**Министерство образования и науки Российской Федерации**

**Федеральное государственное автономное образовательное**

**учреждение высшего образования**

**«Университет ИТМО»**

**Мегафакультет:** Компьютерных технологий и управления

**Направление:** 09.03.04 «Программная инженерия»

**Лабораторная работа 5**

По дисциплине:

«Вычислительная математика»

Вариант 10

На тему:

«Интерполяция функций»

**Выполнила:**

студентка группы **P3214**

Маньшина Елена

Витальевна

**Преподаватель:**

Малышева Татьяна

Алексеевна

Санкт-Петербург

2020

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

Для исследования использовать:

• линейную и квадратичную интерполяцию;

• многочлен Лагранжа;

• многочлен Ньютона.

**Проведение исследования**

2.1С помощью линейной и квадратичной интерполяции найти приближенное значение функции при х= Х1 (см. табл. 1 - 4). Подробные вычисления привести в отчете.

|  |  |
| --- | --- |
| **№ варианта** | **X1** |
| **10** | **0.622** |

Таблица 2

|  |  |
| --- | --- |
| **x** | **y** |
| **0,593** | **0,532** |
| **0,598** | **0,5356** |
| **0,605** | **0,5406** |
| **0,613** | **0,5462** |
| **0,619** | **0,5504** |
| **0,627** | **0,5559** |
| **0,632** | **0,5594** |

Линейная

Значение x=0,622 находится между узлами

Квадратичная

Используем квадратичную интерполяцию. Составим систему уравнений для ближайших узлов к точке x=0,622: соответственно

*В результате решения системы*

2.2. Найти приближенное значение функции при х= Х1 (см. табл. 1 - 4) с помощью многочлена Лагранжа. Подробные вычисления привести в отчете.

|  |  |
| --- | --- |
| **№ варианта** | **X1** |
| **10** | **0.622** |

Таблица 2

|  |  |
| --- | --- |
| **x** | **y** |
| **0,593** | **0,532** |
| **0,598** | **0,5356** |
| **0,605** | **0,5406** |
| **0,613** | **0,5462** |
| **0,619** | **0,5504** |
| **0,627** | **0,5559** |

Многочлен Лагранжа

2.3. Используя первую или вторую интерполяционную формулу Ньютона, вычислить значения функции при данных значениях аргумента (для значения Х2 и Х3, см. табл. 5 - 8). Подробные вычисления привести в отчете.

|  |  |  |
| --- | --- | --- |
| **№ варианта** | **X2** | **X3** |
| **10** | **0.523** | **0.759** |

Таблица 6

|  |  |
| --- | --- |
| **x** | **y** |
| **0,5** | **1,532** |
| **0,55** | **2,5356** |
| **0,60** | **3,5406** |
| **0,65** | **4,5462** |
| **0,7** | **5,5504** |
| **0,75** | **6,5559** |
| **0,8** | **7,5594** |

Для вычисления значения функции при х=0,523 воспользуемся формулой Ньютона для интерполирования вперед, т.к. х лежит в левой половине отрезка. Ограничимся конечными разностями шестого порядка:

*Примем =0.5 Тогда*

*Конечные разности*

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| xi |  | yi |  | ∆yi |  | yi |  | yi |  | 4yi |  | yi |  | yi |  |
| x0 | 0,5 | y0 | 1,532 | ∆y0 | 1,0036 | y0 | 0,0014 | y0 | -0,0008 | y0 | -0,0012 | y0 | 0,0059 | y0 | -0,0166 |
| x1 | 0,55 | y1 | 2,5356 | ∆y1 | 1,005 | y1 | 0,0006 | y1 | -0,002 | y1 | 0,0047 | y1 | -0,0107 |  |  |
| x2 | 0,6 | y2 | 3,5406 | ∆y2 | 1,0056 | y2 | -0,0014 | y2 | 0,0027 | y2 | -0,006 |  |  |  |  |
| x3 | 0,65 | y3 | 4,5462 | ∆y3 | 1,0042 | y3 | 0,0013 | y3 | -0,0033 |  |  |  |  |  |  |
| x4 | 0,7 | y4 | 5,5504 | ∆y4 | 1,0055 | y4 | -0,002 |  |  |  |  |  |  |  |  |
| x5 | 0,75 | y5 | 6,5559 | ∆y5 | 1,0035 |  |  |  |  |  |  |  |  |  |  |
| x6 | 0,8 | y6 | 7,5594 |  |  |  |  |  |  |  |  |  |  |  |  |

Для вычисления значения функции при х=0,759 воспользуемся формулой Ньютона для интерполирования назад, т.к. х лежит в правой половине отрезка. Ограничимся конечными разностями шестого порядка:

*Примем =0.5 Тогда*

2.4. Вычислить значения функции, используя интерполяционную формулу Ньютона для неравноотстоящих узлов (для х=Х4, см. табл. 1 - 4). При вычислениях учитывать только разделенные разности первого и второго порядков. Вычисления произвести дважды, используя различные узлы. Подробные вычисления привести в отчете.

|  |  |
| --- | --- |
| **№ варианта** | **X4** |
| **10** | **0.596** |

Таблица 2

|  |  |
| --- | --- |
| **x** | **y** |
| **0,593** | **0,532** |
| **0,598** | **0,5356** |
| **0,605** | **0,5406** |
| **0,613** | **0,5462** |
| **0,619** | **0,5504** |
| **0,627** | **0,5559** |

Для вычисления значения функции при х=0,596 за сначала возьмем 0,593 , затем 0,598

*Принимаем*

*Листинг программы*

**public class *LagrangePolynomial*** {  
 **private double**[] **X**;  
 **private double**[] **Y**;  
 **private double x**;  
  
 **public** LagrangePolynomial(**double**[] X, **double**[] Y, **double** x) {  
 **this**.**X** = X;  
 **this**.**Y** = Y;  
 **this**.**x** = x;  
 }  
  
 **public double** getAnswer() {  
 **double** y = 0;  
 **for** (**int** i = 0; i < **X**.**length**; i++) {  
 **double** l = 1;  
 **for** (**int** j = 0; j < **X**.**length**; j++) {  
 **if** (i != j) {  
 l \*= (**x** - **X**[j]) / (**X**[i] - **X**[j]);  
 }  
 }  
 y += l \* **Y**[i];  
 }  
 System.***out***.printf(**"Многочлен Лагранжа.Приближенное значение функции y=f(x) при x=%.4f для заданной таблицы %f\n"**, **x**, y);  
 **return** y;  
 }  
}

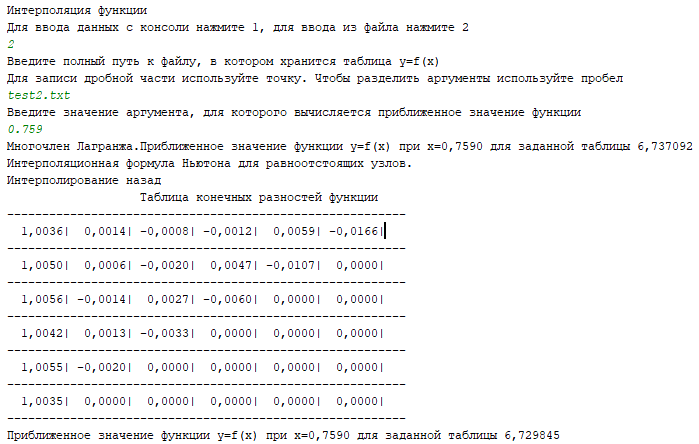
**public class NewtonPolynomial** {  
 **private double**[] **X**;  
 **private double**[] **Y**;  
 **private double x**;  
 **private boolean isEquallySpacedNodes** = **true**; *//являются ли узлы равноотстоящими* **double h**;*//расстояние между равноотстоящими узлами* **double**[][] **deltaY**; *// таблица конечных разностей функций* **public** NewtonPolynomial(**double**[] X, **double**[] Y, **double** x) {  
 **this**.**X** = X;  
 **this**.**Y** = Y;  
 **this**.**x** = x;  
 **int** i = 2;  
 **h** = X[1] - X[0];  
 **while** (**isEquallySpacedNodes** && i < X.**length**) {  
 **isEquallySpacedNodes** = Math.*abs*(X[i] - X[i - 1] - **h**) < 1.0e-10;  
 i++;  
 }  
 }  
  
 **public double** getAnswer() **throws** MethodException {  
 **if** (**isEquallySpacedNodes**) {  
 System.***out***.println(**"Интерполяционная формула Ньютона для равноотстоящих узлов."**);  
 **return** equallySpacedNodesMethod();  
 } **else** System.***out***.printf(**"\nИнтерполяционная формула Ньютона для неравноотстоящих узлов."**);  
 **return** unequalNodesMethod();  
 }  
  
 **public double** unequalNodesMethod() **throws** MethodException {  
 **int** x0\_index = getX0\_indexUnequalNodesMethod(); *//индекс x0 узла* **double** y1 = findSolutionUnequalNodesMethod(x0\_index - 1);  
 **double** y2 = findSolutionUnequalNodesMethod(x0\_index);  
 System.***out***.printf(**"\nПриближенное значение функции y=f(x) при x=%.4f для заданной таблицы %f"**, **x**, (y1 + y2) / 2);  
 **return** (y1 + y2) / 2;  
 }  
  
 **public int** getX0\_indexUnequalNodesMethod() **throws** MethodException {  
 **int** x0\_index = 0; *//индекс x0 узла* **if** (**X**.**length** < 3)  
 **throw new** MethodException(**"В таблице менее трех узлов"**);  
 **if** (**x** < **X**[1]) {  
 **return** 1;  
 }  
 **if** (**x** > **X**[**X**.**length** - 1]) {  
 **return X**.**length** - 3; *//становимся на 3 элемент с конца* }  
 **for** (**int** i = 0; i < **X**.**length**; i++) {  
 **if** (**x** < **X**[i]) {  
 x0\_index = i - 1;  
 **break**;  
 }  
 }  
 **int** ost = **X**.**length** - (x0\_index + 1);*//сколько элементов в массиве осталось после x0\_index* x0\_index = (ost == 1)?(x0\_index-1):x0\_index;  
 **return** x0\_index;  
 }  
  
 **public double** findSolutionUnequalNodesMethod(**int** x0\_index) {  
 **double** y;  
 **double** f\_x0x1\_ = (**Y**[x0\_index + 1] - **Y**[x0\_index]) / (**X**[x0\_index + 1] - **X**[x0\_index]);  
 **double** f\_x1x2\_ = (**Y**[x0\_index + 2] - **Y**[x0\_index + 1]) / (**X**[x0\_index + 2] - **X**[x0\_index + 1]);  
 **double** f\_x0x1x2\_ = (f\_x1x2\_ - f\_x0x1\_) / (**X**[x0\_index + 2] - **X**[x0\_index]);  
 y = **Y**[x0\_index] + f\_x0x1\_ \* (**x** - **X**[x0\_index]) + f\_x0x1x2\_ \* (**x** - **X**[x0\_index]) \* (**x** - **X**[x0\_index + 1]);  
 **return** y;  
 }  
  
 **public double** equallySpacedNodesMethod() **throws** MethodException {  
 **int** x0\_index = getX0\_indexForEquallySpacedNodesMethod(); *//индекс x0 узла* **double** y;  
 **double** t;  
 **if** (x0\_index > **X**.**length** / 2) {  
 *//правая половина отрезка* System.***out***.println(**"Интерполирование назад"**);  
 **int** xn\_index = (x0\_index == **X**.**length** - 1) ? **X**.**length** - 1 : x0\_index + 1;  
 getTable(0, xn\_index);  
 t = (**x** - **X**[xn\_index]) / **h**;  
 y = **Y**[xn\_index];  
 **double** k = 0;  
 **double** T = 1;  
 **for** (**int** j = 0; j < xn\_index; j++) {  
 k = t + j;  
 T = T \* k / (j + 1);  
 y += T \* **deltaY**[xn\_index - 1 - j][j];  
 }  
 } **else** {  
 *//левая половина отрезка* System.***out***.println(**"Интерполирование вперед"**);  
 getTable(x0\_index,**X**.**length**-1 );  
 t = (**x** - **X**[x0\_index]) / **h**;  
 y = **Y**[x0\_index];  
 **double** k = 0;  
 **double** T = 1;  
 **for** (**int** j = 0; j < **X**.**length** - (x0\_index + 1); j++) {  
 k = t - j;  
 T = T \* k / (j + 1);  
 y += T \* **deltaY**[0][j];  
 }  
 }  
 System.***out***.printf(**"Приближенное значение функции y=f(x) при x=%.4f для заданной таблицы %f"**, **x**, y);  
 **return** y;  
 }  
  
  
 **public int** getX0\_indexForEquallySpacedNodesMethod() **throws** MethodException {  
 **int** x0\_index = 0; *//индекс x0 узла* **if** (**X**.**length** < 2)  
 **throw new** MethodException(**"В таблице менее двух узлов"**);  
 **if** (**x** < **X**[1]) {  
 **return** 0;  
 }  
 **if** (**x** > **X**[**X**.**length** - 1]) {  
 **return X**.**length** - 1; *//становимся на последний элемент* }  
 **for** (**int** i = 0; i < **X**.**length**; i++) {  
 **if** (**x** < **X**[i]) {  
 x0\_index = i - 1;  
 **break**;  
 }  
 }  
 **return** x0\_index;  
 }  
  
 **public void** getTable(**int** x0\_index, **int** xn\_index) {  
 **deltaY** = **new double**[**X**.**length** - (x0\_index + 1)][**X**.**length** - (x0\_index + 1)];  
 **int** n = xn\_index; *//максимальный индекс строк и столбцов  
 //заполняем первую дельту y* **for** (**int** j = x0\_index; j < xn\_index; j++) {  
 **deltaY**[j - x0\_index][0] = **Y**[j+1] - **Y**[j];  
 }  
 *//заполнение таблицы конечных разностей функций* **for** (**int** i = x0\_index + 1; i < xn\_index; i++) { *//изменяется столбец* **for** (**int** j = x0\_index; j < n - 1; j++) { *//изменяется строка* **deltaY**[j - x0\_index][i - x0\_index] = **deltaY**[j - x0\_index + 1][i - x0\_index - 1] - **deltaY**[j - x0\_index][i - x0\_index - 1];  
 }  
 n--;*//с продвижением вправо количество строк в столце уменьшается* }  
 *//печать таблицы конечных разностей функций* System.***out***.println(**" Таблица конечных разностей функции"**);  
 System.***out***.println(**"---------------------------------------------------------"**);  
 n = xn\_index - x0\_index;  
 **for** (**int** i = 0; i < n; i++) {  
 **for** (**int** j = 0; j < n; j++) { *//изменяется строка* System.***out***.printf(**"%8.4f|"**, **deltaY**[i][j]);  
 }  
 System.***out***.println(**"\n---------------------------------------------------------"**);  
 }  
 }  
  
}

**Результаты работы программы**

**Содержимое test2.txt**

0.5 0.55 0.6 0.65 0.7 0.75 0.8  
1.532 2.5356 3.5406 4.5462 5.5504 6.5559 7.5594  
//x= 0.759 ожидаемое значение 6.737092 (исходя из вычислений, расположенных выше п. 2.3.)

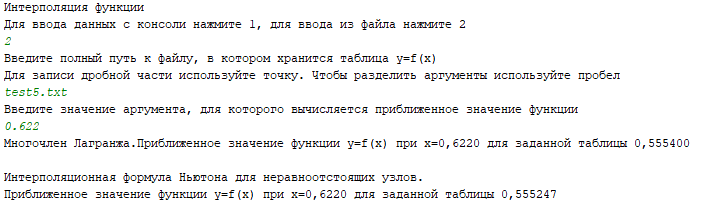
**Консольный вывод**

****

**Содержимое test5.txt**

0.593 0.598 0.605 0.613 0.619 0.627  
0.532 0.5356 0.5406 0.5462 0.5524 0.5559  
//x=0.622 ожидаемое значение 0.552497

**Консольный вывод**

****

**Вывод:** В ходе лабораторной работы я реализовала численные методы интерполирования , с помощью которых можно найти значения функции при заданных значениях аргумента, отличных от узловых точек.(Использовала многочлен Лагранжа, Первую и вторую интерполяционные формулы Ньютона и формулу Ньютона для неравноотстоящих узлов)