**Министерство науки и высшего образования РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра ИС**

отчет

**по лабораторной работе №4**

**по дисциплине «Конструирование программ»**

Тема: Аппроксимация функции по методу наименьших квадратов

|  |  |  |
| --- | --- | --- |
| Студент гр. 8363 |  | Нерсисян А.С. |
| Преподаватель |  | Копыльцов А.В. |

Санкт-Петербург

2020

**Цель работы.**

Заработать навыки нахождения явного функционального зависимости между двумя величинами и полученными в результате измерений. Построенить многочлен наилучшего среднего квадратического приближения по методу наименьших квадратов.

**Основные теоретические положения.**

* 1. **Постановка задачи и вывод формул метода наименьших квадратов**

Задача наименьших квадратов возникает в самых различных областях науки и техники, например, к ней приходят при статистической обработке экспериментальных данных. Пусть функция  задана таблицей приближенных значений , полученных с ошибками  Предположим, что для аппроксимации функции  используется линейная модель:   где  - заданные базисные функции,  - параметры модели, являющиеся одновременно коэффициентами обобщенного многочлена. Часто используется одна из наиболее простых моделей  - полиномиальная модель.

В случае, когда уровень неопределенности исходных данных высок, нет смысла требовать точного совпадения значений обобщенного многочлена  в точках  с заданными значениями , то есть использовать интерполяцию. Кроме того, при интерполяции происходит повторение ошибок наблюдений, в то время как при обработке экспериментальных данных желательно сглаживание ошибок. Тем не менее нужно потребовать, чтобы



Эта же система в матричной форме имеет вид 

Существуют разные дополнительные критерии, позволяющие решить эту систему, так как в общем случае при  она, вообще говоря, несовместна. Выбор , позволяющий наилучшим образом удовлетворить (3.1.2) в методе наименьших квадратов, состоит в том минимизируется среднее квадратическое уклонение



Итак, линейная задача метода наименьших квадратов состоит в следующем. Надо найти обобщенный многочлен , для которого среднеквадратическое уклонение  Этот многочлен называется *многочленом наилучшего среднего квадратического приближения.* Так как набор функций  всегда заранее определен, задача заключается в нахождении вектора  при условии  Для решения нашей задачи воспользуемся общим приемом дифференциального исчисления, а именно выпишем необходимые условия экстремума функции нескольких переменных (приравняем частные производные нулю):



Тогда получим  Изменим в первом слагаемом порядок суммирования:



Уравнение называется нормальной системой метода наименьших квадратов.

Если вернуться к обозначениям формулы, то, как нетрудно видеть, систему (3.1.5) можно записать в виде



Матрица  называется матрицей Грама. Если еще ввести вектор , то система (3.1.6) перепишется в виде  - система линейных уравнений относительно вектора . Можно показать, что если среди точек  нет совпадающих и , то определитель системы (3.1.6) отличен от нуля, и, следовательно, эта система имеет единственное решение:  Обобщенный полином с такими коэффициентами будет обладать минимальным средним квадратическим отклонением .

Если , то обобщенный многочлен, если система функций  степенная, совпадает с полиномом Лагранжа для системы точек , причем  При  построение такого точного интерполяционного многочлена невозможно. Таким образом, аппроксимация функций представляет собой более общий процесс, чем интерполирование.

Если , то нормальная система (3.1.5) принимает следующий вид:



Запишем систему (3.1.7) в развернутом виде в двух наиболее простых случаях при  и  В случае, когда приближение осуществляется многочленом первой степени , уравнения метода наименьших квадратов имеют следующий вид:









- нормальная система для  в развернутом виде. Пусть теперь  Аналогично получим 







- нормальная система для  в развернутом виде для квадратичного сглаживания.

Метод вычисления параметров  с помощью решения нормальной системы кажется весьма привлекательным. Действительно, задача сводится к стандартной системе линейных алгебраический уравнений с квадратной матрицей. Однако вычислительная практика показывает, что без специального выбора базисных функций  уже при  нормальная система обычно оказывается плохо обусловленной. Причина в том, что система базисных функций, будучи формально независимой, на практике часто близка к линейно зависимой. Особенно этим «грешит» система степенных функций , широко применяемая при аппроксимации алгебраическими многочленами. Лучший результат получается, если использовать систему ортогональных на отрезке  функций. Пример такой системы на  дает система многочленов Чебышева .

В настоящее время в вычислительной практике нормальная система, как правило, не используется. Применяются другие, более надежные методы, например метод сингулярного разложения матрицы .

**Экспериментальные результаты.**

**Задание № 1**

По методу наименьших квадратов аппроксимировать таблично заданную функцию многочленом наилучшего среднеквадратического приближения .

|  |  |
| --- | --- |
|  |  |
| Номера варианта |
| 11 |
| 0.1 | 8.15 |
| 0.3 | 8.41 |
| 0.5 | 8.58 |
| 0.7 | 8.84 |
| 0.9 | 9.28 |
| 1.1 | 9.46 |
| 1.3 | 10.02 |
| 1.5 | 10.11 |
| 1.7 | 10.61 |
| 1.9 | 11.03 |
| 2.1 | 11.34 |
| 2.3 | 11.86 |
| 2.5 | 12.33 |
| 2.7 | 12.81 |
| 2.9 | 13.21 |
| 3.1 | 13.67 |
| 3.3 | 14.23 |
| 3.5 | 14.68 |
| 3.7 | 15.35 |
| 3.9 | 15.93 |

**Дано:** Вариант 11

**Обработка результатов эксперимента.**

**Задание № 1. решение:**









Функции  возвращают число столбцов и строк матрицы ,  соответственно наименьшее и наибольшее значение элементов в ,  - число элементов в векторе ,  - индекс последнего элемента в векторе  с учетом значения переменной .

Построим линейную и квадратичную модель по формулам (3.1.8) и (3.1.9). Для этого вычислим следующие величины. Введем еще одну предопределенную переменную пакета Mathcad . Она определяет допустимую погрешность для различных алгоритмов аппроксимации, интегрирования, решения уравнений и так далее. По умолчанию . Вычислим следующие величины:



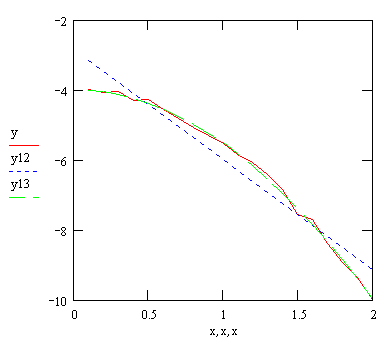












Коэффициенты системы нормальных уравнений линейной модели, то есть системы (3.1.8), находятся в матрице , коэффициенты квадратичной модели (3.1.9) - в матрице . Решение обеих систем линейных уравнений произведено с помощью обратной матрицы. Вектор  содержит коэффициенты линейной модели, вектор  - квадратичной.

Далее вычисляются невязки по обеим моделям и находятся средние квадратические ошибки . Видно, что исходным данным хорошо удовлетворяет квадратичная модель  Этот факт отчетливо виден и на приведенном графике.

Mathcad не назывался бы математическим пакетом, если бы не умел решать алгебраические системы различными, в том числе и более эффективными способами. Одним из таких способов является конструкция **Given – Find**. Это две команды: **Given** (Дано) и **Find** (Найти). Сначала задается какое-нибудь начальное приближение, например, для квадратичной модели



затем за ключевым словом **Given** нужно записать анализируемую систему, связывая левые и правые части уравнений знаком «эквивалентно» (жирным знаком «равно» из панели равенств и отношений или же нажимая сразу обе клавиши ), после этого должно идти второе ключевое слово **Find**. Эта функция возвращает решение анализируемой системы:



Неудобство применения пары **Given – Find** в том, что решаемая система уравнений должна быть записана в скалярной форме. Вместо функции **Find** можно использовать пару **Given – MinErr**. Функция  дает решение системы уравнений, которое приводит к минимальным невязкам. Число неизвестных системы должно быть равно числу аргументов функции **MinErr**. В нашем случае

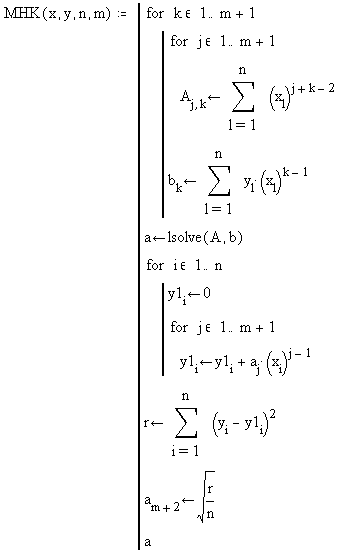


Наконец, для решения линейных систем алгебраических уравнений можно использовать встроенную функцию **lsolve**. Она возвращает вектор решения системы, записанной в матричном виде:



Заметим, что функцию **lsolve** можно использовать в программируемых конструкциях, тогда как пары **Given – Find** и **Given – MinErr** этого не допускают.

Приведем в заключение подпрограмму, реализующую вычисления по формуле (3.1.7) в общем случае коэффициентов сглаживающего многочлена заданной степени. Все операторы этой подпрограммы легко отождествляются с той или иной частью формулы (3.1.7). Параметры подпрограммы: - вектора исходных данных,  - число точек сетки таблично заданной функции,  - требуемая степень сглаживающего многочлена. В результате работы подпрограмма МНК выдает вектор коэффициентов многочлена , записанных в следующем порядке: . Последняя  компонента вектора результата содержит среднюю квадратическую ошибку представления исходных табличных данных построенным сглаживающим многочленом:



Для нашего примера

Видно, что для исходной таблично заданной функции многочленом наилучшего приближения является уже полученный ранее многочлен второй степени

**Выводы.**

В ходе выполнения данной лабораторной была выпполнена aппроксимация функции по методу наименьших квадратов. Былы найдены явные функциональные зависимости между двумя величинами и полученными в результате измерений.

**ПРИЛОЖЕНИЕ А**

**Код программы**

#include <iostream>

#include <conio.h>

using namespace std;

double det(double \*line1, double \*line2, double \*line3, double \*line4, double \*line5, int deg)

{

if (deg == 1) return line1[0];

else

{

double result = 0;

int i = deg;

result += line1[0] \* det(line2 + 1, line3 + 1, line4 + 1, line5 + 1, NULL, deg - 1);

result -= line2[0] \* det(line1 + 1, line3 + 1, line4 + 1, line5 + 1, NULL, deg - 1);

i -= 2;

if (i == 0) return result;

result -= line3[0] \* det(line1 + 1, line2 + 1, line4 + 1, line5 + 1, NULL, deg - 1);

i--;

if (i == 0) return result;

result -= line4[0] \* det(line1 + 1, line2 + 1, line3 + 1, line5 + 1, NULL, deg - 1);

i--;

if (i == 0) return result;

result -= line5[0] \* det(line1 + 1, line2 + 1, line3 + 1, line4 + 1, NULL, deg - 1);

return result;

}

}

int main()

{

double a[13][21], temp, line[6][5], d, c[5]; // 8 иксов и 5 икс-игриков

do

{

for (int i = 0; i < 20; ++i)

{

cout << "x" << i << " = ";

cin >> a[0][i];

for (int j = 1; j < 8; ++j)

a[j][i] = a[j - 1][i] \* a[0][i];

cout << "y" << i << " = ";

cin >> a[8][i];

for (int j = 9; j < 13; ++j)

a[j][i] = a[j - 1][i] \* a[0][i];

}

for (int i = 0; i < 13; ++i)

{

temp = 0;

for (int j = 0; j < 20; ++j)

temp += a[i][j];

a[i][20] = temp;

}

line[0][0] = 20;

for (int i = 1; i < 5; ++i)

line[0][i] = a[i-1][20];

for (int i = 1; i < 5; ++i)

for (int j = 0; j < 5; ++j)

line[i][j] = a[i + j][20];

for (int i = 0; i < 5; ++i)

line[5][i] = a[i + 8][20];

for (int i = 2; i < 5; ++i)

{

d = det(line[0], line[1], line[2], line[3], line[4], i);

int j = i - 2;

c[0] = det(line[5], line[1], line[2], line[3], line[4], i) / d;

c[1] = det(line[0], line[5], line[2], line[3], line[4], i) / d;

if (j == 0) cout << "m = 1\n" << c[1] << "x+" << c[0] << endl;

else

{

c[2] = det(line[0], line[1], line[5], line[3], line[4], i) / d;

j--;

if (j == 0) cout << "m = 2\n"<< c[2] << "x^2+" << c[1] << "x+" << c[0] << endl;

else

{

c[3] = det(line[0], line[1], line[2], line[5], line[4], i) / d;

j--;

if (j == 0) cout << "m = 3\n" << c[3] << "x^3+" << c[2] << "x^2+" << c[1] << "x+" << c[0] << endl;

else

{

c[4] = det(line[0], line[1], line[2], line[3], line[5], i) / d;

j--;

if (j == 0) cout << "m = 4\n" << c[4] << "x^4+" << c[3] << "x^3+" << c[2] << "x^2+" << c[1] << "x+" << c[0] << endl;

}

}

}

}

temp = \_getch();

} while (temp != 27);

return 0;

}

Протокол

1. Задали исходные данные задачи в виде массива данных.
2. По методу наименьших квадратов аппроксимировали таблично заданную функцию многочленом наилучшего среднеквадратического приближения.