

        for (int i = 1; i < n; i++) {
            term *= tProduct / i;
            if (i % 2 != 0) {
                tProduct *= (t + i / 2);
            } else {
                tProduct *= (t - i / 2);
            }
            int index = mid_index + (i - 1) / 2;
            if (index < 0 || index >= n) continue;  // Avoid out-of-bounds
            result += term * centralDifferencesTable[index][i];
        }

        return result;
    }

    public double[][] computeDifferences(double[] y) {
        int amount = y.length;
        double[][] differences = new double[amount][amount];

        for (int i = 0; i < amount; i++) {
            differences[i][0] = y[i];
        }

        for (int j = 1; j < amount; j++) {
            for (int i = 0; i < amount - j; i++) {
                differences[i][j] = differences[i + 1][j - 1] -
differences[i][j - 1];
            }
        }

        return differences;
    }

    public static void printDifferencesTable(double[] x, double[][]
differences) {
        int amount = x.length;

        System.out.println("Таблица конечных разностей:");
        System.out.printf("%-10s", "x");
        for (int j = 0; j < amount; j++) {
            System.out.printf("%-10s", "Δ^" + j + "y");
```

```
        }
        System.out.println();

        for (int i = 0; i < amount; i++) {
            System.out.printf("%-10.4f", x[i]);
            for (int j = 0; j < amount - i; j++) {
                System.out.printf("%-10.4f", differences[i][j]);
            }
            System.out.println();
        }
    }
```

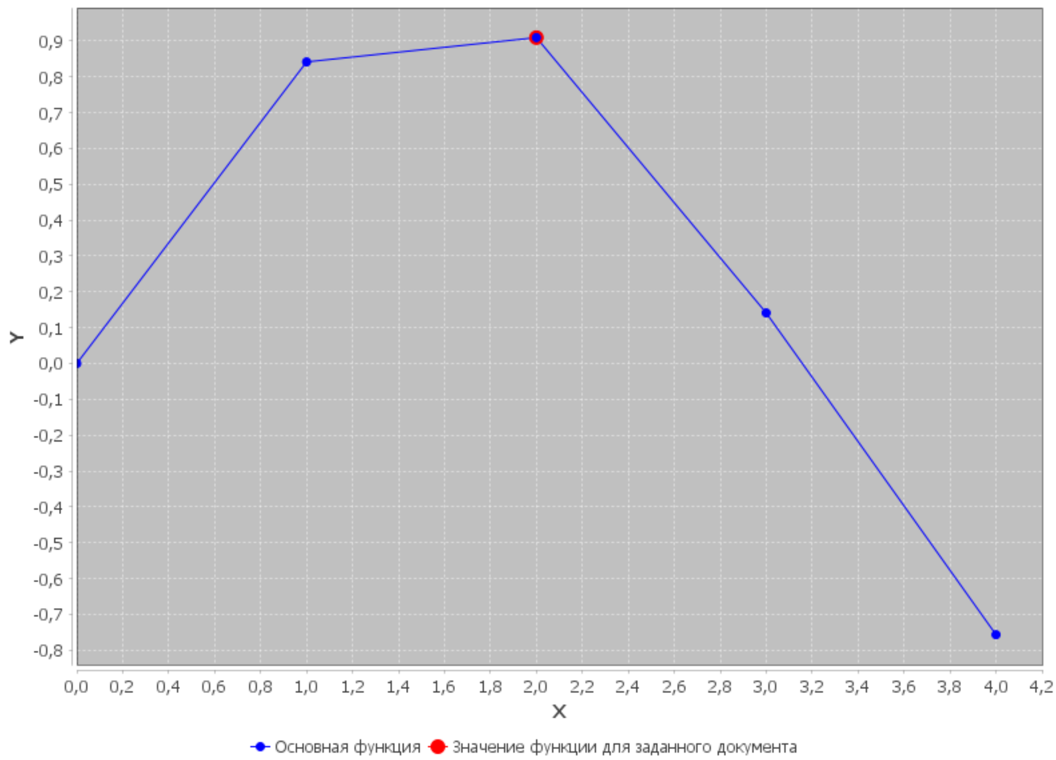
Вывод программы:

Выберете способ ввода: 1 - через консоль, 2 - через файл, 3 - выбрать функцию из предложенных
3
Выберете функцию:1. sin(x) 2. x^2
1
Выберете интервал (введите 2 числа через пробел)
0 5
Введите количество точек на интервале
5
Введите значение x, в котором будем считать приближенное значение функции
2
Таблица конечных разностей:

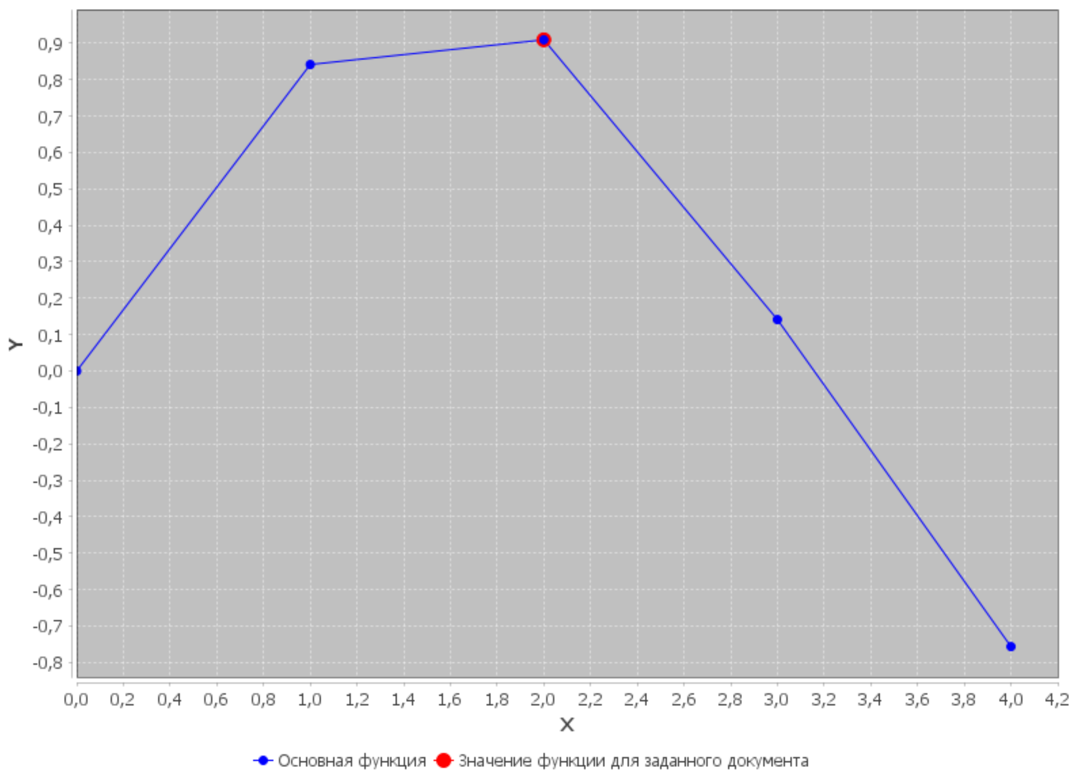
x	Δ^0y	Δ^1y	Δ^2y	Δ^3y	Δ^4y
0,0000	0,0000	0,8415	-0,7736	-0,0624	0,7686
1,0000	0,8415	0,0678	-0,8360	0,7063	
2,0000	0,9093	-0,7682	-0,1297		
3,0000	0,1411	-0,8979			
4,0000	-0,7568				

Многочлен Лагранжа:
Интерполяция при x = 2.0 равна 0.9092974268256817
Нельзя использовать метод Ньютона с разделенными разностями, x не должны быть равноотстоящими
Метод Гаусса:
Интерполяция при x = 2.0 равна 0.9092974268256817

Метод Лагранжа



Метод Гаусса



Вывод:

Во время выполнения работы я изучила понятие интерполяции, а также различные методы ее нахождения. Могу сделать вывод, что метод Лагранжа является наиболее удобным и точным для нахождения интерполяции при равноотстоящих узлах.