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
на тему

Интерполяция сплайнами

Выполнил: студент группы 153503
Кончик Денис Сергеевич

Проверил: Анисимов Владимир Яковлевич

Минск 2022

Содержание

1. ЦЕЛЬ РАБОТЫ 3
2. ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ 3
3. АЛГОРИТМ РЕШЕНИЯ 7
4. ПРОГРАММНАЯ РЕАЛИЗАЦИЯ 8
5. ТЕСТОВЫЕ ПРИМЕРЫ 10
6. ЗАДАНИЕ 11
7. ВЫВОД 12

1. ЦЕЛЬ РАБОТЫ

Изучить построение кубических интерполяционных сплайнов.

2. ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Рассмотрим задачу интерполяции функции *f(x)* на отрезке [*a*, *b*]. Пусть мы имеем узлы $a = x_0 < x_1 < ... < x_n = b$ и значения функции $y_0, ..., y_n$ 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,

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

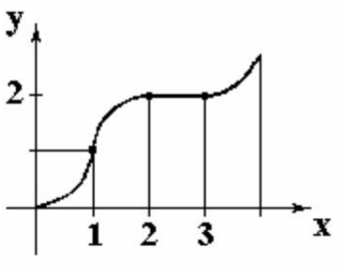


Рис. 7.3.

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. (См. рис. 7.3).

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

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

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:

$$\begin{cases} y_{i-1}=a_i & i=\overline{1,n} \\ a_i+b_ih_i+c_ih_i^2+d_ih_i^3=y_i & i=\overline{1,n}. \end{cases}$$

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:

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

т. е. $(2n-2)$ уравнений.

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

Отсюда

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

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

Объединяя все уравнения, получим систему

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

Решая систему, получим

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

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

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

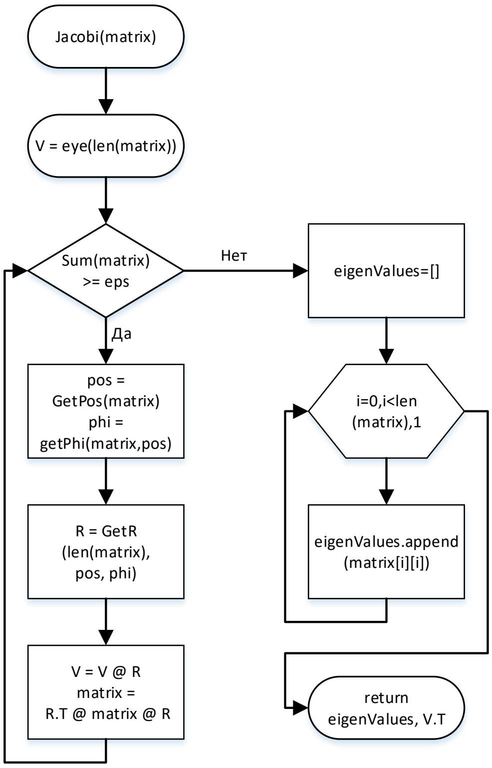
$$c_1 = c_{n+1} = 0.$$

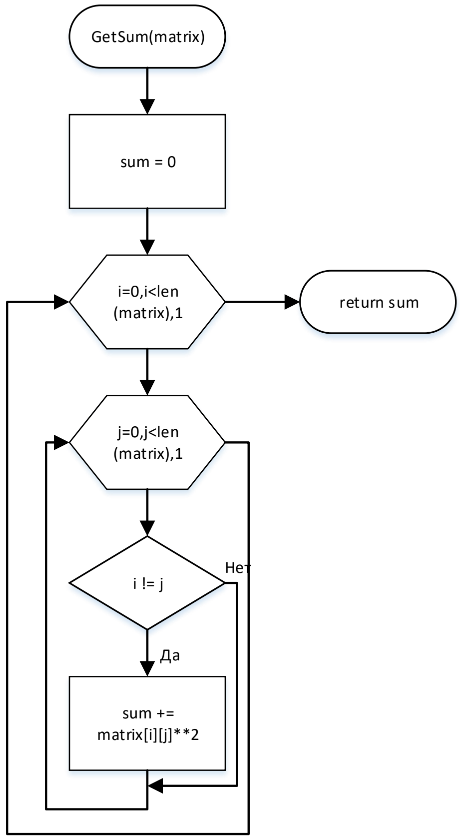
Система трехдиагональна. Будем решать ее методом прогонки. Поскольку для матрицы системы выполнено условие доминирования диагональных элементов

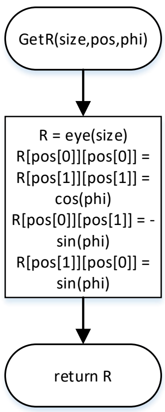
$$\frac{2}{3}h_i + \frac{2}{3}h_{i+1} > \frac{h_i}{3} + \frac{h_{i+1}}{3},$$

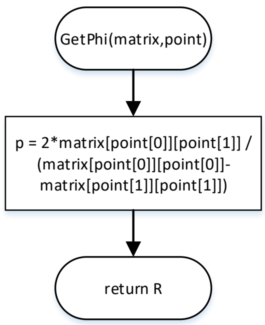
то задача имеет решение, причем единственное, и это решение можно найти методом прогонки.

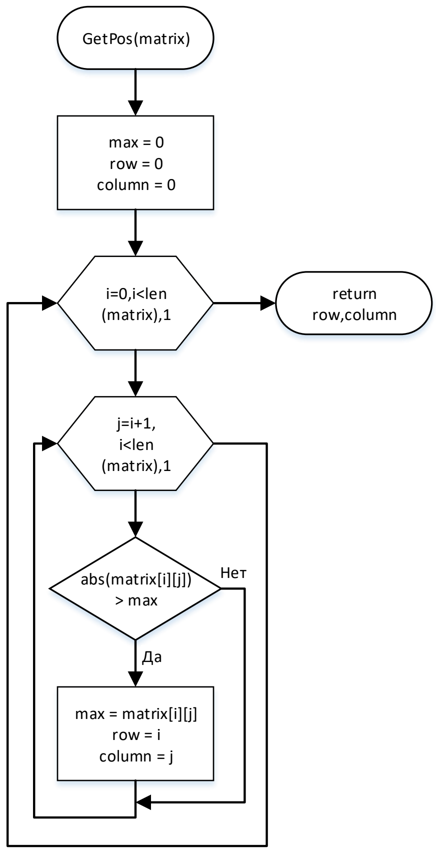
3. АЛГОРИТМ РЕШЕНИЯ

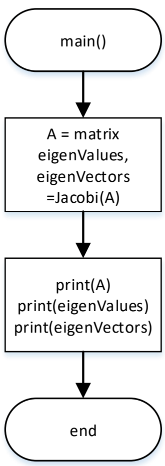












4. ПРОГРАММНАЯ РЕАЛИЗАЦИЯ

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


def input():


    def f(x):
        return np.e ** (-x)
    a, b = 0, 4

    nodes = 6 # Число узлов
    xdot = 0.5*(b-a)

    dots = [] # Список пар (узел, значение в узле)
    for i in range(nodes):
        x = a + (b - a) * i / (nodes - 1)
        y = f(x)
        dots += [(x, y)]

    return dots, f, xdot




def tridiag_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 = np.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(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):
```

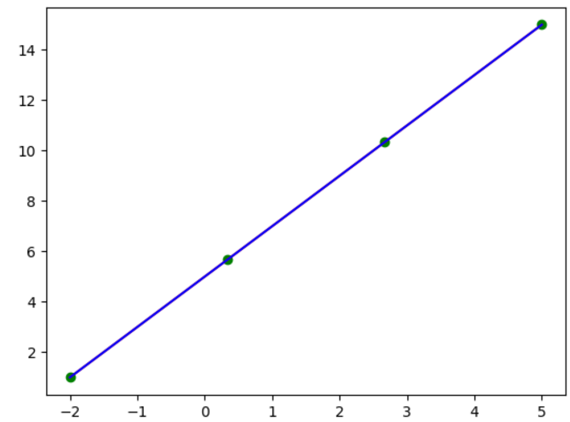
```
                A[i][j] = 0.0
        for i in range(1,n-1):
            A[i+1][i] = h[i+1]
        for i in range(1,n):
            A[i][i] = 2 * (h[i] + h[i+1])
        for i in range(1,n-1):
            A[i][i+1] = h[i+1]

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

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

        def cout():
            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(np.poly1d([d, c[i], b, y[i]]), '\n')

        return evaluate, cout

    dots, f, _ = input()

    (x, y) = map(list, zip(*dots)) # Список x и y отдельно

    print("(x,y) =", dots, '\n')
    plotdots = 10**4
    plt.plot(x, y, 'og') # Зеленые точки (x, y)
    xplot = np.linspace(min(x), max(x), plotdots) # Очень много иксов

    yplot = [f(xdot) for xdot in xplot] # Игреки от исходной функции
    plt.plot(xplot, yplot, 'red') # Построение исходной функции
    spl, cout = Spline(dots)
    yplot = [spl(xdot) for xdot in xplot] # Игреки от сплайна
    plt.plot(xplot, yplot, 'blue') # Построение сплайна
    cout()
    width = 25 # Ширина вывода текста
    _, _, xdot = input()
    print(f"f({xdot}) =".ljust(width), f(xdot))
    print(f"Кубический сплайн({xdot}) =".ljust(width), spl(xdot))
    print(f"Разность({xdot}) =".ljust(width), abs(f(xdot) - spl(xdot)))

    plt.show()
```

5. ТЕСТОВЫЕ ПРИМЕРЫ

Построить кубические интерполяционные сплайны (красный цвет – функция, синий – сплайн). Вычислить значение сплайна и функции в точке, сравнить значения.

Тестовый пример 1

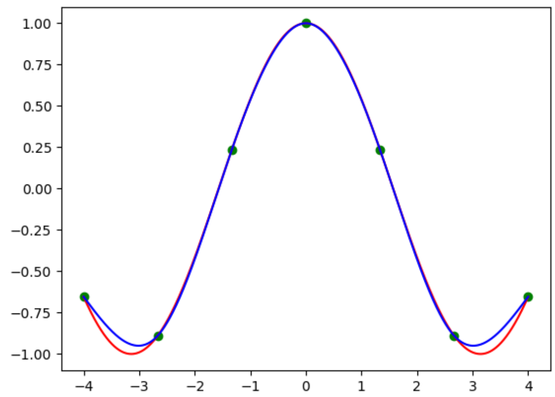
Функция	a	b	узлы	x
$y = 2x + 5$	-2	5	4	3



f(3) = 11
spline(3) = 11
|Δ| = 0

Тестовый пример 2

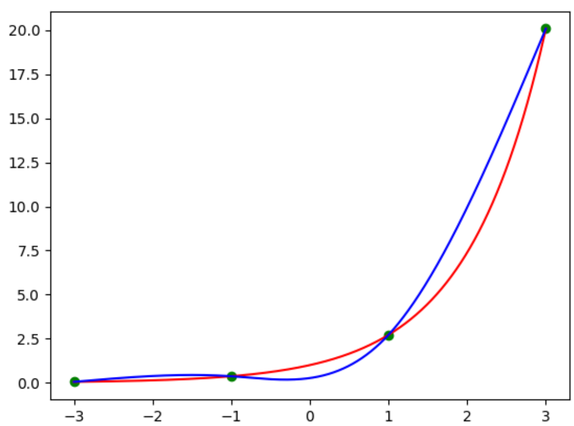
Функция	a	b	узлы	x
$y = \cos(x)$	-4	4	7	3.2



f(1) ≈ -0.9983
spline(1) ≈ -0.9350
|Δ| ≈ 0.0633

Тестовый пример 3

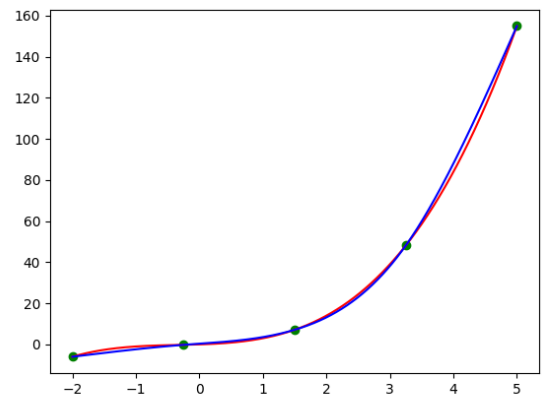
Функция	a	b	узлы	x
$y = \mathrm{ctg}(x)$	-3	3	4	2.2



f(2.2) ≈ 9.0250, spline(2.2) ≈ 11.8386
|Δ| ≈ 2.8136

Тестовый пример 4

Функция	a	b	узлы	x
$y = x^3 + x^2 + x$	-2	5	5	-1.2

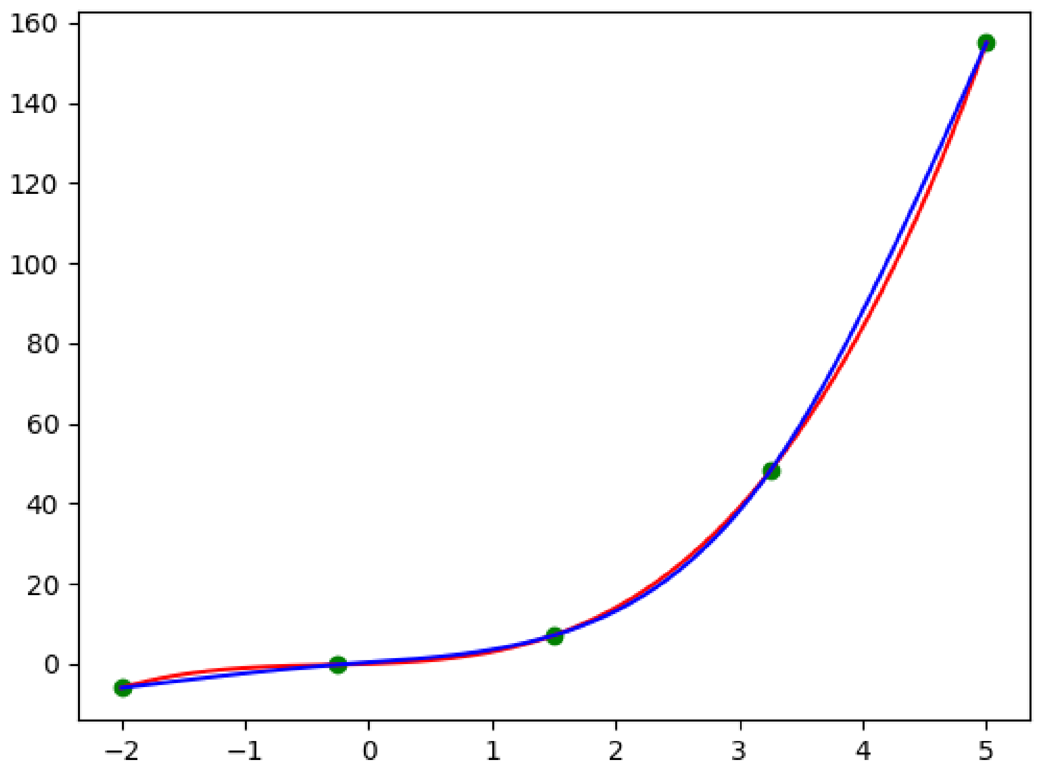


f(-1.2) ≈ -1.4879, spline(-1.2) ≈ -3.0534,
|Δ| ≈ 1.5655

6. ЗАДАНИЕ

Вариант 11

Произвести интерполирование кубическими сплайнами функции заданного варианта. Вычислить значение сплайна в точке $x = \frac{b-a}{2}$. Сравнить значение сплайна со значением функции в соответствующей точке.

<table>
<tr><td>Функция</td><td colspan="2">Отрезок</td><td>Узлы</td></tr>
<tr><td>$y = e^{-x}$</td><td colspan="2">$[0, 4]$</td><td>6</td></tr>
<tr><td colspan="4">$x = \frac{b-a}{2} = \frac{4-0}{2} = 2$</td></tr>
<tr><td colspan="2">$f(2) \approx 0.1353$</td><td colspan="2">$spline(2) \approx 0.1372$</td></tr>
<tr><td colspan="4">$|\Delta| \approx 0.0019$</td></tr>
<tr><td colspan="4">
</td></tr>
</table>

7. ВЫВОД

Таким образом, в ходе выполнения лабораторной работы была изучена интерполяция функций с помощью кубических сплайнов. Составлен алгоритм и программа нахождения сплайнов, проверена правильность работы на тестовых примерах. Согласно заданному варианту произведено интерполирование кубическими сплайнами функции заданного варианта, вычислено значение функции и сплайна в точке.