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-22-15

(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 $[a; b]$ действительной оси существует некоторая непрерывная функция $y = f(x)$, значения которой известны лишь в n+1 точке данного отрезка, которые обозначим через $x_0 = a$, $x_1 = a + h$, …, $x_n = b$, где $h = (b–a)/n$. Требуется найти для каждых двух соседних точек (узлов) x_i и x_{i+1} кубический полином, аппроксимирующий данную функцию в каждой точке интервала $(x_i; x_{i+1})$, 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

$$e^{-\frac{(x-\frac{15}{10})^2}{2}}$$

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

x_i	1	2	3	4	5	6
$f(x_i)$	0.882496902	0.882496902	0.324652467	0.043936933	0.002187491	$4.00652e^{-5}$

Составим систему линейных алгебраических уравнений для нахождения коэффициентов кубического полинома

$$a_i = y_{i-1},$$

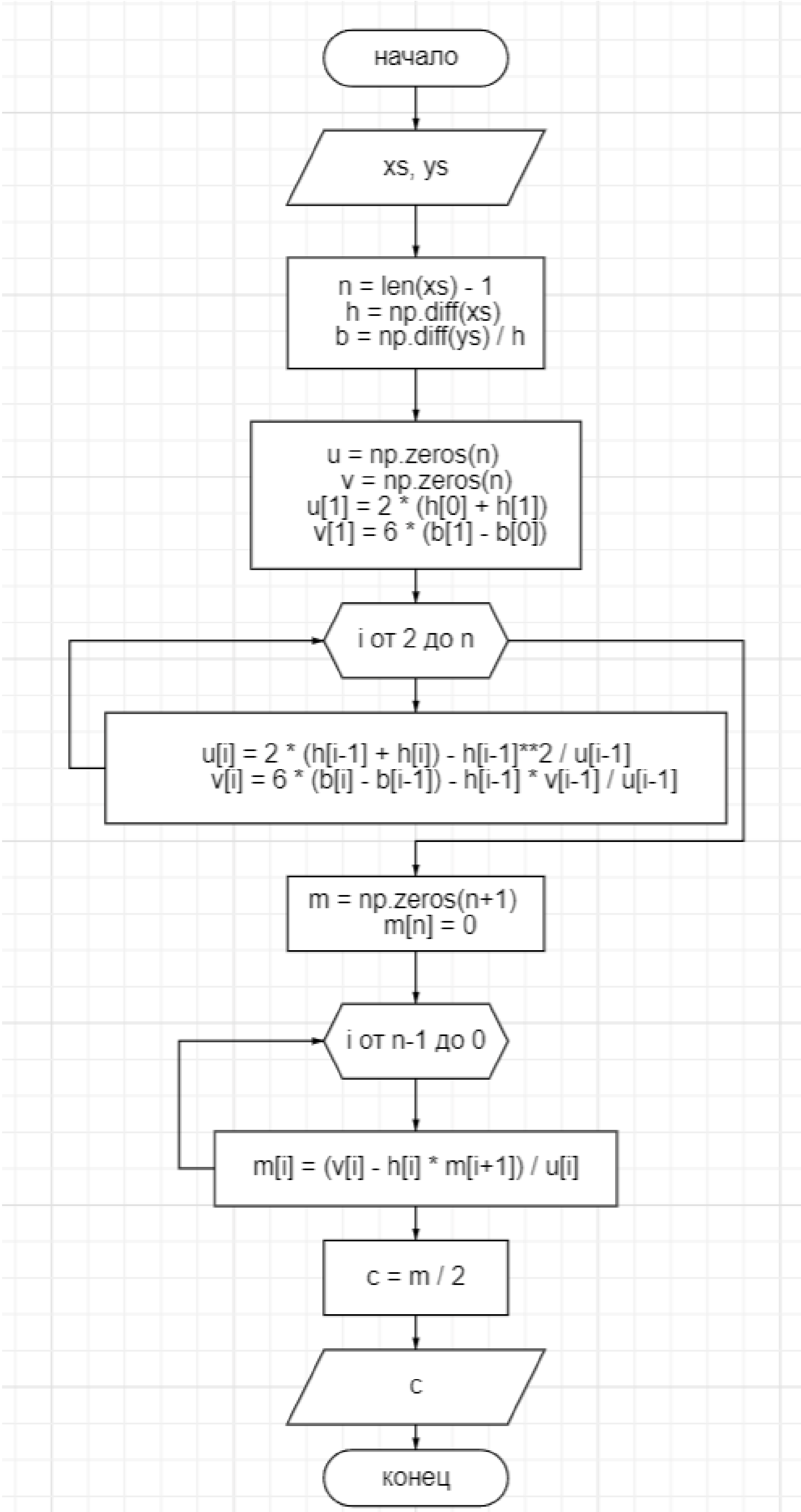
$$h_i c_i + 2(h_i + h_{i+1})c_{i+1} + h_{i+1}c_{i+2} = 3\left(\frac{y_{i+1} - y_i}{h_{i+1}} - \frac{y_i - y_{i-1}}{h_i}\right), c_1 = c_{n+1} = 0,$$

$$d_i = \frac{c_{i+1} - c_i}{3h_i}$$

$$b_i = \left[\frac{(y_i - y_{i-1})}{h_i}\right] - \frac{1}{3}h_i(c_{i+1} + 2c_i),$$

$$i = \overline{1, n-1}$$

3. Блок-схема алгоритма матричной прогонки



Листинг реализации метода матричной прогонки на языке Python

```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]
```

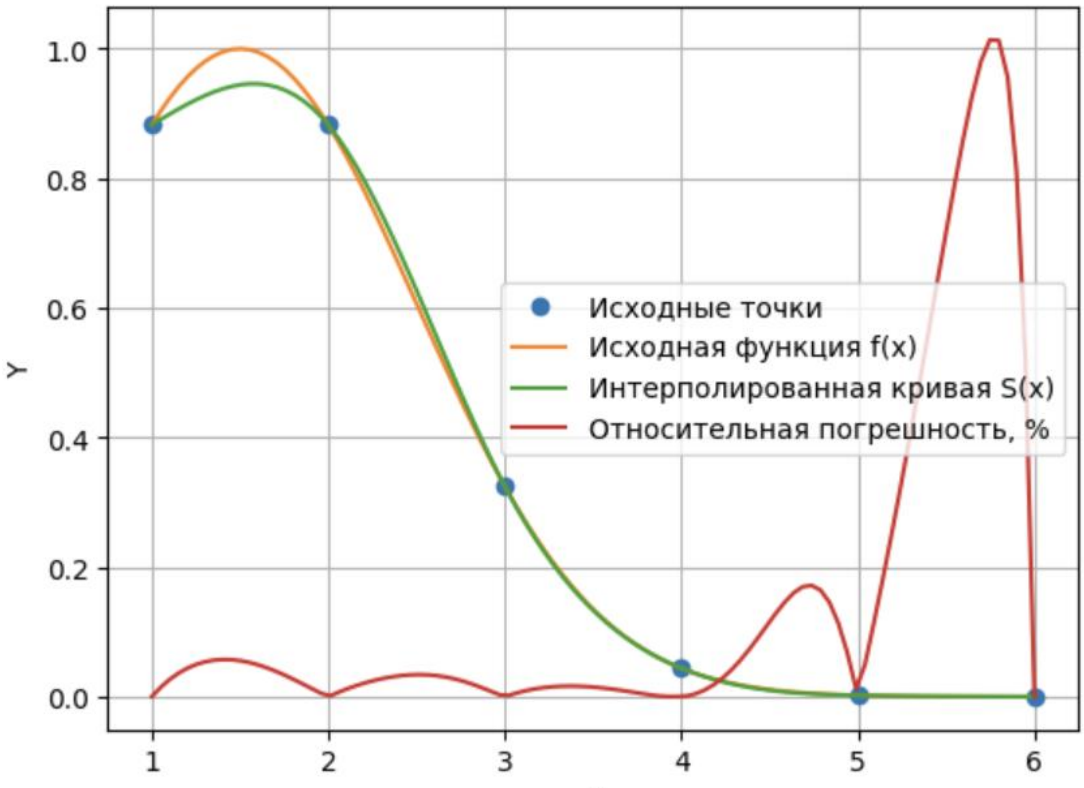
Коэффициенты для отрезков интерполированной кривой:

```
Отрезок 1: a=0.8824969025845955, b=0.16497588299906638, c=0.0, d=-0.16497588299906638
Отрезок 2: a=0.8824969025845955, b=-0.3299517659981328, c=-0.49492764899719915, d=0.26703497976908624
Отрезок 3: a=0.32465246735834974, b=-0.5187021246852724, c=0.3061772903100595, d=-0.06819069935972936
Отрезок 4: a=0.04393693362340742, b=-0.11091964214434152, c=0.10160519223087142, d=-0.03243499259175444
Отрезок 5: a=0.002187491118182885, b=-0.005014235457861997, c=0.004300214455608096, d=-0.0014334048185360318
```

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()
```

```
Среднеквадратическое отклонение: 0.00036667797473491365
```



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.