Министерство науки и высшего образования 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 »

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

Лабораторная работа № 3

Тема: Построение и программная реализация алгоритма сплайн-интерполяции табличных функций.

Студент: Чаушев Александър

Группа: ИУ7-46Б

Оценка (баллы) _______________

Преподаватель : Градов В. М.

Москва.
2020 г.

Цель работы: Получение навыков владения методами интерполяции таблично заданных функций с помощью кубических сплайнов**.**

Входные данные:

1. Таблица функции с количеством узлов N.
2. Значение аргумента x.

Результат работы программы:

1. Значения y(x).
2. Сравнить результаты интерполяции полиномом Ньютона 3-ей степени и кубическим сплайном.

Краткие теоретические сведения и алгоритм.

Принципиальное отличие идеи сплайн-интерполяции от интерполяции полиномом состоит в том, что полином один, а сплайн состоит из нескольких полиномов, а именно их количество равно количеству инервалов, внутри которых мы производим интерполяцию.

Кубический сплайн это гладкая функция, область определения которой разбита на конечное число отрезков, на каждом из которых она совпадает с некоторым кубическим многочленом (полиномом). Это кривая, состоящаяиз „состыкованных" полиномов третьей степени (y(IV)(x) = 0). В точках стыковки значения и производные двух соседних полиномов равны Сплайн состоит из нескольких полиномов, а именно их количество равно количеству интервалов, внутри которых мы производим интерполяцию.

Пусть $y = f(x)$ функция в интервал [a,b] и у нас есть таблица от стойности $y_i = f(x_i)$ в точках (узлах) $a \le x_0 < x_1 < x_2 < ... < x_n \le b$

Пусть наша таблица:

x_i	x_0	x_1	...	x_n
y_i	y_0	y_1	...	y_n

Определение:

Интерполационний сплайн $S_k(f,x)$ это функция, которая

(1)$S_k(f,x)$ это полином$f_i(x)$ степен k в каждый подинтервал $[x_{i-1}, x_i],\ i = \overline{1,n}$

(2) $S_k(f,x)$ интерполирует функцию, тоесть $S_k(f,x_i) = y_i,\ \ i = \overline{0,n}$

(3)$S_k(f,x)$ и его производные до (k – 1) непрерывные в [a, b].

Кубический сплайн:

Здесь k = 3 и в каждый подинтервал $[x_{i-1}, x_i]$, $i = \overline{1,n}$, сплайн $S_3(f, x)$ это полином 3 степени. Так мы ищем коэфиценты которые определяют сплайн ai, bi, ci, di. Тоесть 4n неизвестные

Мы ищем S3 :

$S_3(x) = f_i(x) = a_i + b_i(x - x_{i-1}) + c_i(x - x_{i-1})^2 + d_i(x - x_{i-1})^3$ для $x \in [x_{i-1}, x_i]$ i=1,..,n

Условия $\boxed{f_i(x_{i-1}) = y_{i-1}}$, $\boxed{f_i(x_i) = y_i}$ i=1,..n, - интерполация, (2n условия) (2)

$S_3'(x)$, $S_3''(x)$ непрерывные в внешних точек $x_1, x_2, ..., x_{n-1}$

$f_i'(x_i) = f_{i+1}'(x_i)$ (n-1 условия) (3*)

$f_i''(x_i) = f_{i+1}''(x_i)$ (n-1 условия) (3**)

У нас есть 4n-2 условия чтобы найти 4n неизвестни. Две условия остаются свободные. Следовательно кубический сплайн одинственый при заданые две дополнительные условия.

Если y1 = y2 = 0 сплайн называется Естественный сплайн

Общая формула кубического сплайна

$$S_3(x) = \begin{cases} f_1(x) = a_1 + b_1(x - x_0) + c_1(x - x_0)^2 + d_1(x - x_0)^3, & x \in [x_0, x_1] \\ \dots & \\ f_i(x) = a_i + b_i(x - x_{i-1}) + c_i(x - x_{i-1})^2 + d_i(x - x_{i-1})^3, & x \in [x_{i-1}, x_i] \\ \dots & \\ f_n(x) = a_n + b_n(x - x_{n-1}) + c_n(x - x_{n-1})^2 + d_n(x - x_{n-1})^3, & x \in [x_{n-1}, x_n] \end{cases}$$

Алгоритм:

1. Мы строим таблицу .
2. Потом задаем X.
3. Ищем интервал k , куда попадает X.
4. y = ϕ(x) = ai + bi(x−xi−1) + ci(x−xi−1)2 + di(x−xi−1)3
5. Вывод y.

Ответы на контрольные вопросы:

1. Выписать значения коэффициентов сплайна, построенного на двух точках.

- *Это есть прямая линия. Следовательно c = 0, d = 0 , a(i) = y(i – 1)*

B = (y(i) - y(i – 1)) / (x(i) - x(i – 1)

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

1. φ ``(x0) = 0
2. φ ``(x2) = 0

   1 и 2 – граничные условия.
3. φ1 `(x1) = φ2 `(x1)
4. φ1 `` (x1) = φ2 ``(x1)

   3 и 4 – условия для промежуточной точки
5. φ1(x0) = y0
6. φ1(x1) = y1
7. φ2(x1) = y1
8. φ2(x2) = y2

3. Определить начальные значения прогоночных коэффициентов, если принять, что для коэффициентов сплайна справедливо $C_1=C_2$.

- *Воспользуемся формулой* C(i)= ξ (i+1)* c (i+1) + η и получим C1 = ξ2 * c2 + η2 так как C1 = C2, по условию, сделаем преобразования C2(1 - ξ2) - η2 = 0 и отсюда получаем, что ξ2 = 1 и η2 = 0.

Код программы:

```
def f(x):
   return x**2 + 2*x - 2
   #return x**3 - 3*x + 2

def tabl(xn, xk, kol):
   XY = []
   sh = (xk-xn)/(kol-1)
   xt = xn
   while(xt <= xk):
      XY.append([xt, f(xt)])
      xt = xt+sh
   return XY
```

```
def print_t(XY, kazd):
    print("    x    ", " y    ")
    for i in range(len(XY)):
        if(i % kazd == 0):
            print("{:5.3f}".format(XY[i][0]), "{:5.3f}".format(XY[i][1]))

def opr(XY, x):
    for i in range(1, len(XY)):
        if XY[i-1][0] <= x < XY[i][0]:
            return i-1
    return len(XY)-1

def spline(XY, x, inach):

    hi = [0]
    #Шаг между x в табл.
    for i in range(1, len(XY)):
        hi.append(XY[i][0] - XY[i-1][0])

    #нахождение коэф. методом прогонки

    #начальные кси и эта из c1 = 0 и K = 1, M = 0, P = 0
    ksi = [0, 0, 0]
    eta = [0, 0, 0]

    #нахождение кси и эты
    for i in range(2, len(XY)):
        A = hi[i-1]
        B = -2*(hi[i-1]+hi[i])
        D = hi[i]
        F = -3*((XY[i][1]-XY[i-1][1])/hi[i] - (XY[i-1][1]-XY[i-2][1])/hi[i-1])
        ksi.append(D/(B - A*ksi[i]))
        eta.append((A*eta[i]+F)/(B - A*ksi[i]))
    """
    for i in range(len(XY)):
        print(A[i], " ", B[i], " ", D[i], " ", F[i])
    """

    """
    for i in range(len(eta)):
        print(eta[i], " ", end = "")
    print()
    """
```

```
    ci = []
    for i in range(len(XY)+1):
        ci.append(0)

    ci[1] = 0#eta и ci не оч., исправить
    ci[len(XY)] = 0 #т.к. eta[n] охожа на 2 гранич. условие на границах y" = 0
    for i in range(len(XY)-1, 1, -1):
        #print(ksi[i+1], " ", ci[i+1], " ", eta[i+1]," ", end = "")
        ci[i] = ksi[i+1]*ci[i+1]+eta[i+1]
        #print(ci[i])

    ai = [0]
    bi = [0]
    di = [0]
    #Находим a, b, d
    for i in range(1, len(XY)):
        ai.append(XY[i-1][1])
        bi.append(  ( (XY[i][1] - XY[i-1][1]) / hi[i]) - (hi[i] / 3* (ci[i+1]+2*ci[i]) )  )
        di.append((ci[i+1]-ci[i])/(3*hi[i]))

    IT = [ai, bi, ci, di]
    return IT


xn = -5
xk = 5
kol = 2

XY = tabl(xn, xk, kol)

print_t(XY, 1)
x = float(input("Введите x: "))
if(x <= xn or x >= xk):
    print("X не входит в область определения таблицы")
else:
    inach = opr(XY, x)
    inach += 1
    IT = spline(XY, x, inach)
    p1 = x-XY[inach-1][0]
    res = IT[0][inach] + IT[1][inach]*p1 + IT[2][inach]*p1**2 +
IT[3][inach]*p1**3
```

```
print("Вычисленное значение f(x): {:6.4f}".format(res))
print("Точное значение f(x): {:6.4f}".format(f(x)))
```