Министерство образования Республики Беларусь
Учреждение образования
“БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ”

Факультет компьютерных систем и сетей
Кафедра информатики
Дисциплина: Методы численного анализа

ОТЧЁТ
к лабораторной работе 7
на тему

Интерполяция сплайнами

Выполнил: студент группы 053502
Герчик Артём Вадимович

Проверил: Анисимов Владимир Яковлевич

Минск 2022

Оглавление

Цели выполнения задания

1) Изучить построение кубических интерполяционных сплайнов

Краткие теоретические сведения

Рассмотрим задачу интерполяции функции *f(x)* на отрезке *[a, b]*. Пусть мы имеем узлы $a = x_0 < x_1 < \ldots < x_n = b$ и значения функции $y_0, \ldots, y_n$ в данных узлах. Отрезок разбивается узлами на *n* элементарных отрезков $[x_{i-1}, x_i]$, где $h_i = x_i - x_{i-1}$ - длина элементарного отрезка, $i = \overline{1, n}$.

Сплайном называется функция *S(x)*, которая на каждом элементарном отрезке является многочленом и непрерывна на всем отрезке [*a, b*], вместе со своими производными до некоторого порядка.

Степенью сплайна называется наивысший порядок степени многочлена.

Дефектом сплайна называется разность между его степенью и наивысшим порядком непрерывной на [a, b] производной.

Пример. Рассмотрим функцию

$$S(x) = \begin{cases} x^2, & 0 \le x < 1 \\ -x^2 + 4x - 2, & 1 \le x < 2 \\ 2, & 2 \le x < 3 \\ \dfrac{x^3}{27} - x + 4, & 3 \le x < 4. \end{cases}$$

Очевидно, функция $S(x)$ является кубическим сплайном на отрезке [0, 4], так как она непрерывна в узловых точках.

Действительно,

$$S(1-0) = S(1+0) = 1,\ S(2-0) = S(2+0) = 2,\ S(3-0) = S(3+0) = 2.$$

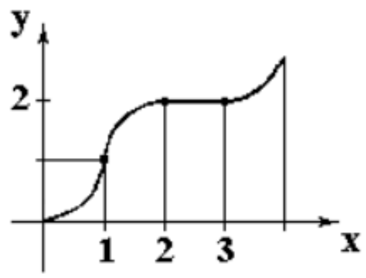

Рис. 7.1.

Найдем дефект сплайна.

$S'(1-0)=S'(1+0)=2,\ S'(2-0)=S'(2+0)=0,\ S'(3-0)=S'(3+0)=0.$

В то же время $S''(2-0)=-2,\ \ S''(2+0)=0.$

Таким образом, наибольший порядок непрерывной производной функции S на отрезке $[0,4]$ равен 1 и, следовательно, дефект сплайна равен 2. Смотри рисунок 7.1.

Отметим, что в общем случае сам сплайн многочленом не является. Чтобы он был многочленом, необходимо и достаточно, чтобы его дефект равнялся нулю.

Будем рассматривать кубические сплайны, у которых непрерывны первая и вторая производные.

Тогда на отрезке $[x_{i-1},x_i]$ сплайн *S(x)* имеет вид

$$S(x)=a_i+b_i(x-x_{i-1})+c_i(x-x_{i-1})^2+d_i(x-x_{i-1})^3, \quad i=\overline{1,n}. \qquad (7.1)$$

Таким образом, для построения кубического сплайна необходимо найти 4*n* неизвестных коэффициентов многочленов (7.1).

Очевидно $S(x_i)=y_i$, $i=\overline{0,n}$. Найдем *S(x)*. Для этого требуется определить значения *4n* неизвестных коэффициентов. Очевидно для этого необходимо иметь *4n* уравнений для определения коэффициентов.

Подставим левый конец отрезка (x_{i-1}) в уравнение:

$$S(x_{i-1})=y_{i-1}=a_i, \qquad i=\overline{1,n}$$

$$S(x_{i+1})=y_i=a_i+b_ih_i+c_ih_i^2+d_ih_i^3, \qquad i=\overline{1,n}.$$

В итоге получаем *2n* уравнений:

$$\begin{cases} y_{i-1} = a_i & i = \overline{1,n} \\ a_i + b_i h_i + c_i h_i^2 + d_i h_i^3 = y_i & i = \overline{1,n}. \end{cases}$$

Далее во всех внутренних узлах должны совпадать первая и вторая производные *S(x)*. Имеем

$$S'(x) = b_i + 2c_i(x - x_{i-1}) + 3d_i(x - x_{i-1})^2,$$

$$S''(x) = 2c_i + 6d_i(x - x_{i-1}), \qquad i = \overline{1, n-1}.$$

Приравниваем во внутренних узлах значения левых и правых производных. Получим:

$$\begin{cases} b_i + 2c_i h_i + 3d_i h_i^2 = b_{i+1} & i = \overline{1,n}, \\ c_i + 3d_i h_i = c_{i+1}, \end{cases}$$

т. е. *(2n-2)* уравнений.

Недостающие 2 уравнения можно задать разными способами. Обычно берут $S''(x_0) = S''(x_n) = 0$.

Отсюда

$$2c_1 = 0, \qquad 2c_n + 6d_n h_n = 0.$$

Для удобства положим еще $c_{n+1} = 0$.

Объединяя все уравнения, получим систему

$$
\begin{cases}
y_{i-1} = a_i & i = \overline{1,n} \\
a_i + b_i h_i + c_i h_i^2 + d_i h_i^3 \quad y_i \quad = & i \quad \overline{1,n} \quad = \\
b_i + 2c_i h_i + 3d_i h_i^2 \quad b_{i+1} & i \quad \overline{1,n-1} \\
c_i + 3d_i h_i \quad c_{i+1} & i \quad \overline{1,n-1} \\
c_n + 3d_n h_k = 0 & \\
c_1 = 0 & \\
c_{n+1} = 0. &
\end{cases}
$$

Решая систему, получим

$$
\begin{cases}
b_i h_i + c_i h_i^2 + d_i h_i^3 \quad y_i - y_{i-1} \quad = & i = \overline{1,n} \\
2c_i h_i + 3d_i h_i^2 \quad b_{i+1} - b_i \quad = & i \quad \overline{1,n-1} \\
d =_i \dfrac{c_{i+1} - c_i}{3h_i} & \\
c_1 = c_{n+1} = 0, &
\end{cases}
$$

далее

$$
\begin{cases}
a_i = y_{i-1} & i = \overline{1,n} \\
b_i h_i + c_i h_i^2 + \left(\dfrac{(c_{i+1} - c_i) h_i^2}{3} \right) = y_i - y_{i-1} & i = \overline{1,n} \\
2c_i h_i + (c_{i+1} - c_i) h_i = b_{i+1} - b_i & i = \overline{1,n-1} \\
d =_i \dfrac{c_{i+1} - c_i}{3h_i} & i = \overline{1,n} \\
c_1 = c_{n+1} = 0. &
\end{cases}
$$

Откуда

$$
b_i = \frac{y_i - y_{i-1}}{h_i} - c_i h_i - \frac{(c_{i+1} - c_i) h_i}{3}, \qquad i = \overline{1,n}.
$$

$$2c_i h_i + (c_{i+1} - c_i)h_i = \frac{y_{i+1} - y_i}{h_{i+1}} - c_{i+1}h_{i+1} - \frac{(c_{i+2} - c_{i+1})h_{i+1}}{3} - \frac{y_i - y_{i-1}}{h_i} + c_i h_i + \frac{(c_{i+1} - c_i)h_i}{3}.$$

Таким образом, задача определения коэффициентов сплайна свелась к решению системы

$$c_i\left(\frac{h_i}{3}\right) + c_{i+1}\left(\frac{2}{3}h_i + \frac{2}{3}h_{i+1}\right) + c_{i+2}\left(\frac{h_{i+1}}{3}\right) = \frac{y_{i+1} - y_i}{h_{i+1}} - \frac{y_i - y_{i-1}}{h_i}, \quad i = \overline{1, n-1} \qquad (7.2)$$

$$c_1 = c_{n+1} = 0.$$

Это треухдиагональная система и ее целесообразно решать методом прогонки. Поскольку для матрицы системы выполнено условие доминирования диагональных элементов

$$\frac{2}{3}h_i + \frac{2}{3}h_{i+1} > \frac{h_i}{3} + \frac{h_{i+1}}{3},$$

то система имеет решение, причем единственное, и это решение можно найти методом прогонки.

Метод прогонки применяется для трехдиагональных систем, которые имеют матрицу вида:

$$\begin{pmatrix} b_1 & c_1 & 0 & \dots & \dots & \dots & 0 \\ a_2 & b_2 & c_2 & 0 & \dots & \dots & 0 \\ 0 & a_3 & b_3 & c_3 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & \dots & 0 & a_{n-1} & b_{n-1} & c_{n-1} \\ 0 & \dots & \dots & 0 & 0 & a_n & b_n \end{pmatrix},$$

т. е. матрицу, у которой ненулевыми могут быть только элементы, стоящие на главной и двух смежных с главной диагоналях. Т.е. трехдиагональная систем имеет вид:

$$\begin{cases} b_1x_1 + c_1x_2 = d_1 \\ a_2x_1 + b_2x_2 + c_2x_3 = d_2 \\ a_3x_2 + b_3x_3 + c_3x_4 = d_3 \\ \dots\dots\dots\dots\dots\dots \\ a_{n-1}x_{n-2} + b_{n-1}x_{n-1} + c_{n-1}x_n = d_{n-1} \\ a_nx_{n-1} + b_nx_n = d_n \end{cases}$$

Суть метода прогонки заключается в построении рекуррентной последовательности для нахождения прогоночных коэффициентов A_i и B_i, а каждое неизвестное представляется в виде

$$x_i = A_ix_{i+1} + B_i\,.$$

Для удобства полагают, что $a_n = 0$, $c_n = 0$ и тогда формулы для прогоночных коэффициентов принимают следующий вид:

$$\begin{cases} A_i = \dfrac{-c_i}{a_iA_{i-1} + b_i} \\ B_i = \dfrac{d_i - a_iB_{i-1}}{a_iA_{i-1} + b_i}. \end{cases} \qquad i = \overline{1,n}\,.$$

При этом

$$\begin{cases} \mathrm{x_i} = A_i \cdot x_{i+1} + B_i, \quad i = \overline{1,n-1} \\ x_n = B_n. \end{cases}$$

Проводя обратный ход метода прогонки, последовательно найдем значения неизвестных $x_n, x_{n-1}, \dots, x_1$.

Программная реализация

```
import numpy



def input_values():
    ##### EXAMPLES #####

    # def function(x): return numpy.tan(x)
    # left_border, dots_count, right_border = -1.5, 11,
1.5

    # def f(x): return numpy.sin(x)
    # left_border, dots_count, right_border = -1, 7, 1

    ##### EXAMPLES #####

    def function(x): return numpy.cosh(x)  # func of my
var

    left_border, dots_count, right_border = 0, 6, 2

    dots = []
    for i in range(dots_count):
        x = left_border + (right_border - left_border) *
i / (dots_count - 1)
        y = function(x)
```

```
        dots += [(x, y)]

    return dots, function



print("Интерполяция сплайнами\n")


dots, f = input_values()
(x, y) = map(list, zip(*dots))
print("(x,y) =", dots, '\n')



def triple_diagonal_solve(A, b):
    A = A.copy()
    b = b.copy()
    n = len(A)

    A[0][1] /= A[0][0]
    for i in range(1, n - 1):
        A[i][i + 1] /= (A[i][i] - A[i][i - 1] * A[i - 1]
[i])

    b[0] /= A[0][0]
    for i in range(1, n):
        b[i] = (b[i] - A[i][i - 1] * b[i - 1]) / (A[i][i]
- A[i][i - 1] * A[i - 1][i])
```

```
    x = numpy.zeros(n)
    x[-1] = b[-1]
    for i in range(n - 2, -1, -1):
        x[i] = b[i] - A[i][i + 1] * x[i + 1]

    return x


def spline_method(dots):
    n = len(dots) - 1
    (x, y) = map(list, zip(*dots))

    h = [None]
    for i in range(1, n + 1):
        h += [x[i] - x[i - 1]]

    A = [[None] * (n) for i in range(n)]
    for i in range(1, n):
        for j in range(1, n):
            A[i][j] = 0.0
    for i in range(1, n - 1):
        A[i + 1][i] = h[i + 1]
    for i in range(1, n):
        A[i][i] = 2 * (h[i] + h[i + 1])
```

```
    for i in range(1, n - 1):
        A[i][i + 1] = h[i + 1]

    F = []
    for i in range(1, n):
        F += [3 * ((y[i + 1] - y[i]) / h[i + 1] - (y[i] - y[i - 1]) / h[i])]

    A = [A[i][1:] for i in range(len(A)) if i]

    c = triple_diagonal_solve(A, F)
    c = [0.0] + list(c) + [0.0]

    def evaluate(x_dot):
        for i in range(1, len(x)):
            if x[i - 1] <= x_dot <= x[i]:
                val = 0
                val += y[i]
                b = (y[i] - y[i - 1]) / h[i] + (2 * c[i] + c[i - 1]) * h[i] / 3
                val += b * (x_dot - x[i])
                val += c[i] * ((x_dot - x[i]) ** 2)
                d = (c[i] - c[i - 1]) / (3 * h[i])
                val += d * ((x_dot - x[i]) ** 3)
                return val
        return None
```

```
    def output():
        print("Кубический сплайн: ", '\n')
        for i in range(1, len(x)):
            val = 0
            b = (y[i] - y[i - 1]) / h[i] + (2 * c[i] +
c[i - 1]) * h[i] / 3
            d = (c[i] - c[i - 1]) / (3 * h[i])
            print(x[i - 1], x[i], )
            print(numpy.poly1d([d, c[i], b, y[i]]), '\n')

    return evaluate, output


spl, cout = spline_method(dots)

cout()

x_dot = 1.0
print(f"Исходная функция от ({x_dot}) = ", f(x_dot))
print(f"Кубический сплайн от ({x_dot}) = ", spl(x_dot))
print(f"Дельта от ({x_dot}) = ", abs(f(x_dot) -
spl(x_dot)))
```

Полученные результаты

Тестовый пример 1

$tg(x)$	[0,1.5]	6	0,9316

Интерполяция сплайнами

(x,y) = [(0.0, 0.0), (0.3, 0.3093362496096232), (0.6, 0.6841368083416923), (0.9, 1.260158217550339), (1.2, 2.5721516221263183), (1.5, 14.10141994717172)]

Кубический сплайн:

0.0 0.3

$-1.174 x^3 - 1.057 x^2 + 0.8198 x + 0.3093$

0.3 0.6

$8.295 x^3 + 6.409 x^2 + 2.425 x + 0.6841$

0.6 0.9

$-26.98 x^3 - 17.87 x^2 - 1.013 x + 1.26$

0.9 1.2

$119.4 x^3 + 89.61 x^2 + 20.51 x + 2.572$

1.2 1.5

$-99.57 x^3 + 47.39 x + 14.1$

Исходная функция от (0.75) = 0.9315964599440726
Кубический сплайн от (0.75) = 1.1011074046476517
Дельта от (0.75) = 0.16951094470357908

Тестовый пример 2

$sh(x)$	[0,2]	6	1,1752

Интерполяция сплайнами

(x,y) = [(0.0, 0.0), (0.4, 0.4107523258028155), (0.8, 0.888105982187623), (1.2, 1.5094613554121725), (1.6, 2.37556795320023), (2.0, 3.6268604078470186)]

Кубический сплайн:

0.0 0.4

$0.1617 x^3 + 0.1941 x^2 + 1.079 x + 0.4108$

0.4 0.8

$0.2319 x^3 + 0.4724 x^2 + 1.345 x + 0.8881$

0.8 1.2

$0.1199 x^3 + 0.6163 x^2 + 1.781 x + 1.509$

1.2 1.6

$0.8627 x^3 + 1.651 x^2 + 2.688 x + 2.376$

1.6 2.0

$-1.376 x^3 + 3.348 x + 3.627$

Исходная функция от (1.0) = 1.1752011936438014
Кубический сплайн от (1.0) = 1.177009499493866
Дельта от (1.0) = 0.0018083058500646398

ЗАДАНИЕ

Вариант 7

ЗАДАНИЕ. Произвести интерполирование кубическими сплайнами приведенных в таблице функций. Вычислить значение сплайна в точке $x = 0.5*(b-a)$.

Значение сплайна в точке $x = 0.5*(b-a)$ записать в качестве ответа. Сравнить

его со значением функции в соответствующей точке.

7.	$ch(x)$	[0,2]	6	1,5431

Полученные результаты:

Интерполяция сплайнами

(x,y) = [(0.0, 1.0), (0.4, 1.0810723718384547), (0.8, 1.3374349463048447), (1.2, 1.8106555673243747), (1.6, 2.5774644711948853), (2.0, 3.7621956910836314)]

Кубический сплайн:

0.0 0.4

$$0.5472 x^{3} + 0.6567 x^{2} + 0.3778 x + 1.081$$

0.4 0.8

$$0.002736 x^{3} + 0.66 x^{2} + 0.9045 x + 1.337$$

0.8 1.2

$$0.09132 x^{3} + 0.7695 x^{2} + 1.476 x + 1.811$$

1.2 1.6

$$0.8309 x^{3} + 1.767 x^{2} + 2.491 x + 2.577$$

1.6 2.0

$-1.472 x^3 + 3.197 x + 3.762$

Исходная функция от (1.0) = 1.5430806348152437
Кубический сплайн от (1.0) = 1.5454550042033537
Дельта от (1.0) = 0.002374369388109976

Выводы

Таким образом, в ходе выполнения лабораторной работы была освоена интерполяция функций с помощью кубических сплайнов. Составлена программа, на тестовых примерах проверена правильность её работы, вычислено значение функции в точке согласно заданному варианту.