

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** Зайцева А. А.

**Группа** ИУ7 – 42Б

**Оценка (баллы) \_\_\_\_\_\_\_**

**Преподаватель** Градов В. М.

Москва.

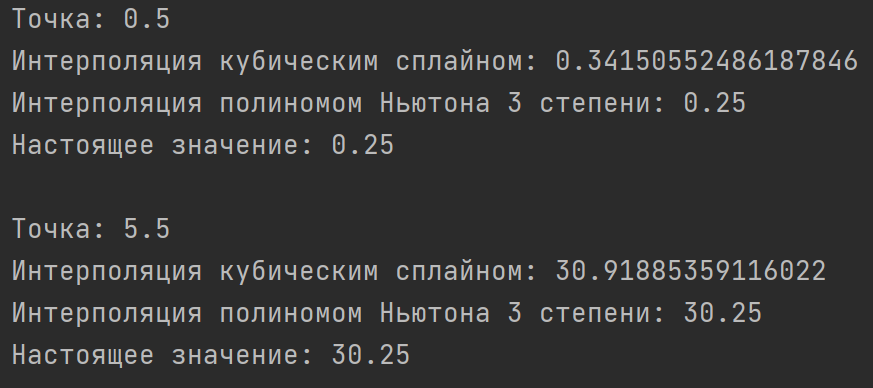
2021 г

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

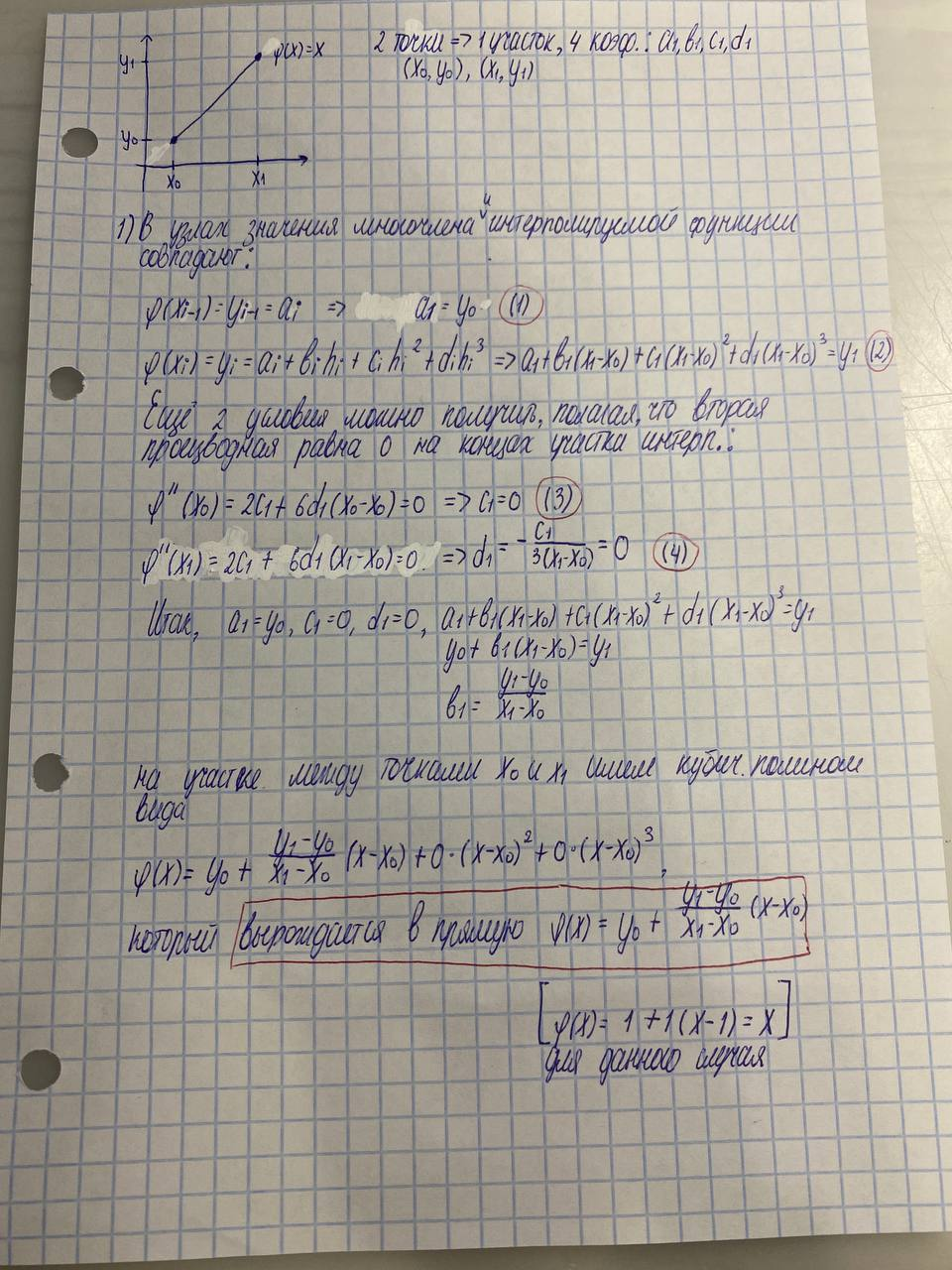
1. Исходные данные
   1. Таблица функции с количеством узлов N. Задать с помощью формулы y=x^2 в диапазоне [0..10] с шагом 1.
   2. Значение аргумента x в первом интервале, например, при х=0.5 и в середине таблицы, например, при x= 5.5. Сравнить с точным значением.
2. Код программы

|  |
| --- |
| import numpy as np  # Таблица функции с количеством узлов N(11). Задать с помощью формулы # y=x^2 в диапазоне [0..10] с шагом 1. # [(0, 0), (1, 1), (2, 4), (3, 9), (4, 16), (5, 25), (6, 36), (7, 49), (8, 64), (9, 81), (10, 100)] def make\_table(N=11):    x = np.linspace(0, 10, N, dtype=int)    y = x \*\* 2    return {'x': x, 'y': y}  # создаем таблицу data = make\_table() N = len(data['x'])  # находим hi data['h'] = [None] + list(data['x'][i] - data['x'][i - 1]                          for i in range(1, N)) # находим fi data['f'] = [None, None] + list(3 \* (((data['y'][i] - data['y'][i - 1]) / data['h'][i]) -                                ((data['y'][i - 1] - data['y'][i - 2]) / data['h'][i - 1]))                                for i in range(2, N)) # прямой ход: при известных кси2=0 и эта2=0, находим все прогоночные коэффициенты data['xi'] = [None, None] + [0] for i in range(3, N + 1):    xi\_i = -data['h'][i - 1] / (data['h'][i - 2] \* data['xi'][i - 1] +                     2 \* (data['h'][i - 2] + data['h'][i - 1]))    data['xi'].append(xi\_i)  data['eta'] = [None, None] + [0] for i in range(3, N + 1):    eta\_i = ((data['f'][i - 1] - data['h'][i - 2] \* data['eta'][i - 1]) /             (data['h'][i - 2] \* data['xi'][i - 1] + 2 \* (data['h'][i - 2] + data['h'][i - 1])))    data['eta'].append(eta\_i)  # обратные ход: при условии c[N+1]=0, определяем все коэффициенты c # с помощью прогоночных коэффициентов data['c'] = [None] + [0] \* (N) for i in range(N - 1, 0, -1):    data['c'][i] = data['xi'][i + 1] \* data['c'][i + 1] + data['eta'][i + 1]  # с помощью коэффициентов c находим коэффициенты b и d data['b'] = [None] + list((data['y'][i] - data['y'][i - 1]) / data['h'][i] -                          data['h'][i] \* (data['c'][i + 1] - 2 \* data['c'][i]) / 3                          for i in range(1, N))  data['d'] = [None] + list((data['c'][i + 1] - data['c'][i]) / 3 / data['h'][i]                          for i in range(1, N))  # находим все коэффициенты a из условия, что в узлах значения многочлена и интерполируемой функции совпадают data['a'] = [None] + list(data['y'][i - 1] for i in range(1, N))   # поиск коэффициентов полинома на участке, в котором находится точка x # а также точки, с которой начинается этот участок def choose\_coeffs(data, x):    i\_beg = 0    for i in range(1, N - 1):        if x < data['x'][i]:            i\_beg = i            break    return [data['a'][i\_beg], data['b'][i\_beg], data['c'][i\_beg], data['d'][i\_beg],            data['x'][i\_beg - 1]]  # подсчет значения полинома в точке x на участке, начинающемся с точки x0 def count\_polynom3(a, b, c, d, x0, x):    return (a + b \* (x - x0) + c \* (x - x0) \*\* 2 +            d \* (x -x0) \*\* 3)  # вводим значения x и интерполируем кубическим сплайном x1 = 0.5 coeffs1 = choose\_coeffs(data, x1) res\_spline1 = count\_polynom3(\*coeffs1, x1)  x2 = 5.5 coeffs2 = choose\_coeffs(data, x2) res\_spline2 = count\_polynom3(\*coeffs2, x2)  # интерполируем полиномом Ньютона 3 степени (lab1.py - код из 1 ЛР) import lab1 res\_newton1 = lab1.approximate\_newton(data['x'], data['y'], 3, x1) res\_newton2 = lab1.approximate\_newton(data['x'], data['y'], 3, x2)  # подсчитываем настоящее значение res\_real1 = x1 \*\* 2 res\_real2 = x2 \*\* 2  # вывод и сравнение результатов print('Точка:', x1) print('Интерполяция кубическим сплайном:', res\_spline1) print('Интерполяция полиномом Ньютона 3 степени:', res\_newton1) print('Настоящее значение:', res\_real1) print()  print('Точка:', x2) print('Интерполяция кубическим сплайном:', res\_spline2) print('Интерполяция полиномом Ньютона 3 степени:', res\_newton2) print('Настоящее значение:', res\_real2) |

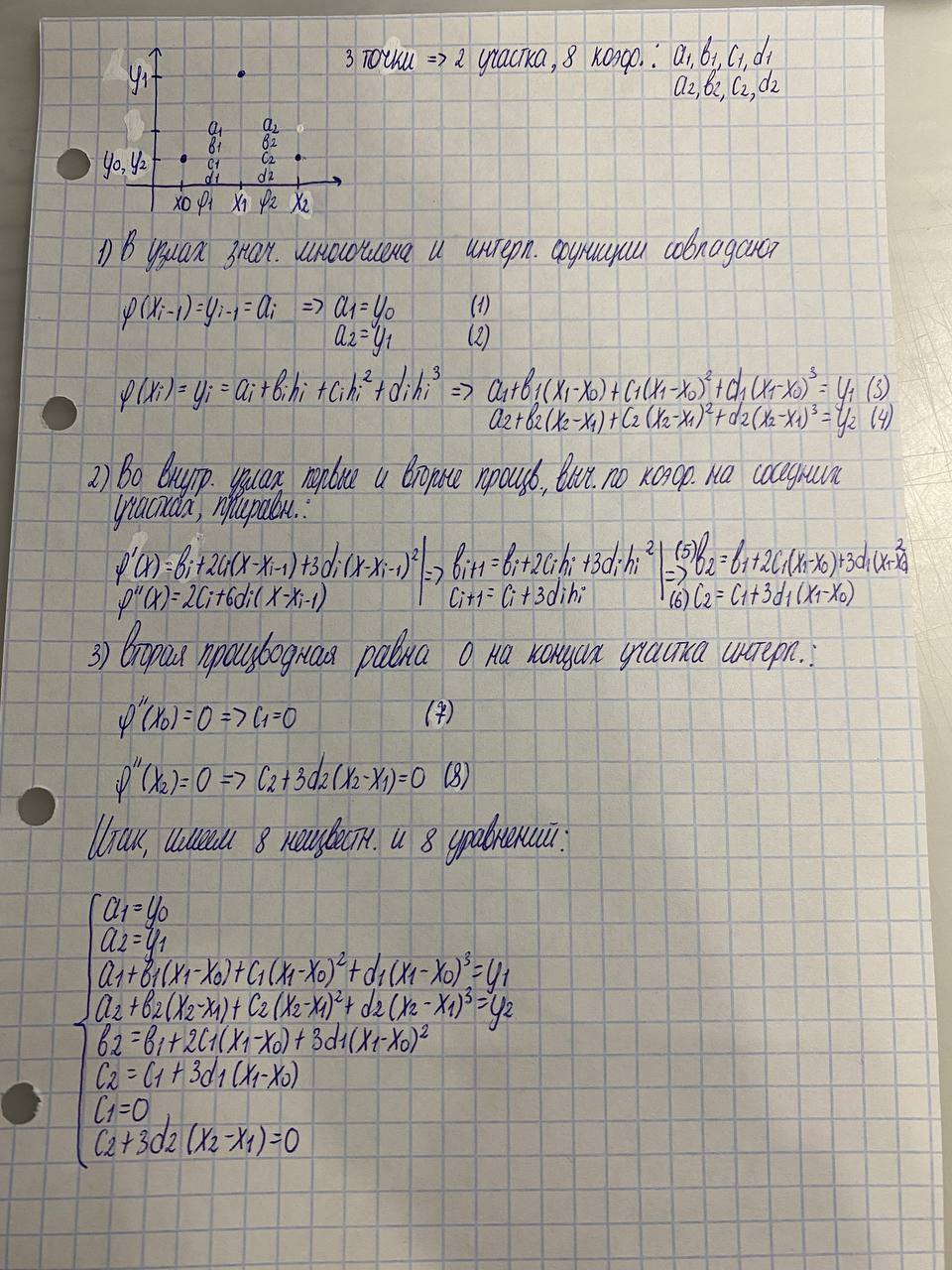
1. Результаты работы
   1. Значения y(x).
   2. Сравнить результаты интерполяции кубическим сплайном и полиномом Ньютона 3-ей степени



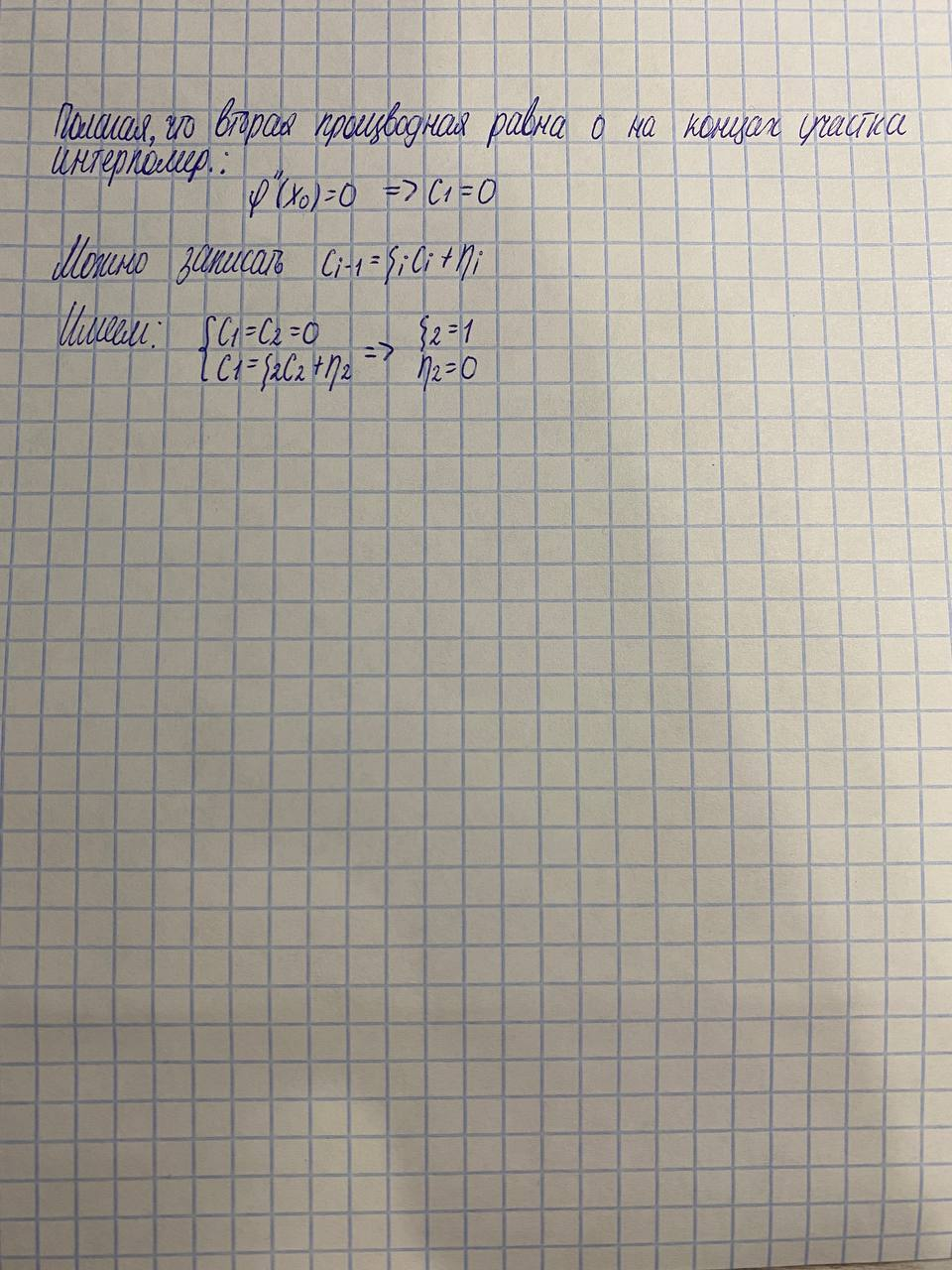
1. Вопросы при защите лабораторной работы
   1. Получить выражения для коэффициентов кубического сплайна, построенного на двух точках.



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



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



* 1. 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.

