НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
"МОСКОВСКИЙ ЭНЕРГЕТИЧЕСКИЙ ИНСТИТУТ"

Кафедра математического и компьютерного моделирования

Численные методы
Отчет по лабораторной работе №4
"Приближение функций."
Вариант 33

Студент:	Волков Павел Евгеньевич
Преподаватель:	Амосова Ольга Алексеевна
Группа:	А-14-19

Москва
2021

Задача 4.1

Постановка задачи

В таблице приведены данные о численности населения некоторых крупнейших стран мира по годам с 1950 -2000 г.г. Заполнить последние два столбца таблицы (взять сведения из интернета). На основе этих данных для конкретного варианта построить наилучший многочлен по МНК. Найти численность населения страны в 2019 году и сравнить полученное значение с актуальным значением (взять из интернета). Решить ту же задачу на основе интерполяционного многочлена. То есть построить интерполяционный многочлен по значениям с 1950-2020 г.г. Вычислить значение для 2019 года и сравнить с актуальными данными.

N	Страна	1950	1960	1970	1980	1990	2000	2010	2020
4.1.33	Чехия	8.9	9.6	9.9	10.3	10.4	10.3	10.5	10.7

Теоретический материал

Требуется найти многочлен P_m заданной степени $m, (m < n)$ такой, чтобы величина среднеквадратичного отклонения (СКО)

$$\sigma(P_m, f) = \sqrt{\frac{1}{n+1}\sum_{i=0}^{n}(P_m(x_i) - f_i)^2}$$

была минимальной. Для нахождения этого минимума будем использовать условие экстремума функции нескольких переменных:

$$\frac{\partial \rho}{\partial a_k} = 0, k = 0{,}1 \dots m$$

Таким образом задача о нахождении многочлена степени m сводится к поиску решения следующей симметричной системы:

$$\begin{cases} s_0 a_0 + s_1 a_1 + s_2 a_2 + \cdots + s_m a_m = b_0 \\ s_1 a_0 + s_2 a_1 + s_3 a_2 + \cdots + s_{m+1} a_m = b_1 \\ \dots \\ s_m a_0 + s_{m+1} a_1 + s_{m+2} a_2 + \cdots + s_{2m} a_m = b_m \end{cases}$$

Где $s_k = \sum_{i=0}^{n} x_i^k \quad b_k = \sum_{i=0}^{n} f_i x_i^k$

Остается опытным путем определить, какую степень многочлена m необходимо взять, чтобы получить наилучшее приближение, так как сначала с ростом m среднеквадратичное отклонение сначала убывает, а затем начинает возрастать, причем при больших m нормальная система наименьших квадратов становится плохо обусловленной, а при $m = n$ многочлен совпадает с интерполяционным многочленом.

Решим ту же задачу с помощью интерполяционного многочлена, записанного в форме Лагранжа:

$$L_n(x) = \sum_{i=0}^{n} f_i \prod_{k=0}^{n} \frac{(x - x_k)}{(x_i - x_k)}, (i \neq k)$$

Решение

Прежде чем разработывать алгоритм решения данной задачи, сформулируем несколько требований к основной подпрограмме:

- На вход подпрограмме подается 2 массива - массив аргументов и массив значений
- На выход подпрограмма возвращает функцию, реализующую искомый многочлен
- Для решения задачи методом наименьших квадратов также необходима функция, вычисляющая среднеквадратичное отклонение.

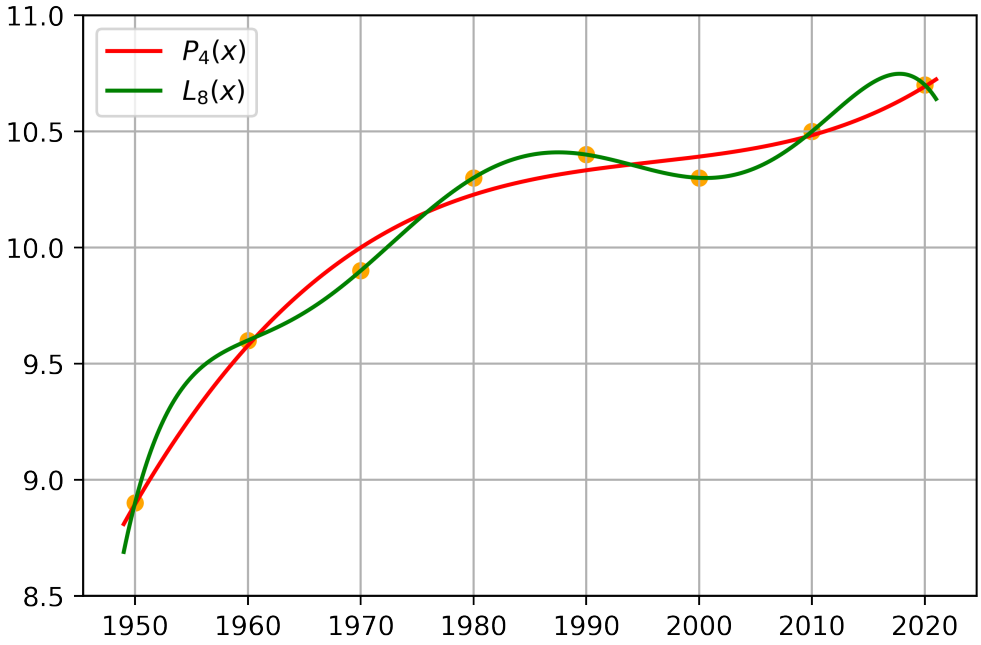
Теперь опишем алгоритм решения задачи по методу наименьших квадратов:

1. Опишем вспомогательные функции, позволяющие вычислить коэффициенты s_k и b_k нормальной системы наименьших квадратов.
2. Сформируем матрицу системы и решим ее с помощью встроенного метода библиотеки NumPy np.linalg.solve()
3. В качестве конечного результата сформируем функцию, реализующую искомый многочлен.
4. В конце необходимо "подобрать"степень многочлена, сравнивая среднеквадратичные отклонения для различных m.

Для построения интерполяционного многочлена в форме Лагранжа используем прямую формулу, то есть по определению.

Анализ результатов

Анализ среднеквадратичных отклонений показал, что наименьшее свое значение оно показывает при степени многочлена $m = 4$. Ниже представлены графики обоих многочленов.



Вычислив значение численности населения Чехии, и сравнив их с реальным значением получили следующие реузльтаты:

Метод наименьших квадратов: 10.664

Интерполяционный многочлен: 10.734

Реальное значение: 10.669

Метод наименьших квадратов дал более точное значение в связи с тем, что такие данные, как численность населения заданы с некоторой погрешностью, из-за чего полином, полученый с помощью МНК дает более точно представление о динамике численности населения.

Код программы

```
import numpy as np
import matplotlib.pyplot as plt

x = [1950, 1960, 1970, 1980, 1990, 2000, 2010, 2020]
y = [8.9, 9.6, 9.9, 10.3, 10.4, 10.3, 10.5, 10.7]

def Least_Squares(x: list, y: list, m: int):
    s = lambda k: sum([elem**k for elem in x])
    b = lambda k: sum([y[i]*x[i]**k for i in range(len(x))])
    NormalSystem = np.zeros((m+1, m+1))
    for i in range(m+1):
        for j in range(i, m+1):
            temp_coef = s(i + j)
            NormalSystem[i, j] = temp_coef
            NormalSystem[j, i] = temp_coef
    d = np.array([b(i) for i in range(m+1)])
    a = np.linalg.solve(NormalSystem, d)
    return a

def Average_Square_Deviation(Polynom, x, y):
    return (1.0/(len(x)) * sum([(Polynom(x[i]) - y[i])**2 for i in
        range(len(x))]))**0.5

devs = []
for m in range(1, 6):
    a = Least_Squares(x, y, m)
    Polynom = lambda x: sum([a[i]*x**i for i in range(m+1)])
    devs.append(Average_Square_Deviation(Polynom, x, y))
print(devs)#min(dev) = 0.060136. Polynom degree - 4

a = Least_Squares(x, y, 4)
Polynom = lambda x: sum([a[i]*x**i for i in range(5)])

def Lagrange_Polynom(arg):
    res = 0
    for i in range(len(x)):
        tmp = 1
        for k in range(len(x)):
            if i != k:
                tmp *= (arg - x[k]) * 1.0/(x[i] - x[k])
        res += y[i]*tmp
```

```
    return res

print("Least Squares method:", Polynom(2019))
print("Lagrange Polynom:", Lagrange_Polynom(2019))
print("Real value:", 10.669)
```

Задача 4.2

Постановка задачи

Дана функция $y = f(x)$. Приблизить функцию методом интерполяции, используя многочлен Лагранжа. Степень n подобрать таким образом, чтобы максимальная величина погрешности на отрезке $[a, b]$ не превышала заданной величины ε. Построить графики многочленов и графики погрешностей. Приблизить функцию методом интерполяции, указанным в индивидуальном варианте (Квадратичный сплайн с дополнительным условием $y'(a) = f'(a)$). Сравнить полученные результаты.

$$f(x) = x \sin(2 - x), [1, 4]$$
$$\varepsilon = 0.001$$

Теоретический материал

Так как имеется свобода выбора отрезков разбиения, то имеет смысл выбрать точки таким образом, чтобы погрешность интерполяции была минимальной. Для этого воспользуемся свойством наименее уклонятся от нуля на отрезке $[-1, 1]$ многочленов Чебышева. То есть в качестве узлов интерполяции возьмем нули многочлена Чебышева:

$$x_k = \frac{a+b}{2} + \frac{b-a}{2} \cos\left(\pi \frac{2k+1}{2n+2}\right), k = 0, 1 \ldots, n$$

Определение Сплайном степени m называется функция $S_m(x)$, обладающая следующими свойствами:

1. Функция $S_m(x)$ непрерывна на $[a, b]$ со своими производными.
2. На каждом отрезке $[x_i, x_{i+1}]$ функция $S_m(x)$ совпадает с некоторым алгебраическим многочленом $P_{m,i}(x)$ степени m.
3. $S_m(x_i) = y_i$, $i = 0, 1 \ldots n$

Определение Величина $R_n(x) = |f(x) - P_n(x)|$ называется остаточным членом интерполяции или погрешностью интерполяции.

Теорема Пусть функция $f(x)$ дифференцируема $(n+1)$ раз на отрезке $[a, b]$, содержащем узлы интерполяции x_i. Тогда для погрешности интерполяции в точке $x \in [a, b]$ справедлива оценка:

$$R_n(x) = |f(x) - P_n(x)| \leq \frac{M_{n+1}}{(n+1)!}|\omega_{n+1}(x)|$$

Где $M_{n+1} = max|f^{(n+1)}(x)|$, а $\omega_{n+1}(x) = (x - x_0)\dots(x - x_n)$

Решение

Коэффициенты многочленов сплайна будем искать следующим образом:

1. Запишем многочлены в форме $P_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)(x - x_{i+1})$.
2. Из условий интерполяции получаем: $a_i = f(x_i)$ а также $a_i + b_i(x_{i+1} - x_i) = a_{i+1} = f(x_{i+1})$, то есть $b_i = \dfrac{a_{i+1} - a_i}{x_{i+1} - x_i}$.
3. Коэффициенты c_i будем определять из условий непрерывности производной: $P_i'(x_{i+1}) = P_{i+1}'(x_{i+1})$. То есть: $b_i - 2c_i x_i = b_{i+1} - 2x_{i+2}c_{i+1}$. В конечном итоге получаем следующую формулу для c_{i+1}: $c_{i+1} = \dfrac{b_{i+1} - b_i}{2x_{i+2}} + \dfrac{x_i}{x_{i+2}}c_i$.

Анализ результатов

Код программы