

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное   
образовательное учреждение высшего образования

«Московский государственный технологический университет «СТАНКИН»

(ФГБОУ ВО «МГТУ «СТАНКИН»)

|  |  |
| --- | --- |
| Институт  информационных систем и технологий | Кафедра прикладной математики |

ОТЧЕТ О ВЫПОЛНЕНИИ

ЛАБОРАТОРНОЙ РАБОТЫ № 2

ПО ДИСЦИПЛИНЕ «ВЫЧИСЛИТЕЛЬНАЯ МАТЕМАТИКА»

НА ТЕМУ «ИНТЕРПОЛИРОВАНИЕ КУБИЧЕСКИМ СПЛАЙНОМ ДЕФЕКТА 1»

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| СТУДЕНТ(КИ) | | 2 | | КУРСА | бакалавриата | ГРУППЫ | ИДБ-22-14 |
|  | *(уровень профессионального образования)* | | | | | | |
| Набойщикова Артемия Андреевича | | | | | | | |
| *(Фамилия Имя Отчество)* | | | | | | | |
| Направление: | | | 09.03.04 «Программная инженерия» | | | | |
| Профиль подготовки: | | | «Системный анализ и проектирование программных комплексов» | | | | |

| Отчет сдан: | «5» мая 2024 г. | | |
| --- | --- | --- | --- |
| Проверил: | Москалев П.В., профессор, д.ф.-м.н. |  |  |
|  | *(Фамилия И.О. должность/звание, степень)* |  | *(Подпись)* |

МОСКВА 2024

**Лабораторная работа № 2: «Интерполирование кубическим сплайном дефекта 1»**

**Цель работы:** изучить метод интерполяции кубическим сплайном дефекта 1 и применить его на практике для получения заданной сплайн-функции *f*(*x*) на отрезке [*a*; *b*].

**Постановка задачи.** Пусть на отрезке [*а*; *b*] действительной оси существует некоторая непрерывная функция *y* = *f*(*x*), значения которой известны лишь в *n*+1 точке данного отрезка, которые обозначим через *x*0 = *a*, *x*1 = *a* + *h*, …, *xn* = *b*, где ℎ = (*b*–*a*)/*n*. Требуется найти для каждых двух соседних точек (узлов) *xi* и *xi*+1 кубический полином, аппроксимирующий данную функцию в каждой точке интервала (*xi*; *xi*+1), значения которого совпадают со значениями функции на концах интервала.

**Задание на лабораторную работу**

1. По номеру варианта *N* для функции *y* = *f*(*x*) выбрать отрезок [*a*; *b*] и разбить его на пять подотрезков.
2. Задать интерполируемую функцию *y* = *f*(*x*) таблично и составить систему линейных алгебраических уравнений для нахождения коэффициентов кубического полинома.
3. Составить блок-схему и написать реализацию метода матричной прогонки.
4. Построить функциональные зависимости интерполируемой функции *y* = *f*(*x*) и построенного кубического сплайна *y* = *s*(*x*) и оценить погрешность сплайн-интерполяции.
5. Провести анализ выполненной лабораторной работы и сделать выводы.

**Выполнение лабораторной работы**

**1. По номеру варианта *N* = 15** выберем интерполируемую функцию

и отрезок интерполяции [1; 6], который разобьём на пять подотрезков:

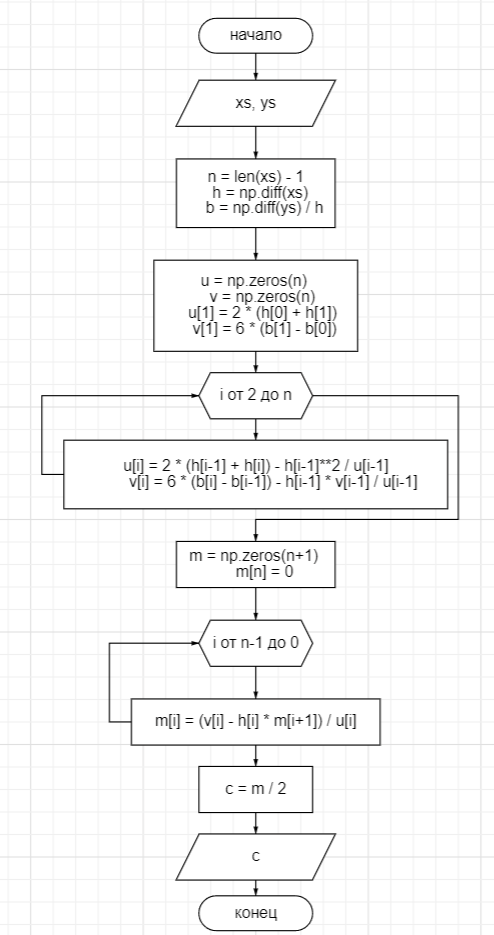
[1; 2], [2; 3], [3; 4], [4; 5], [5; 6].

**2. Зададим интерполируемую функцию** таблично

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| *xi* | 1 | 2 | 3 | 4 | 5 | 6 |
| *f*(*xi*) | 0.882496902 | 0.882496902 | 0.324652467 | 0.043936933 | 0.002187491 | 4.00652e-5 |

Составим систему линейных алгебраических уравнений для нахождения коэффициентов кубического полинома

**3. Блок-схема алгоритма** матричной прогонки



Листинг реализации метода матричной прогонки на языке Python

def cubic\_spline\_interpolation\_my(xs, ys):

    n = len(xs) - 1  # количество точек

    h = np.diff(xs)  # вычисляем шаги между точками

    b = np.diff(ys) / h  # вычисляем угловые коэффициенты

    # инициализируем массивы для метода прогонки

    u = np.zeros(n)

    v = np.zeros(n)

    u[1] = 2 \* (h[0] + h[1])

    v[1] = 6 \* (b[1] - b[0])

    # прямой проход метода прогонки

    for i in range(2, n):

        u[i] = 2 \* (h[i-1] + h[i]) - h[i-1]\*\*2 / u[i-1]

        v[i] = 6 \* (b[i] - b[i-1]) - h[i-1] \* v[i-1] / u[i-1]

    # обратный проход метода прогонки

    m = np.zeros(n+1)

    m[n] = 0

    for i in range(n-1, 0, -1):

        m[i] = (v[i] - h[i] \* m[i+1]) / u[i]

    coefficients = []

    for i in range(n):

        a = ys[i]

        b = (ys[i+1] - ys[i]) / h[i] - h[i] \* (2 \* m[i] + m[i+1]) / 6

        c = m[i] / 2

        d = (m[i+1] - m[i]) / (6 \* h[i])

        coefficients.append((a, b, c, d))

    # функция для интерполяции в каждом отрезке

    def S(x, i):

        a, b, c, d = coefficients[i]

        t = x - xs[i]

        return a + b\*t + c\*t\*\*2 + d\*t\*\*3

    return coefficients, S

coefficients, interpolated\_function = cubic\_spline\_interpolation\_my(xs, ys)

# Печать коэффициентов для каждого отрезка

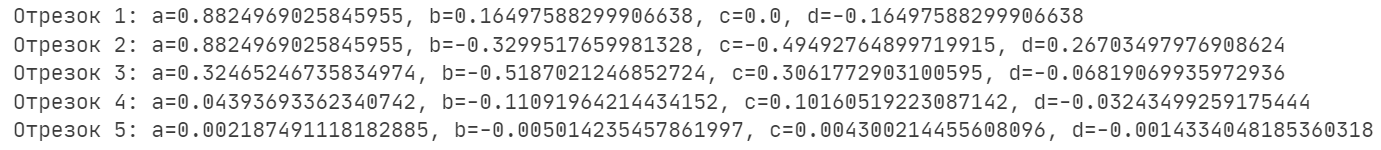
for i, coeffs in enumerate(coefficients):

    print(f'Отрезок {i+1}: a={coeffs[0]}, b={coeffs[1]}, c={coeffs[2]}, d={coeffs[3]}')

x\_interp\_my = np.linspace(xs[0] + 0.001, xs[-1], 100)

y\_interp\_my = [interpolated\_function(x, np.searchsorted(xs, x) - 1) for x in x\_interp\_my]

Коэффициенты для отрезков интерполированной кривой:



**4. Построение зависимостей** интерполируемой функции и сплайн-функции на языке Python

xs\_plot = np.linspace(1 + 0.001, 6, 100)

ys\_plot = [variant\_function(x) for x in xs\_plot]

xs\_interp = np.linspace(1 + 0.001, 6, 100)

ys\_interp = [interpolated\_function(x, np.searchsorted(xs, x) - 1) for x in xs\_interp]

abs\_errs = [abs(ys\_plot[i] - ys\_interp[i]) for i in range(len(ys\_plot))]

rel\_errs = [abs(abs\_errs[i] / ys\_plot[i])  for i in range(len(ys\_plot))]

square\_diffs = [diff \*\* 2 for diff in abs\_errs]

std\_err = sum(square\_diffs) / len(square\_diffs) # среднеквадратическое отклонение

print(abs\_errs)

print(f"Среднеквадратическое отклонение: {std\_err}")

plt.figure()

plt.plot(xs, ys, 'o', label='Исходные точки')

plt.plot(xs\_plot, ys\_plot, label='Исходная функция f(x)')

plt.plot(xs\_interp, ys\_interp, label='Интерполированная кривая S(x)')

plt.plot(xs\_plot, rel\_errs, label="Относительная погрешность, %")

plt.xlabel('X')

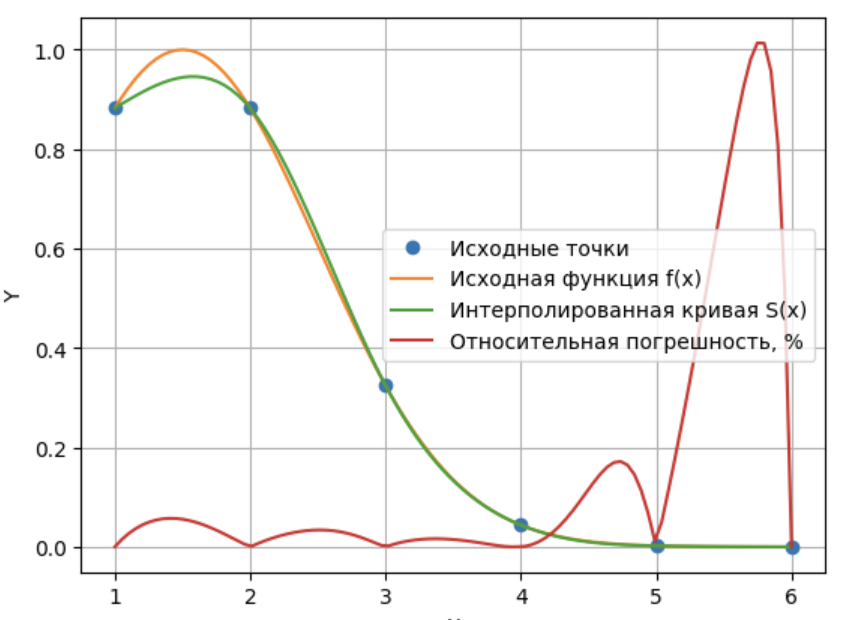
plt.ylabel('Y')

plt.legend()

plt.grid(True)

plt.show()





**5. Анализ выполненной работы и выводы**

Был изучен метод интерполяции кубическим сплайном и применён на практике для получения сплайна функции f(x). Полученное среднеквадратичное отклонение позволяет сказать, что сплайн обладает достаточно высокой степенью точности, хотя отрезок [1; 2]

не удалось правильно интерполировать ввиду недостатка информации о вершине.