Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ Информатика и системы управления (ИУ)

КАФЕДРА Программное обеспечение ЭВМ и информационные технологии (ИУ7)

ЛАБОРАТОРНАЯ РАБОТА №3
« Построение и программная реализация алгоритма сплайн-интерполяции табличных функций »

Студент группы ИУ7-41Б
Костев Дмитрий

Москва 2021

Цель работы

Получение навыков владения методами интерполяции таблично заданных функций с помощью кубических сплайнов.

Исходные данные

1. Таблица функции с количеством узлов N. Задать с помощью формулы $y = x^2$ в диапазоне [0..10] с шагом 1.

2. Значение аргумента x в первом интервале, например, при x=0.5 и в середине таблицы, например, при x= 5.5. Сравнить с точным значением.

Код программы

```
# Лабораторная работа №3

import numpy

def f(x):
    return x * x

def values():
    print("Enter start, end and step separated by a space:")
    start, stop, step = map(float, input().split())
    x_values = numpy.arange(start, stop + 0.1, step)
    y_values = [f(x) for x in x_values]
    return x_values, y_values

def print_table(x_values, y_values):
    print(" {:^10} {:^10}".format("X", "Y"))
    for i in range(len(x_values)):
        print("|{:^10} |{:^10}|".format(
            round(x_values[i], 4), round(y_values[i], 4)))

def find_min(x_values, x):
    for i in range(len(x_values) - 1):
        if x > x_values[i] and x < x_values[i + 1]:
            return i

# Функция интерполяции
def interpolation(x_values, y_values, x):
    n = len(x_values)
    index = find_min(x_values, x)
    if not index:
        print("The value is either present in the table or is
outside of it")
        return

    h = [0]
    for i in range(1, n):
        h.append(x_values[i] - x_values[i - 1])

    A = [0, 0]
    for i in range(2, n):
        A.append(h[i - 1])

    B = [0, 0]
    for i in range(2, n):
```

```
        B.append(-2 * (h[i - 1] + h[i]))

    D = [0, 0]
    for i in range(2, n):
        D.append(h[i])

    F = [0, 0]
    for i in range(2, n):
        F.append(-3 * ((y_values[i] - y_values[i - 1]) / h[i]
-
                               (y_values[i - 1] - y_values[i - 2]) /
h[i - 1]))

    # Прямой ход.
    E = [0 for i in range(n + 1)]
    η = [0 for i in range(n + 1)]
    for i in range(2, n):
        E[i + 1] = D[i] / (B[i] - A[i] * E[i])
        η[i + 1] = (A[i] * η[i] + F[i]) / (B[i] - A[i] *
E[i])

    # Обратный ход.
    c = [0 for i in range(n + 1)]
    for i in range(n - 2, -1, -1):
        c[i] = E[i + 1] * c[i + 1] + η[i + 1]

    a = [0]
    for i in range(1, n):
        a.append(y_values[i - 1])

    b = [0]
    for i in range(1, n):
        b.append((y_values[i] - y_values[i - 1]) /
                 h[i] - h[i] / 3 * (c[i + 1] + 2 * c[i]))

    d = [0]
    for i in range(1, n):
        d.append((c[i + 1] - c[i]) / (3 * h[i]))

    diff = x - x_values[index - 1]
    return a[index] + b[index] * (diff) + c[index] * ((diff)
** 2) + d[index] * ((diff) ** 3)
```

```
def main():
    x_values, y_values = values()

    print_table(x_values, y_values)

    x = float(input("Enter x: "))

    result = interpolation(x_values, y_values, x)
    if not result:
        return

    print("f({}) = {}".format(x, f(x)))
    print("Interpolation spline: {}".format(round(result,
4)))
    print("Error = {}".format(abs(round(f(x) - result, 4))))


if __name__ == "__main__":
    main()
```

Результат работы программы

Сравнение точности двух методов:

1. Интерполяция кубическими сплайнами

x = 0.5	x = 5.5
0.408493	30.250253

2. Интерполяция полиномом Ньютона 3-й степени

x = 0.5	x = 5.5
0.25	30.25

Ответы на вопросы

1. Получить выражения для коэффициентов кубического сплайна, построенного на двух точках.

   Пусть заданные точки имеют координаты (x1, y1) и (x2, y2), x1 != x2.
   Тогда для f(x) = a + b(x - x1) + c(x - x1)^2 + d(x - x1)^3 имеем:
   Условия для значения в узлах: f(x1) = y1, f(x2) = y2
   a = y1
   a + b(x - x1) + c(x - x1)^2 + d(x - x1)^3 = y2
   Дополнительные условия для концов могут быть выбраны произвольно. Для простоты, приравняем вторые производные к нулю для получения функции с минимальной кривизной: f''(x1) = 0, f''(x2) = 0
   2c = 0
   2c + 6d(x2 - x1) = 0
   Данный набор из 4 уравнений позволяет единственным образом определить все необходимые коэффициенты.
   Очевидно, что c = 0 и d = 0 =>
   => a + b(x2 - x1) = y2 => a = y1, b = (y2 - y1)/(x2 - x1) =>
   => f(x) = y1 + ((y2 - y1) * (x - x1))/(x2 - x1)
   Получили, что функция выродилась в уравнение прямой, проходящей через две точки

2. Выписать все условия для определения коэффициентов сплайна, построенного на 3-х точках

   Сплайн по трём точкам будет состоять из двух функций, поэтому всего неизвестных коэффициентов будет 8.
   Условия:
   1, 2) $a_i = y_i$
   3, 4) $a_i + b_i(x_{i+1} - x_i) + c_i(x_{i+1} - x_i)^2 + d(x_{i+1} - x_i)^3 = y_{i+1}$, $f_1'(x_2) = f_2'(x_2)$
   5) $b_1 + 2c_1(x_2 - x_1) + 3d_1(x_2 - x_1)^2 = b_2$, $f_1''(x_2) = f_2''(x_2)$
   6) $c_1 + 3d_1(x_2 - x_1)^2 = c_2$
   7) $f_1''(x_1) = 0 \Rightarrow c_1 = 0$
   8) $f_2''(x_3) = 0 \Rightarrow c_2 + 3d_2(x_3 - x_2) = 0$

3. Определить начальные значения прогоночных коэффициентов, если принять, что для коэффициентов сплайна справедливо $C_1=C_2$.
   Прогоночные коэффициенты удовлетворяют условиям:
   $c_i = E_{i+1}c_{i+1} + n_{i+1}$
   Поэтому, для $c_1 = c_2$, начальные значения пригоночных коэффициентов будут равны соответственно:
   $E_2 = 1$, $n_1 = 0$

4. Написать формулу для определения последнего коэффициента сплайна C_N , чтобы можно было выполнить обратный ход метода прогонки, если в качестве граничного условия задано $kC_{N-1}+mC_N=p$, где k,m и p - заданные числа.
   $c_i = E_{i+1}c_{i+1} + n_{i+1} \Rightarrow c_{N-1}= E_Nc_N + n_N$
   $kc_{N-1} + mc_N = p \Rightarrow k(E_Nc_N + n_N) + mc_N = p$
   $c_N = (p - kn_N)/(kE_N + m)$