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

**Факультет программной инженерии и компьютерной техники**

**Вычислительная математика**

**Лабораторная работа №4**

**Вариант 8**

Группа: P3266

Выполнили:

Тажибаева Е.В Коляда А.А

Проверила:

Машина Е. А.

Г. Санкт-Петербург

2024

**Цель лабораторной работы**

Найти функцию, являющуюся наилучшим приближением заданной табличной функции по методу наименьших квадратов.

**Вычисления**

Таблица табулирования функции на интервале [-2;0), h=0,2

|  |  |
| --- | --- |
| **x** | **y** |
| -2 | -0,25 |
| -1,8 | -0,29193 |
| -1,6 | -0,32982 |
| -1,4 | -0,35468 |
| -1,2 | -0,35737 |
| -1 | -0,33333 |
| -0,8 | -0,28539 |
| -0,6 | -0,22141 |
| -0,4 | -0,14952 |
| -0,2 | -0,07499 |
| 0 | 0 |
|  |  |

Линейное и квадратичное приближения по 11 точкам на интервале [-2;0), h=0,2

Линейное:

1)

|  |
| --- |
| xi\*yi |
| 0,5 |
| 0,525474 |
| 0,527704 |
| 0,496555 |
| 0,428844 |
| 0,333333 |
| 0,228311 |
| 0,132848 |
| 0,059809 |
| 0,014997 |
| 0 |

Сумма

Сумма

Сумма

2)

|  |
| --- |
| xi^2 |
| 4 |
| 3,24 |
| 2,56 |
| 1,96 |
| 1,44 |
| 1 |
| 0,64 |
| 0,36 |
| 0,16 |
| 0,04 |
| 0 |

Сумма

3)

a=**0,136235, b=-0,10453164**

|  |  |  |  |
| --- | --- | --- | --- |
| **x** | **y** | **P(x)=ax+b** | **E** |
| -2 | -0,25 | -0,3770025 | -0,127 |
| -1,8 | -0,2919298 | -0,3497554 | -0,05783 |
| -1,6 | -0,3298153 | -0,3225083 | 0,007307 |
| -1,4 | -0,3546818 | -0,2952612 | 0,059421 |
| -1,2 | -0,3573698 | -0,2680142 | 0,089356 |
| -1 | -0,3333333 | -0,2407671 | 0,092566 |
| -0,8 | -0,2853881 | -0,21352 | 0,071868 |
| -0,6 | -0,2214131 | -0,1862729 | 0,03514 |
| -0,4 | -0,1495215 | -0,1590258 | -0,0095 |
| -0,2 | -0,074985 | -0,1317787 | -0,05679 |
| 0 | 0 | -0,1045316 | -0,10453 |

4)

Ср. откл=**0,074010539**

|  |
| --- |
| E^2 |
| 0,01613 |
| 0,003344 |
| 5,34E-05 |
| 0,003531 |
| 0,007984 |
| 0,008569 |
| 0,005165 |
| 0,001235 |
| 9,03E-05 |
| 0,003226 |
| 0,010927 |

|  |  |  |
| --- | --- | --- |
| Сумма E^2 |  |  |
| 0,060253 | **0,005478** | 0,074011 |

Квадратичное:   
  
1)

|  |
| --- |
| x^4 |
| 16 |
| 10,4976 |
| 6,5536 |
| 3,8416 |
| 2,0736 |
| 1 |
| 0,4096 |
| 0,1296 |
| 0,0256 |
| 0,0016 |
| 0 |

Сумма 40,5328

2)

|  |
| --- |
| x^3 |
| -8 |
| -5,832 |
| -4,096 |
| -2,744 |
| -1,728 |
| -1 |
| -0,512 |
| -0,216 |
| -0,064 |
| -0,008 |
| 0 |

Сумма -24,2

3)

|  |
| --- |
| x^2 |
| 4 |
| 3,24 |
| 2,56 |
| 1,96 |
| 1,44 |
| 1 |
| 0,64 |
| 0,36 |
| 0,16 |
| 0,04 |
| 0 |

Сумма 15,4

Сумма -11

4)

|  |
| --- |
| x^2\*y |
| -1 |
| -0,94585 |
| -0,84433 |
| -0,69518 |
| -0,51461 |
| -0,33333 |
| -0,18265 |
| -0,07971 |
| -0,02392 |
| -0,003 |
| 0 |

Сумма -4,62258169

5)

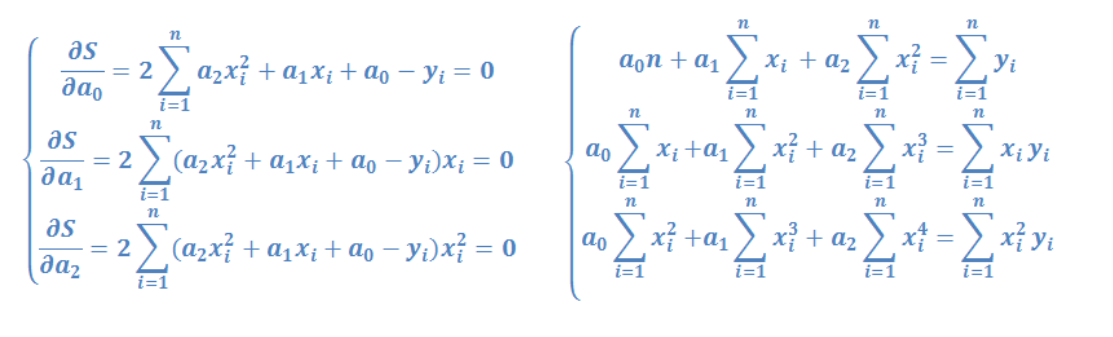
|  |
| --- |
| xi\*yi |
| 0,5 |
| 0,525474 |
| 0,527704 |
| 0,496555 |
| 0,428844 |
| 0,333333 |
| 0,228311 |
| 0,132848 |
| 0,059809 |
| 0,014997 |

Сумма 3,2478736

Сумма -2,6484377

6) Ср. откл= **0,011417713**

|  |  |  |
| --- | --- | --- |
| E^2 | Сумма E^2 |  |
| 0,0001 | 0,001434 | 0,00013 |
| 0,000246 |  |  |
| 7,37E-05 |  |  |
| 5,21E-06 |  |  |
| 6,04E-05 |  |  |
| 1,11E-05 |  |  |
| 6,74E-05 |  |  |
| 0,000361 |  |  |
| 0,000436 |  |  |
| 7,42E-05 |  |  |



a=0,21 , b=0,56 , c=0,02

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **x** | **y** | **P(x)==ax^2+bx+c** | **E** | |
| -2 | | | | -0,25 | | -0,26 | -0,01 |
| -1,8 | | | | -0,2919298 | | -0,3076 | -0,01567 |
| -1,6 | | | | -0,3298153 | | -0,3384 | -0,00858 |
| -1,4 | | | | -0,3546818 | | -0,3524 | 0,002282 |
| -1,2 | | | | -0,3573698 | | -0,3496 | 0,00777 |
| -1 | | | | -0,3333333 | | -0,33 | 0,003333 |
| -0,8 | | | | -0,2853881 | | -0,2936 | -0,00821 |
| -0,6 | | | | -0,2214131 | | -0,2404 | -0,01899 |
| -0,4 | | | | -0,1495215 | | -0,1704 | -0,02088 |
| -0,2 | | | | -0,074985 | | -0,0836 | -0,00861 |
| 0 | | | | 0 | | 0,02 | 0,02 |

**Листинг программы**

*/\*\*  
 \* Класс экспоненциальной функции fi(x) = a\* exp(bx)  
 \* Линеаризуем получем ln(fi(x)) = ln(a) + b \* x  
 \* Таким образом для полинома первой степени подставляем ln(y) и x,  
 \* получаем a[0] - ln(a[0])  
 \* окончательно преобразуем a[0] = pow(e,a[0]) - это и будет a для нашей функциц  
 \* b = a[1]  
 \*/*public class L4FunctionExp extends L4FunctionPoly {  
 public L4FunctionExp(){super(1);}  
  
 @Override  
 public String getFunctionName() {  
 return "Экспоненциальная функция";  
 }  
  
 @Override  
 public void CalcA(Dots dots) {  
 super.CalcA(dots);  
 //Пересчитываем a[0]  
 \_a[0] = Math.*exp*(\_a[0]);  
 // Оставляем без изменений a[1]  
 }  
  
 @Override  
 protected double getY(Dots dots, int index) {  
 double vl = dots.getDotY(index);  
 if(vl <= 0) {  
 \_wasError = true;  
 throw new CalcErrorException("Для экспоненциальной функции значение Y в массиве точек не может быть <= 0");  
 }  
 return Math.*log*(vl);  
 }  
  
 @Override  
 public String toString() {  
 return String.*format*("%f\*exp(%f \*x)",\_a[0],\_a[1]);  
 }  
  
 @Override  
 public double functionOf(double x) {  
 return \_a[0]\*Math.*exp*(x \*\_a[1]);  
 }  
}

*/\*\*  
 \* Класс логарифмической функции fi(x) = a + b\*ln(x)  
 \* Чтобы не путаться с a и b вот так сделаем, а не как в лекции  
 \* Таким образом для полинома первой степени подставляем y и ln(x),  
 \* получаем a[0] - a, a1[1] - b  
 \*/*public class L4FunctionLog extends L4FunctionPoly {  
 public L4FunctionLog() {super(1);}  
  
 @Override  
 public String getFunctionName() {  
 return "Логарифмическая функция";  
 }  
  
 @Override  
 protected double getX(Dots dots, int index) {  
 double vl = dots.getDotX(index);  
 if(vl <= 0) {  
 \_wasError = true;  
 throw new CalcErrorException("Для логарифмической функции значение X в массиве точек не может быть <= 0");  
 }  
 return Math.*log*(vl);  
 }  
  
 @Override  
 public String toString() {  
 return String.*format*("%f\*log(x) + %f",\_a[1],\_a[0]);  
 }  
  
 @Override  
 public double functionOf(double x) {  
 return \_a[0] + Math.*log*(x)\*\_a[1];  
 }  
}

*\*\*  
 \* Класс функций полиномов любой степени  
 \*/*public class L4FunctionPoly implements IFunction {  
  
 */\*\*  
 \* Метод возвращает название вида функции  
 \* @return название вида функции  
 \*/* public String getFunctionName() {  
 return String.*format*("Полином степени %d",\_m);  
 }  
  
 */\*\*  
 \* Коэффициенты полинома a0, a1...  
 \*/* protected double[] \_a = null;  
 */\*\*  
 \* Степень полинома  
 \*/* private int \_m ;  
  
 */\*\*  
 \* Конструктор функции полинома  
 \* @param n - степень полинома  
 \*/* public L4FunctionPoly(int n) {  
 if(n <= 0) throw new IllegalArgumentException();  
 \_a = new double[n+1];  
 \_m = n;  
 }  
  
 */\*\*  
 \* Призна возникновения ошибки при расчете коэффициентов  
 \*/* protected boolean \_wasError = false;  
  
 */\*\*  
 \* getter признака наличия ошибки  
 \* @return наличие ошибки  
 \*/* public boolean getWasError() { return \_wasError; }  
  
 */\*\*  
 \* Метод вычисляет коэффициенты a0,a1,....,am  
 \* @param dots точки графика  
 \*/* public void CalcA(Dots dots){  
  
 double[][] matrix = new double[\_m+1][\_m+1];  
 double[] b = new double[\_m+1];  
  
 //Вычисляем массив B и матрицу  
 for(int i = 0; i <= \_m; i++){  
 double sum = 0.0;  
 for(int j = 0; j < dots.getN(); j++){  
 //sum += dots.getDotY(j)\*Math.pow(dots.getDotX(j),i);  
 sum += getY(dots,j) \* Math.*pow*(getX(dots,j),i);  
 }  
 b[i] = sum;  
 //Вычисляем строку матрицы  
 for(int col = 0; col <= \_m; col++){  
 sum = 0;  
 for(int j = 0; j < dots.getN(); j++){  
 //sum += Math.pow(dots.getDotX(j),col+i); // Обратить внимание на степень  
 sum += Math.*pow*(getX(dots,j),col+i);// Обратить внимание на степень  
 }  
 matrix[i][col] = sum;  
 }  
 }  
 Matrix matr = new Matrix(matrix,b);  
 \_a = matr.calcMatrix();  
 }  
  
 */\*\*  
 \* Метод возвращает то значение Y, которое должно быть помещено в СЛАУ для расчета по методу  
 \* наименьших квадратов  
 \* @param dots точки графика  
 \* @param index индекс точки  
 \* @return искомое значение  
 \*/* protected double getY(Dots dots,int index){  
 return dots.getDotY(index);  
 }  
  
 */\*\*  
 \* Метод возвращает то значение X, которое должно быть помещено в СЛАУ для расчета по методу  
 \* наименьших квадратов  
 \* @param dots точки графика  
 \* @param index индекс точки  
 \* @return искомое значение  
 \*/* protected double getX(Dots dots,int index){  
 return dots.getDotX(index);  
 }  
  
 @Override  
 public String toString() {  
 if(\_a == null) return "Полином не определен";  
 if(\_m == 1)  
 return String.*format*("%f + %f \* x", \_a[0], \_a[1]);  
 if(\_m == 2)  
 return String.*format*("%f + %f \* x + %f \*x\*x", \_a[0], \_a[1], \_a[2]);  
 if(\_m == 3)  
 return String.*format*("%f + %f \* x + %f \*x\*x + %f\*x\*x\*x", \_a[0], \_a[1], \_a[2], \_a[3]);  
 return String.*format*("Полином степени %d",\_m);  
 }  
  
 @Override  
 public double functionOf(double x) {  
 double sum = 0.0;  
 for(int i = 0; i <=\_m; i++) {  
 sum += \_a[i] \* Math.*pow*(x,i);  
 }  
 return sum;  
 }  
  
 @Override  
 public double f1Of(double x) {  
 return 0;  
 }  
}

*/\*\*  
 \* Класс степенной функции fi(x) = a\* pow(x,b)  
 \* Линеаризуем получем ln(fi(x)) = ln(a) + b \* ln(x)  
 \* Таким образом для полинома первой степени подставляем ln(y) и ln(x),  
 \* получаем a[0] - ln(a[0])  
 \* окончательно преобразуем a[0] = pow(e,a[0]) - это и будет a для нашей функциц  
 \* b = a[1]  
 \*/*public class L4FunctionPow extends L4FunctionPoly {  
 public L4FunctionPow() {super(1);}  
  
 @Override  
 public String getFunctionName() {  
 return "Степенная функция";  
 }  
  
 @Override  
 public void CalcA(Dots dots) {  
 super.CalcA(dots);  
 //Пересчитываем a[0]  
 \_a[0] = Math.*exp*(\_a[0]);  
 // Оставляем без изменений a[1]  
 }  
  
 @Override  
 protected double getY(Dots dots, int index) {  
 double vl = dots.getDotY(index);  
 if(vl <= 0) {  
 \_wasError = true;  
 throw new CalcErrorException("Для степенной функции значение Y в массиве точек не может быть <= 0");  
 }  
 return Math.*log*(vl);  
 }  
  
 @Override  
 protected double getX(Dots dots, int index) {  
 double vl = dots.getDotX(index);  
 if(vl <= 0) {  
 \_wasError = true;  
 throw new CalcErrorException("Для степенной функции значение X в массиве точек не может быть <= 0");  
 }  
 return Math.*log*(vl);  
 }  
  
 @Override  
 public String toString() {  
 return String.*format*("%f\*pow(x,%f)",\_a[0],\_a[1]);  
 }  
  
 @Override  
 public double functionOf(double x) {  
 return \_a[0]\*Math.*pow*(x, \_a[1]);  
 }  
}

**Результаты работы программы**

**Изображение выглядит как текст, снимок экрана, дисплей, линия

Автоматически созданное описание**

**Вывод**

В данной лабораторной работе была проведена аппроксимация табличной функции на интервале [-2;0), h=0,2 по методу наименьших квадратов. Были получены линейное и квадратичное приближения по 11 точкам.

Линейное приближение:

• Получена функция P(x) = 0,136235x - 0,10453164, которая является линейным приближением исходной функции.

• Среднеквадратичное отклонение линейного приближения составило 0,074011.

Квадратичное приближение:

• Получена функция P(x) = 0,21x^2 + 0,56x + 0,02, которая является квадратичным приближением исходной функции.

• Среднеквадратичное отклонение квадратичного приближения составило 0,011418.

Квадратичное приближение обладает значительно меньшим среднеквадратичным отклонением по сравнению с линейным. Это означает, что квадратичная функция лучше аппроксимирует исходную функцию на заданном интервале. Что подтверждает ее преимущество в случаях, когда исходная функция имеет нелинейный характер.