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](#_Toc119363919)

[2. ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ 3](#_Toc119363920)

[3. АЛГОРИТМ РЕШЕНИЯ 7](#_Toc119363921)

[4. ПРОГРАММНАЯ РЕАЛИЗАЦИЯ 8](#_Toc119363922)

[5. ТЕСТОВЫЕ ПРИМЕРЫ 10](#_Toc119363923)

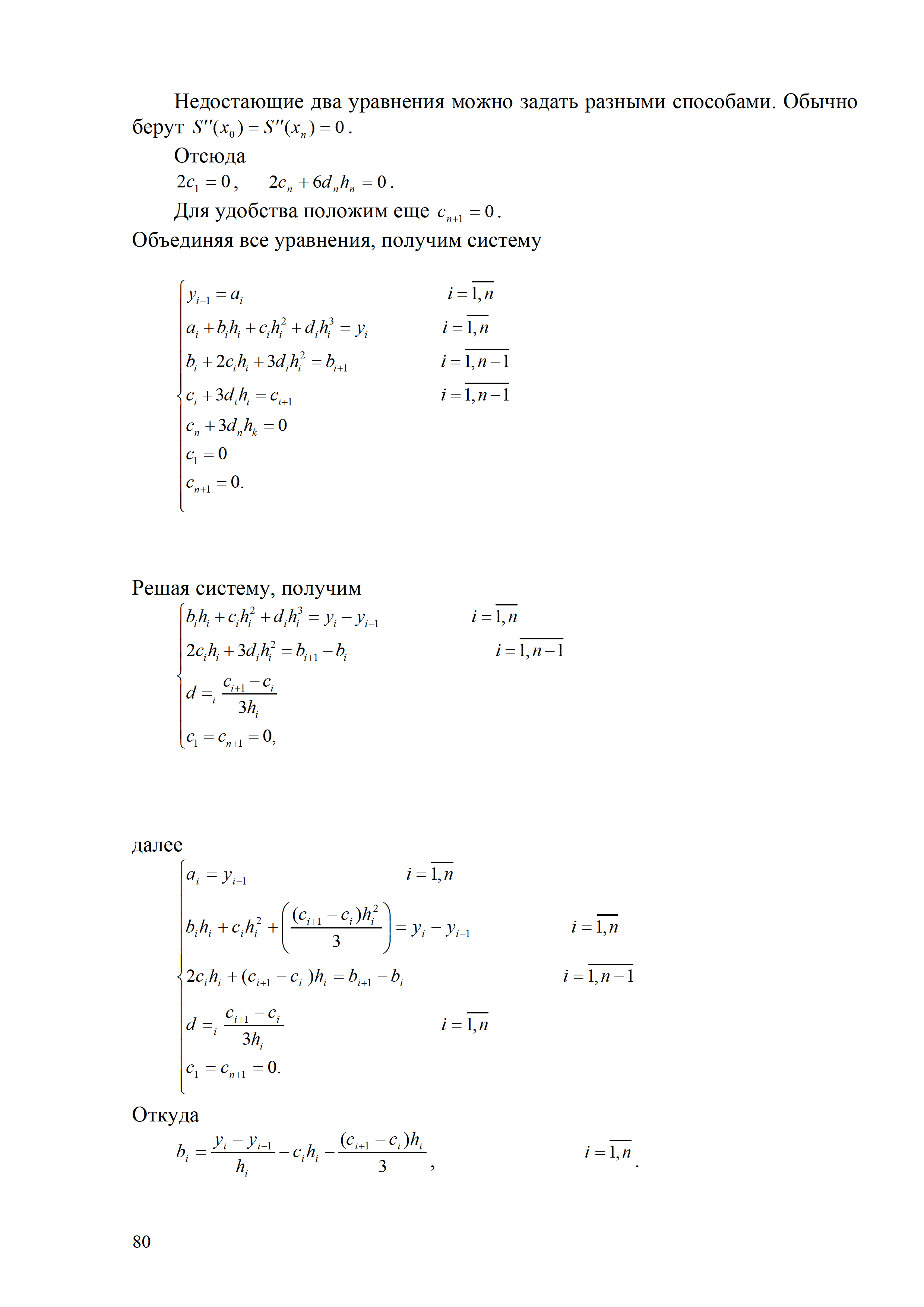
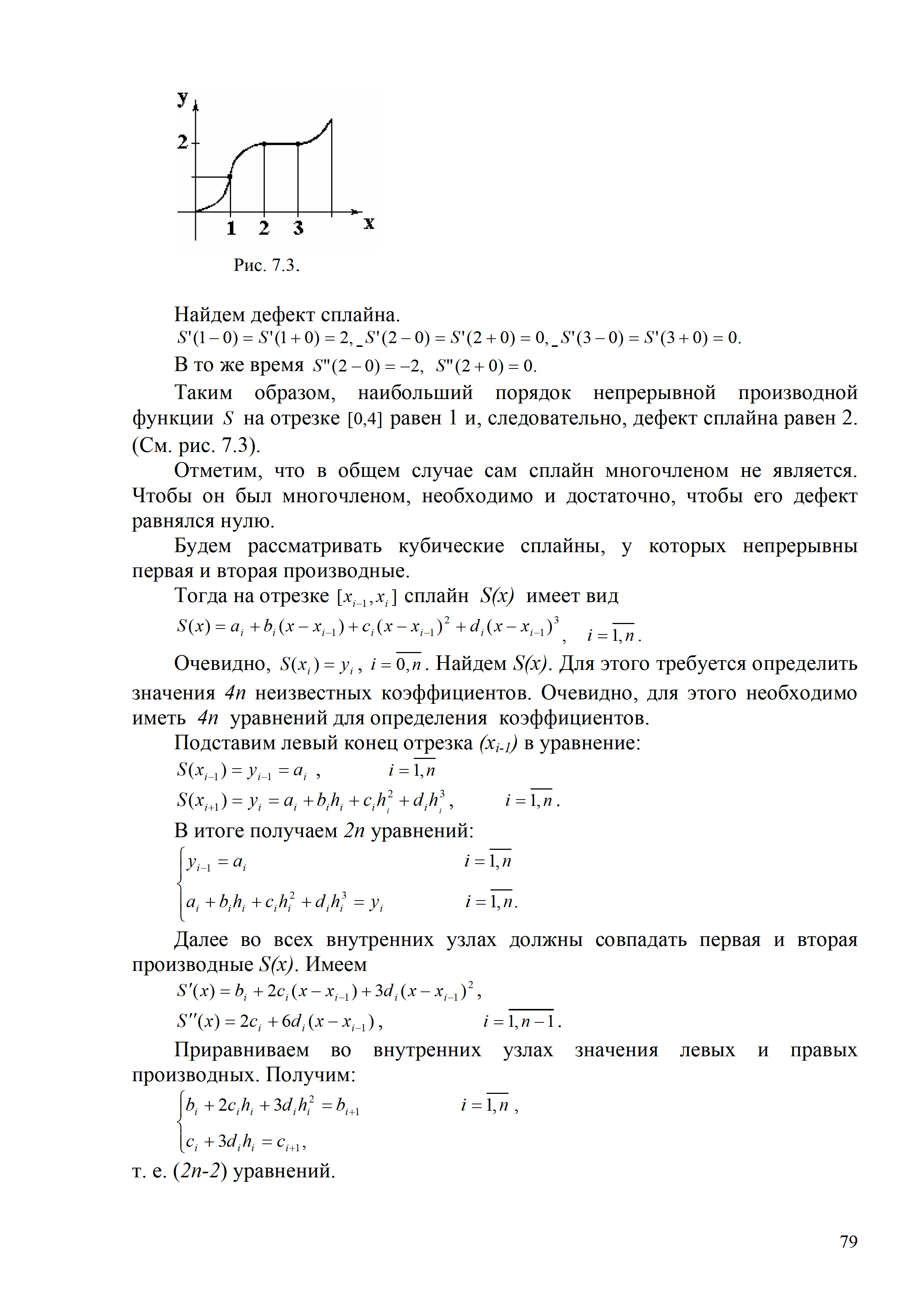
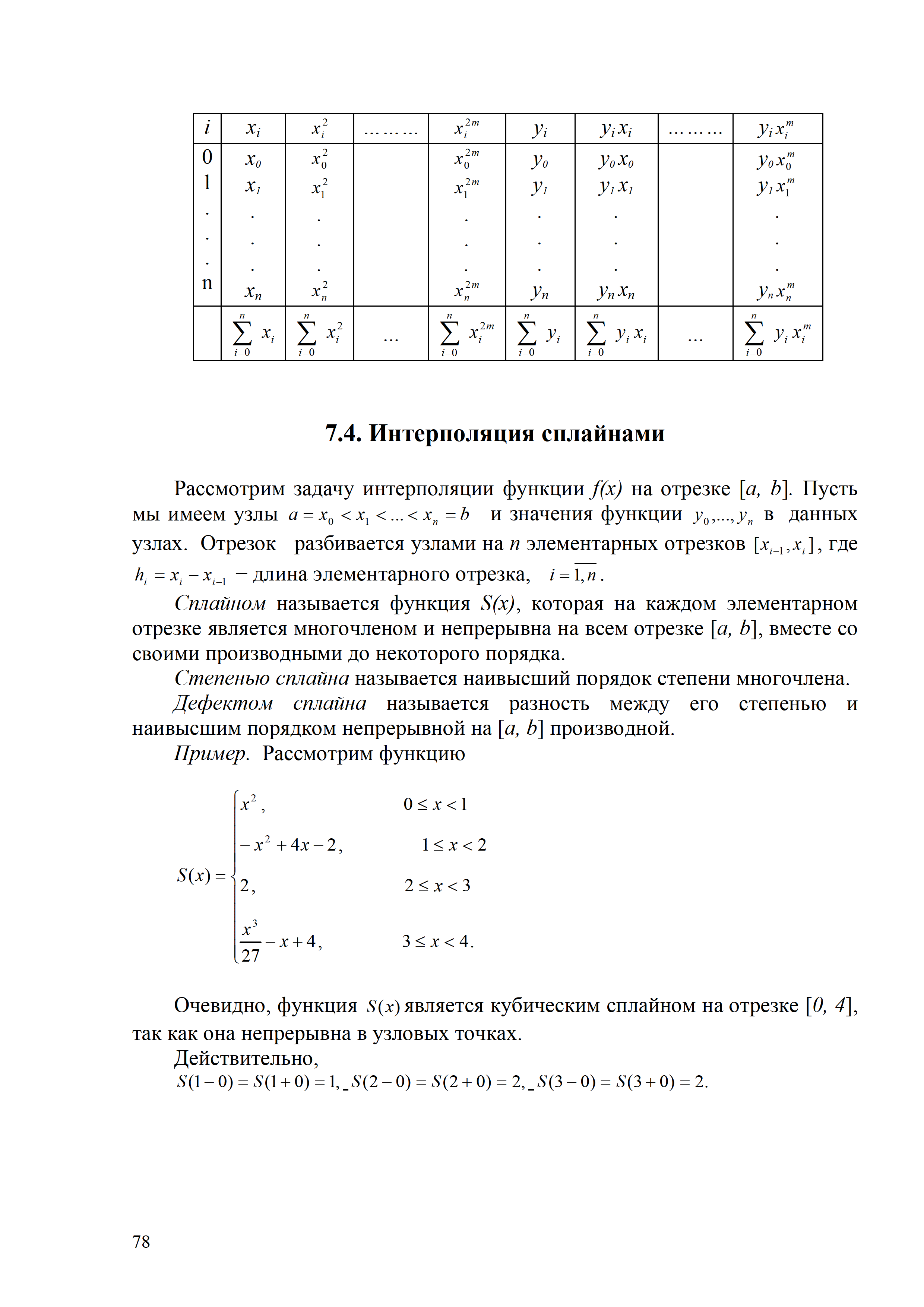
[6. ЗАДАНИЕ 11](#_Toc119363924)

[7. ВЫВОД 12](#_Toc119363925)

1. **ЦЕЛЬ РАБОТЫ**

Изучить построение кубических интерполяционных сплайнов.

1. **ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ**



# **АЛГОРИТМ РЕШЕНИЯ**



# **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 | |  | -2 | 5 | 4 | 3 | | Тестовый пример 2   |  |  |  |  |  | | --- | --- | --- | --- | --- | | Функция | a | b | узлы | x | |  | -4 | 4 | 7 | 3.2 | |
| *f(3) = 11*  *spline(3) = 11*  *= 0* | *f(1) ≈ -0.9983*  *spline(1) ≈ -0.9350*  *≈ 0.0633* |
| Тестовый пример 3   |  |  |  |  |  | | --- | --- | --- | --- | --- | | Функция | a | b | узлы | x | |  | -3 | 3 | 4 | 2.2 | | Тестовый пример 4   |  |  |  |  |  | | --- | --- | --- | --- | --- | | Функция | a | b | узлы | x | |  | -2 | 5 | 5 | -1.2 | |
| *f(2.2) ≈ 9.0250, spline(2.2) ≈ 11.8386*  *≈ 2.8136* | *f(-1.2) ≈ -1.4879, spline(-1.2) ≈ -3.0534,*  *≈ 1.5655* |

6. ЗАДАНИЕ

Вариант 11

Произвести интерполирование кубическими сплайнами функции заданного варианта. Вычислить значение сплайна в точке. Сравнить значение сплайна со значением функции в соответствующей точке.

|  |  |  |  |
| --- | --- | --- | --- |
| Функция | Отрезок | | Узлы |
|  |  | |  |
|  | | | |
| *f(2)* | | *spline(2)* | |
|  | | | |
|  | | | |

# **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.