МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«Национальный исследовательский ядерный университет «МИФИ»
(НИЯУ МИФИ)
Институт Интеллектуальных Кибернетических Систем
Кафедра Кибернетики

Лабораторная работа №4:

По курсу «Численные методы»

Вариант 16

Работу выполнил: студент группы Б17-511: Чудновец И.В.

Проверил: Саманчук В.Н.

Москва 2019

Постановка задачи

Аппроксимировать таблично заданную функцию по методу наименьших квадратов, используя полиномы по девятый порядок включительно.

X	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Y	5	6	8	10	12	13	12	10	8	10	8	11	7	9	11	10	9	12	11	6

Методика решения

Для решения задачи была написана программа на языке Python, в которой реализован алгоритм аппроксимирования таблично заданной функции, используя полиномы по девятый порядок включительно.

Теоретическая справка

Суть метода наименьших квадратов в выборе независимой системы функций

$f_i(x), i = 0, 1, \cdots, m$ и подборе неизвестных коэффициентов $a_0, a_1, \cdots, a_m$ с целью минимизации функции:

$$F(a_0, a_1, \cdots, a_m) = \sum_{j=1}^{n}(y_j - \sum_{i=0}^{m} a_i * f_i(x))^2 \to \min$$

Для случая аппроксимации полиномами по 9-ый порядок включительно:

Нужно решить СЛАУ 10-го порядка относительно коэффициентов $a_0, a_1, \cdots, a_9$:

$$\begin{cases} a_0 * \sum_{i=1}^{20} {x_i}^{18} + a_1 * \sum_{i=1}^{20} {x_i}^{17} + \cdots + a_9 * \sum_{i=1}^{20} {x_i}^{9} = \sum_{i=1}^{20} y_i * {x_i}^{9} \\ a_0 * \sum_{i=1}^{20} {x_i}^{17} + a_1 * \sum_{i=1}^{20} {x_i}^{16} + \cdots + a_9 * \sum_{i=1}^{20} {x_i}^{8} = \sum_{i=1}^{20} y_i * {x_i}^{8} \\ a_0 * \sum_{i=1}^{20} {x_i}^{16} + a_1 * \sum_{i=1}^{20} {x_i}^{15} + \cdots + a_9 * \sum_{i=1}^{20} {x_i}^{7} = \sum_{i=1}^{20} y_i * {x_i}^{7} \\ a_0 * \sum_{i=1}^{20} {x_i}^{15} + a_1 * \sum_{i=1}^{20} {x_i}^{14} + \cdots + a_9 * \sum_{i=1}^{20} {x_i}^{6} = \sum_{i=1}^{20} y_i * {x_i}^{6} \\ a_0 * \sum_{i=1}^{20} {x_i}^{14} + a_1 * \sum_{i=1}^{20} {x_i}^{13} + \cdots + a_9 * \sum_{i=1}^{20} {x_i}^{5} = \sum_{i=1}^{20} y_i * {x_i}^{5} \\ a_0 * \sum_{i=1}^{20} {x_i}^{13} + a_1 * \sum_{i=1}^{20} {x_i}^{12} + \cdots + a_9 * \sum_{i=1}^{20} {x_i}^{4} = \sum_{i=1}^{20} y_i * {x_i}^{4} \\ a_0 * \sum_{i=1}^{20} {x_i}^{12} + a_1 * \sum_{i=1}^{20} {x_i}^{11} + \cdots + a_9 * \sum_{i=1}^{20} {x_i}^{3} = \sum_{i=1}^{20} y_i * {x_i}^{3} \\ a_0 * \sum_{i=1}^{20} {x_i}^{11} + a_1 * \sum_{i=1}^{20} {x_i}^{10} + \cdots + a_9 * \sum_{i=1}^{20} {x_i}^{2} = \sum_{i=1}^{20} y_i * {x_i}^{2} \\ a_0 * \sum_{i=1}^{20} {x_i}^{10} + a_1 * \sum_{i=1}^{20} {x_i}^{9} + \cdots + a_9 * \sum_{i=1}^{20} {x_i}^{1} = \sum_{i=1}^{20} y_i * {x_i}^{1} \\ a_0 * \sum_{i=1}^{20} {x_i}^{9} + a_1 * \sum_{i=1}^{20} {x_i}^{8} + \cdots + a_9 * \sum_{i=1}^{20} 1 = \sum_{i=1}^{20} y_i \end{cases}$$

Сумма квадратов отклонений между узловыми значениями функции и аппроксимирующей кривой вычисляются по формуле:

$$\sum_{j=1}^{20}(y_j - \sum_{n=0}^{9} a_i * {x_j}^{n})^2$$

Решение задачи

least_squares.py

```
# -*- coding: utf-8 -*-
from sympy import *
import matplotlib.pyplot as plt
import numpy as np
```

```
init_printing()


class LeastSquares:
    @staticmethod
    def create_polinom(x_, x_list, y_list, n):
        # n - порядок полинома
        if len(x_list) != len(y_list):
            return
        x_buffer = [1 for _ in range(len(x_list))]
        y_buffer = [i for i in y_list]

        A = zeros(n + 1)
        b = Matrix([0 for _ in range(n + 1)])

        # последняя строка матрицы A и матрица b
        for j in range(n, -1, -1):
            b[j] = sum(y_buffer)
            A[n, j] = sum(x_buffer)
            x_buffer = [x_buffer[i] * x_list[i] for i in range(len(x_list))]
            y_buffer = [y_buffer[i] * x_list[i] for i in range(len(x_list))]

        # n первых строк матрицы A
        for i in range(n - 1, -1, -1):
            for j in range(n, 0, -1):
                A[i, j] = A[i + 1, j - 1]
            A[i, 0] = sum(x_buffer)
            x_buffer = [x_buffer[j] * x_list[j] for j in range(len(x_list))]

        print("Matrix A:")
        display(A)
        print("Matrix b:")
        display(b)

        # Решение СЛАУ
        solution = tuple(linsolve((A, b)))[0]
        coeffs = [N(i, 5) for i in solution]

        # Формирование многочлена n-ой стпени
        x_current = 1
        res = 0
        for i in range(len(coeffs) - 1, -1, -1):
            res += x_current * coeffs[i]
            x_current *= x_
        s = 0
        for i, x_i in enumerate(x_list):
            s += (y_list[i] - res.subs(x_, x_i)) ** 2
        print("Сумма квадратов отклонений между узловыми значениями функции и
аппроксимирующей кривой:", s)
        return res, coeffs

    @staticmethod
    def split_intervals(x_list, accuracy):
        x_values = []
        for i in range(len(x_list) - 1):
            x_i = x_list[i]
            x_values.append(x_i)
            step = (x_list[i + 1] - x_i) / (accuracy + 1)
            for j in range(1, accuracy + 1):
                x_values.append(x_i + step * j)
        x_values.append(x_list[-1])
        return x_values
```

```
    @staticmethod
    def plot(x_list, coeffs):
        y_list = [np.polyval(coeffs, x_i) for x_i in x_list]
        plt.plot(x_list, y_list)
```

test_approximation.ipynb

```
from least_squares import LeastSquares
import matplotlib.pyplot as plt
from sympy import *
import pylab
pylab.rcParams['figure.figsize'] = (15.0, 10.0)
plt.rcParams.update({'font.size': 22})

init_printing()

var('x')
---
y_20 = [5, 6, 8, 10, 12, 13, 12, 10, 8, 10, 8, 11, 7, 9, 11, 10, 9, 12, 11, 6]
x_20 = [x_i for x_i in range(1, len(y_20) + 1)]
display(x_20)
display(y_20)
---
polinom_9, coeffs = LeastSquares.create_polinom(x, x_20, y_20, 9)
x_list = LeastSquares.split_intervals(x_20, 10)
LeastSquares.plot(x_list, coeffs)
plt.scatter(x_20, y_20, color='red', s=100)
plt.plot(x_20, y_20)
```

Условные обозначения:

--- - означает, что код в Jupyter Notebook разделён в разные ячейки

Результат работы

[5, 6, 8, 10, 12, 13, 12, 10, 8, 10, 8, 11, 7, 9, 11, 10, 9, 12, 11, 6]

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20]

Matrix A:

426453788542828686799730	2203639771002776930922 0	1142003663611187899466	59376590676022063500	3098689489300027490
2203639771002776930922 0	1142003663611187899466	59376590676022063500	3098689489300027490	162401629714694580
1142003663611187899466	59376590676022063500	3098689489300027490	162401629714694580	8553403807182266
59376590676022063500	3098689489300027490	162401629714694580	8553403807182266	453084917113500
3098689489300027490	162401629714694580	8553403807182266	453084917113500	24163571680850
162401629714694580	8553403807182266	453084917113500	24163571680850	1299155279940
8553403807182266	453084917113500	24163571680850	1299155279940	70540730666
453084917113500	24163571680850	1299155279940	70540730666	3877286700
24163571680850	1299155279940	70540730666	3877286700	216455810
1299155279940	70540730666	3877286700	216455810	12333300

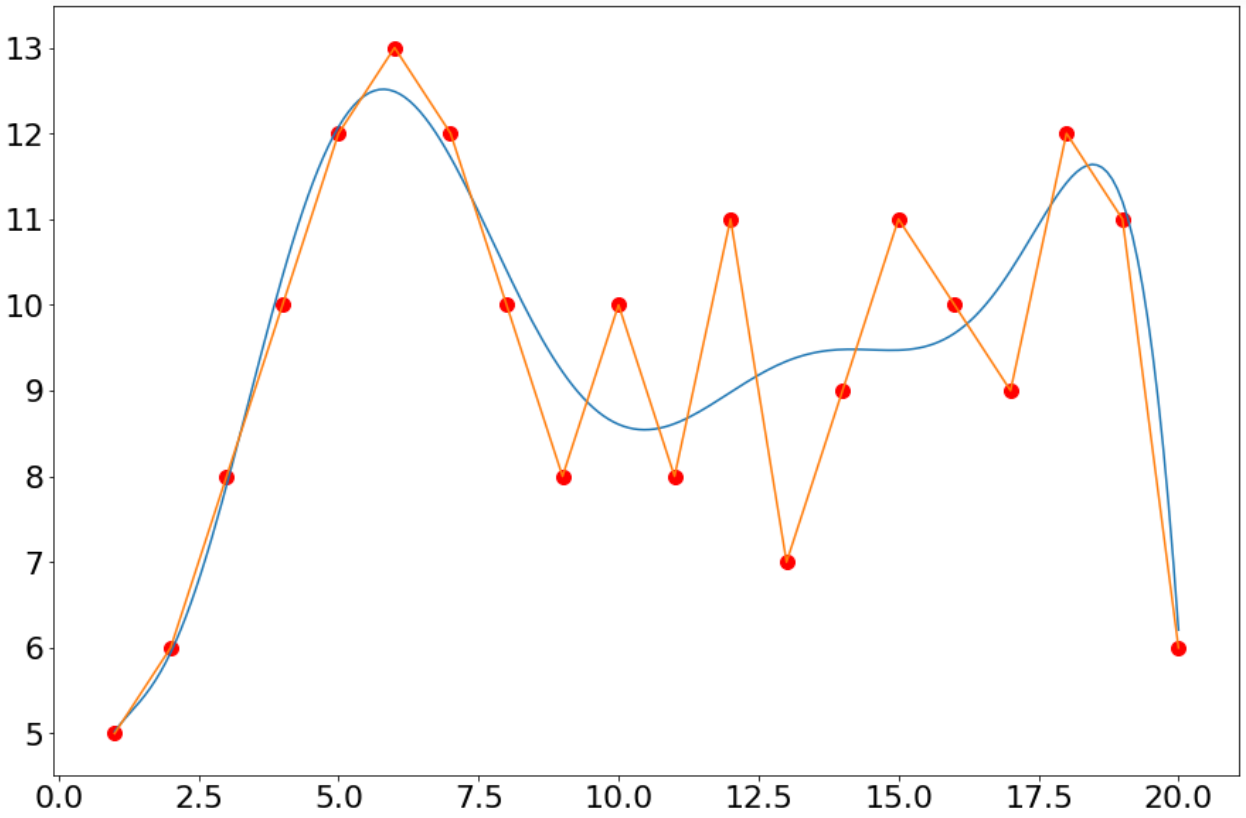
$$\begin{bmatrix}
16240162971469458 0 & 8553403807182266 & 453084917113500 & 24163571680850 & 1299155279940 \\
8553403807182266 & 453084917113500 & 24163571680850 & 1299155279940 & 70540730666 \\
453084917113500 & 24163571680850 & 1299155279940 & 70540730666 & 3877286700 \\
24163571680850 & 1299155279940 & 70540730666 & 3877286700 & 216455810 \\
1299155279940 & 70540730666 & 3877286700 & 216455810 & 12333300 \\
70540730666 & 3877286700 & 216455810 & 12333300 & 722666 \\
3877286700 & 216455810 & 12333300 & 722666 & 44100 \\
216455810 & 12333300 & 722666 & 44100 & 2870 \\
12333300 & 722666 & 44100 & 2870 & 210 \\
722666 & 44100 & 2870 & 210 & 20
\end{bmatrix}$$

Matrix b:

$$\begin{bmatrix}
11530123865015 \\
633627448859 \\
35227970231 \\
1987656299 \\
114347255 \\
6757979 \\
415751 \\
27299 \\
2015 \\
188
\end{bmatrix}$$

Сумма квадратов отклонений между узловыми значениями функции и аппроксимирующей кривой: 20.655

$$7.2558 \cdot 10^{-8}x^9 - 7.5469 \cdot 10^{-6}x^8 + 0.00032398x^7 - 0.0074375x^6 + 0.098527x^5 - 0.75463x^4 + 3.1597x^3 - 6.4954x^2 + 6.9809x + 2.0418$$



Заключение

В работе требовалось аппроксимировать таблично заданную функцию по методу наименьших квадратов, используя полиномы по девятый порядок включительно. Для решения данной задачи была написана программа. Результат представлен в виде скриншота выше.