Лабораторная работа № 7

студента группы ИТз-221

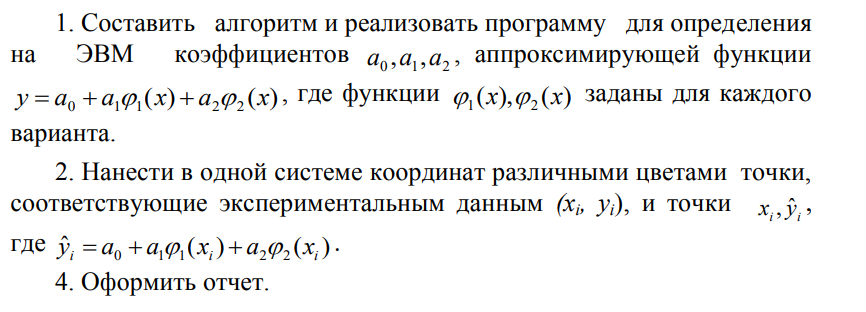
Дмитриева Дмитрия Анатольевича

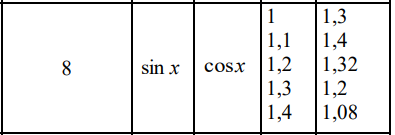
*Выполнение: \_\_\_\_\_\_\_\_\_ Защита: \_\_\_\_\_\_\_\_\_\_\_*

Аппроксимация функций методом наименьших квадратов

*Цель работы*: приобрести навыки при обработке экспериментальных данных; алгоритмизация метода наименьших квадратов; программирование метода наименьших квадратов.

**Содержание работы**

****

****

**Ход работы:**

***Вариант - 8***

1. Изучил теоретический материал.
2. Создал исходный модуль программы на языке высокого уровня C# (Приложение А), реализующий приближенное вычисление и определения на ЭВМ коэффициентов a0, a1, a2 заданной аппроксимирующей функцией. (рис. 1).

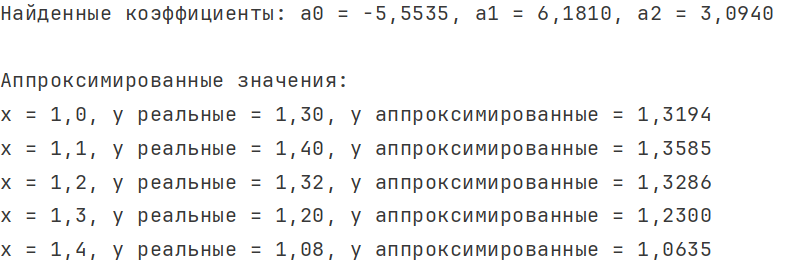


Рисунок 1 – Результат выполнения программы

1. Создал блок-схему будущей программы (Приложение Б)
2. Нанес в одной системе координат различными цветами точки, соответствующие экспериментальным данным (рис. 2). Реальные данные показаны синим цветом, а аппроксимированные оранжевым.

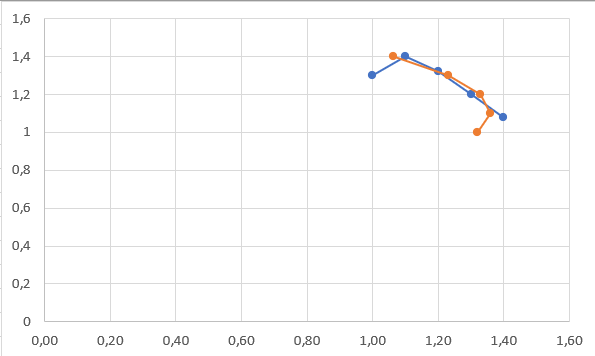


Рисунок 2 – График экспериментальных данных

**Контрольные вопросы**:

1. Функции sin (x) и cos(x) являются периодическими с периодом 2π, что означает их повторяемость через равные интервалы. Они также непрерывны и ограничены, принимая значения только в пределах от -1 до 1. Кроме того, синус — нечетная функция, а косинус — четная, что определяет их симметрии относительно начала координат и оси ординат соответственно.
2. Система уравнений, используемая для определения коэффициентов многочлена a0, a1, a2​ в методе наименьших квадратов, строится на основе нормальных уравнений. Эти уравнения получаются из условия минимизации суммы квадратов отклонений между реальными значениями функции и значениями, рассчитанными с помощью аппроксимирующего многочлена. В результате решения данной системы находится такой набор коэффициентов, который обеспечивает наилучшее приближение заданных данных в смысле метода наименьших квадратов.
3. В методе наименьших квадратов невязка — это разница между фактическими значениями и предсказанными моделью. Метод стремится минимизировать сумму квадратов этих отклонений, чтобы обеспечить наилучшую аппроксимацию данных.

**Вывод**: приобрел навыки при обработке экспериментальных данных, алгоритмизации метода наименьших квадратов; научился программировать метод наименьших квадратов.

ПРИЛОЖЕНИЕ А

Код приложения

class Program

{

static void Main()

{

double[] x = [1.0, 1.1, 1.2, 1.3, 1.4];

double[] y = [1.3, 1.4, 1.32, 1.2, 1.08];

double[] functionX1 = new double[x.Length];

double[] functionX2 = new double[x.Length];

for (int i = 0; i < x.Length; i++)

{

functionX1[i] = Math.Sin(x[i]);

functionX2[i] = Math.Cos(x[i]);

}

double sumFunctionX1 = 0, sumFunctionX2 = 0;

for (int i = 0; i < x.Length; i++)

{

sumFunctionX1 += functionX1[i];

sumFunctionX2 += functionX2[i];

}

double sumFunctionX1Squared = 0, sumFunctionX1FunctionX2 = 0, sumFunctionX2Squared = 0;

for (int i = 0; i < x.Length; i++)

{

sumFunctionX1Squared += functionX1[i] \* functionX1[i];

sumFunctionX1FunctionX2 += functionX1[i] \* functionX2[i];

sumFunctionX2Squared += functionX2[i] \* functionX2[i];

}

double[,] matrixA = {

{ x.Length, sumFunctionX1, sumFunctionX2 },

{ sumFunctionX1, sumFunctionX1Squared, sumFunctionX1FunctionX2 },

{ sumFunctionX2, sumFunctionX1FunctionX2, sumFunctionX2Squared }

};

double[] matrixB = new double[3];

for (int i = 0; i < x.Length; i++)

{

matrixB[0] += y[i];

matrixB[1] += functionX1[i] \* y[i];

matrixB[2] += functionX2[i] \* y[i];

}

double detA = Determinant(matrixA);

double[] coefficients = new double[3];

for (int i = 0; i < 3; i++)

{

double[,] matrixTemp = (double[,])matrixA.Clone();

for (int j = 0; j < 3; j++)

{

matrixTemp[j, i] = matrixB[j];

}

coefficients[i] = Determinant(matrixTemp) / detA;

}

Console.WriteLine($"Вычисленные коэффициенты: a0 = {coefficients[0]:F4}, a1 = {coefficients[1]:F4}, a2 = {coefficients[2]:F4}");

Console.WriteLine("\nАппроксимированные значения:");

for (int i = 0; i < x.Length; i++)

{

double yApprox = coefficients[0] + coefficients[1] \* functionX1[i] + coefficients[2] \* functionX2[i];

Console.WriteLine($"x = {x[i]:F1}, y реальные = {y[i]:F2}, y аппроксимированные = {yApprox:F4}");

}

}

static double Determinant(double[,] matrix)

{

return matrix[0, 0] \* (matrix[1, 1] \* matrix[2, 2] - matrix[1, 2] \* matrix[2, 1])

- matrix[0, 1] \* (matrix[1, 0] \* matrix[2, 2] - matrix[1, 2] \* matrix[2, 0])

+ matrix[0, 2] \* (matrix[1, 0] \* matrix[2, 1] - matrix[1, 1] \* matrix[2, 0]);

}

}

ПРИЛОЖЕНИ Б

Блок схема

