**ФПИиКТ**

**Рабочий протокол и отчет по**

**домашней работе №5**

**Вариант №6**

Ибадуллаев Алибаба Эльбрус оглы

Группа: P3215

Преподаватель: Малышева Татьяна Алексеевна

Санкт-Петербург

2022г.

# Лабораторная работа №5 «Интерполяция функции».

1. Цель лабораторной работы: решить задачу интерполяции, найти значения функции при заданных значениях аргумента, отличных от узловых точек.

Для исследования использовать:

* многочлен Лагранжа;
* многочлен Ньютона;
* многочлен Гаусса.

**Обязательное задание (до 80 баллов)**

**Варианты заданий для лабораторной работы №5**

1. **Вычислительная часть задачи (в отчет)**

***Таблица 2***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| x | y |  | № варианта | X1 | X2 |
| 0,50 | 1,5320 |  | 6 | 0,751 | 0,651 |
| 0,55 | 2,5356 |  |  |  |  |
| 0,60 | 3,5406 |  |  |  |  |
| 0,65 | 4,5462 |  |  |  |  |
| 0,70 | 5,5504 |  |  |  |  |
| 0,75 | 6,5559 |  |  |  |  |
| 0,80 | 7,5594 |  |  |  |  |

1. **Методы для реализации в программе:**

* 1 - Многочлен Лагранжа,
* 2 - Многочлен Ньютона с конечными разностями,

## Вычислительная реализация задачи:

**Метод Ньютона (формулой Ньютона для интерполирования назад)**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| X | Y | y | y | y | y | y | y |
| 0,5 | 1,532 | 1,0036 | 0,0014 | -0,0008 | -0,0012 | 0,0059 | -0,0166 |
| 0,55 | 2,5356 | 1,005 | 0,0006 | -0,002 | 0,0047 | -0,0107 |  |
| 0,6 | 3,5406 | 1,0056 | -0,0014 | 0,0027 | -0,006 |  |  |
| 0,65 | 4,5462 | 1,0042 | 0,0013 | -0,0033 |  |  |  |
| 0,7 | 5,5504 | 1,0055 | -0,002 |  |  |  |  |
| 0,75 | 6,5559 | 1,0035 |  |  |  |  |  |
| 0,8 | 7,5594 |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
| x1 | x2 |  | t1 | t2 |  | F(X1) | F(X2) |
| 0,751 | 0,651 |  | -0,98 | -2,98 |  | 6,57603287 | 4,56629484 |

**Метод Гаусса (x>a)**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| X | Y | y | y | y | y | y | y |
| 0,5 | 1,532 | 1,0036 | 0,0014 | -0,0008 | -0,0012 | 0,0059 | -0,0166 |
| 0,55 | 2,5356 | 1,005 | 0,0006 | -0,002 | 0,0047 | -0,0107 |  |
| 0,6 | 3,5406 | 1,0056 | -0,0014 | 0,0027 | -0,006 |  |  |
| 0,65 | 4,5462 | 1,0042 | 0,0013 | -0,0033 |  |  |  |
| 0,7 | 5,5504 | 1,0055 | -0,002 |  |  |  |  |
| 0,75 | 6,5559 | 1,0035 |  |  |  |  |  |
| 0,8 | 7,5594 |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
| x1 | x2 |  | t1 | t2 |  | F(X1) | F(X2) |
| 0,751 | 0,651 |  | 2,02 | 0,02 |  | 6,57603287 | 4,56629484 |

## Листинг программы:

private static double **methodNewton**(int n, double x, Double[] xs, Double[] ys, StringBuilder str) {  
 double result = 0;  
 double sum = 1;  
 double[][] matrix = new double[n][n];  
 double t;  
 // fulling the matrix with coefficients  
 for (int i = 0 ; i < n ; i++) {  
 for (int j = 0 ; j < n - i; j++){  
 if (i == 0) matrix[i][j] = ys[j];  
 else {  
 matrix[i][j] = matrix[i-1][j+1] - matrix[i-1][j];  
 }  
 }  
 }  
 int pos = 0;  
 **//find position in array**  
 for (int i = 0; i < n; i++ ){  
 if (xs[i]<x) pos=i;  
 else break;  
 }  
 **//brain of method**  
 if (pos < xs.length/2) {  
 t = (x - xs[pos])/(xs[1]-xs[0]);  
 for (int j = 0 ; j < n - pos; j++){  
 if (j == 0) {  
 result = matrix[0][pos];  
 continue;  
 }  
 sum \*= matrix[j][pos];  
 for (int k = 0; k < j; k++){  
 sum \*= t - k;  
 }  
 sum /= *findFactorial*(j);  
 result += sum;  
 System.*out*.println(sum);  
 sum = 1;  
 }  
 } else {  
 t = (x - xs[n-1])/(xs[1]-xs[0]);  
 for (int j = 0 ; j < n; j++){  
 if (j == 0) {  
 result = matrix[0][n-1];  
 continue;  
 }  
 sum \*= matrix[j][n-j-1];  
 for (int k = 0; k < j; k++){  
 sum \*= t + k;  
 }  
 sum /= *findFactorial*(j);  
 System.*out*.println(t);  
 result += sum;  
  
 sum = 1;  
 }  
 }  
 **// writing the formula**  
 double h = xs[1] - xs[0];  
 for (int j = 0 ; j < n ; j++){  
 if (j == 0) {  
 str.append(matrix[0][0]).append(" + ");  
 continue;  
 }  
 for (int k = 0; k < j; k++){  
 str.append("(x-").append(xs[k]).append(")");  
 }  
 str.append("(").append(matrix[j][0]/*findFactorial*(j)/Math.*pow*(h,j)).append(")+");  
 }  
 return result;  
}  
  
private static double **methodLaGranje**(int n, double x, Double[] xs, Double[] ys, StringBuilder str) {  
 double sum = 1;  
 double result = 0;  
 for (int i = 0; i < n; i++){  
 for (int j = 0; j < n; j++){  
 if (j == i ) continue;  
 sum \*= (x - xs[j])/(xs[i] - xs[j]);  
 str.append("(x-(").append(xs[j]).append("))/(").append(xs[i] - xs[j]).append(")\*");  
 }  
  
 str.append(ys[i]).append("+");  
 sum = sum \* ys[i];  
 result += sum;  
 sum=1;  
 }  
 return result;  
}  
  
private static double **findFactorial**(double x){  
 double result = 1;  
 for (int i = 1; i <= x; i++){  
 result \*= i;  
 }  
 return result;  
}

## Вывод:

Я научился пользоваться разными методами интерполирования, и также разобрался в теоретической части в этой области.