**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** Варламова Е.А.

**Группа** ИУ7-41Б

**Оценка (баллы) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Преподаватель** Градов В.М.

Москва.

2021 г

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

1. **Исходные данные**

1. Таблица функции с количеством узлов N, заданная с помощью формулы y = x2 в диапазоне [0..10] с шагом 1.

2. Значение аргумента x в первом интервале, например, при х=0.5 и в середине таблицы, например, при x= 5.5.

1. **Код программы**

(код интерполяции полиномом Ньютона не приведён, однако был взят из 2 лабораторной)

def spline(n, x, y, plot\_x):

n -= 1

h = [0]

for i in range(1, n + 1):

h.append(x[i] - x[i - 1]) # 1 .. n

ksi = [0, 0, 0]

eta = [0, 0, 0]

for i in range(2, n + 1):

ksi.append(-(h[i]) / (h[i - 1] \* ksi[i] + 2 \* (h[i - 1] + h[i])))

f = 3 \* ((y[i] - y[i - 1]) / h[i] - (y[i - 1] - y[i - 2]) / h[i - 1])

eta.append((f - h[i - 1] \* eta[i]) / (h[i - 1] \* ksi[i] + 2 \* (h[i - 1] + h[i])))

c = []

for i in range (n + 2):

c.append(0)

for i in range (n + 1, 1, -1):

c[i - 1] = c[i] \* ksi[i] + eta[i]

a = [0]

b = [0]

d = [0]

for i in range(1, n + 1):

a.append(y[i - 1])

b.append((y[i] - y[i-1]) / h[i] - h[i] \* (c[i+1] + 2 \* c[i]) / 3)

d.append((c[i+1] - c[i]) / (3 \* h[i]))

pos = -1

for i in range(n):

if x[i] <= plot\_x and x[i + 1] > plot\_x:

pos = i + 1

break

return (a[pos] + b[pos] \* (plot\_x - x[pos-1]) + c[pos] \* (plot\_x - x[pos-1])\*\*2 + d[pos] \* (plot\_x - x[pos-1])\*\*3)

n = 11

x = []

y = []

for i in range(n):

x.append(i)

y.append(i \* i)

print(spline(n, x, y, 0.5))

print(Newton\_interpolation(3, x, y, 0.5))

1. **Результат работы программы**

|  |  |  |
| --- | --- | --- |
|  | **x = 0.5** | **x = 5.5** |
| Сплайн-интерполяция | 0.3415055 | 30.2503453 |
| Интерполяция полиномом Ньютона 3-ей степени | 0.25 | 30.25 |

1. **Ответы на вопросы**

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

Точки: (x0, y0) (x1, y1)

ψ(x)=ai +bi(x−xi-1 )+c(x−xi-1)2 +d(x−xi-1)3

h = x1 - x0

В узлах значения многочлена и интерполируемой функции совпадают:

ψ( x0) = y0 = a //

ψ( x1) = y1 = a +b\* h+ c \* h2 + d \* h3 (1)

Так как участок всего один, примем, что вторая производная равна нулю на концах участка интерполирования, тогда:

ψ’’( x) = 2 \* c + 6 \* d \* (x – x0)

ψ’’( x0) = 2 \* c = 0 => c=0 //

ψ’’( x1) = 2 \* c + 6 \* d \* h = 0

0 + 6 \* d \* h= 0 => d = 0 //

Подставим в (1) коэффициенты:

y1 = y0 + b \* h => b = (y1 – y0) / h = (y1 – y0) / (x1 - x0) //

**Итог: a = y0; b = (y1 – y0) / (x1 - x0); c = d = 0**

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

В узлах значения многочлена и интерполируемой функции совпадают. Поэтому для первого интервала:

ψ1(x0) = y0 (1)

ψ1(x1) = y1  (2)

Для второго интервала:

ψ2(x1) = y1 (3)

ψ2(x2) = y2 (4)

Приравниваем во внутреннем узле первые и вторые производные:

ψ1’(x1) = ψ2’(x1) (5)

ψ1’’(x1) = ψ2’’(x1) (6)

Вторые производные на концах участка интерполирования задаются равными нулю:

ψ1’’(x0) = 0 (7)

ψ2’’(x2) = 0 (8)

Получили 8 условий, что логично: 3 точки создают 2 интервала, на каждом из которых определяются 4 коэффициента, значит нужно 8 уравнений.

*3. Определить начальные значения прогоночных коэффициентов, если принять, что для коэффициентов сплайна справедливо C1 = C2.*

ci-1 =ξici +ηi => c1 = ξ \* c2 + η

**Получаем: ξ = 1, η = 0**

*4. Написать формулу для определения последнего коэффициента сплайна СN, чтобы можно было выполнить обратный ход метода прогонки, если в качестве граничного условия задано kCN-1+mCN=p, где k,m и p - заданные числа.*

CN-1 = ξN \* CN + ηN

(p - m \*CN) / k = ξN \* CN + ηN

p – m\*CN = k \* ξN \* CN + k \* ηN

CN = (p - k \* ηN) / (k \* ξN + m)