Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Теоретическая информатика и компьютерные технологии»

Лабораторная работа №1

по курсу «Численные методы»

«Приближение функции кубическими сплайнами»

Студент: Шемякин В.А.

Группа: ИУ9-62Б

Преподаватель: Домрачева А.Б.

Москва 2025

1 Постановка задачи

Дано: функция $y = f(x)$ задана конечным набором точек

$y_i = f(x_i), \quad i = \overline{0, n}$ на отрезке $[a, b]$, $a = x_0, \quad b = x_n, \quad x_i = a{+}ih$, $h = \dfrac{(b-a)}{n}$

x_i	x_0	x_1	$\ldots$	x_{n-1}	x_n
y_i	y_0	y_1	$\ldots$	y_{n-1}	y_n

Найти: интерполяционную функцию $y = g(x)$: $g(x_i) = f(x_i), \quad i = \overline{0, n}$ (т.е. функцию, совпадающую со значениями $y_i = f(x_i), \quad i = \overline{0, n}$ в узлах интерполяции $x_i, \quad i = \overline{0, n}$):

1. Протабулировать функцию $f(x)$ на отрезке $[a, b]$ с шагом $h = \frac{(b-a)}{n}$, где $n = 32$ и распечатать таблицу (x_i, y_i), $i = \overline{0, n}$.
2. Для заданных узлов (x_i, y_i) построить кубический сплайн (распечатать массивы a, b, c, d).
3. Вычислить значения $f(x)$ в точках $x_i' = a + \left(i - \frac{1}{2}\right) h$
4. Вычислить значения изначальной функции и сплайна в точке, задаваемой с экрана.

Индивидуальный вариант (№5): $y = f(x)$ задана функцией: $y = x * \sqrt{x+1}$ на отрезке $[0, 3]$.

2 Основные теоретические сведения

2.1 Метод Гаусса

Метод Гаусса используется для решения систем линейных уравнений. Этот метод преобразует систему в верхнетреугольный вид, что позволяет последовательно находить решения для каждой переменной.

Прямой ход: на первом этапе метод Гаусса преобразует систему к верхнетреугольному виду. Это достигается путем последовательного приведения матрицы коэффициентов к ступенчатому виду, где все элементы ниже главной диагонали равны нулю.

Обратный ход: после приведения матрицы к верхнетреугольному виду, производится обратный ход, на котором находятся неизвестные. Начинают с последнего уравнения и поднимаются вверх, подставляя найденные значения в предыдущие уравнения.

Условия применимости: метод Гаусса применим, если все ведущие элементы (элементы на главной диагонали) не равны нулю.

2.2 Сплайн-интерполяция

Интерполяционной функцией называется функция $y = g(x)$, проходящая через заданные точки, называемые узлами интерполяции:

$$g(x_i) = f(x_i), \quad i = \overline{0, n}.$$

При этом в промежуточных точках равенство выполняется с некоторой погрешностью

$$g(x_i^*) \approx f(x_i^*).$$

Задача интерполяции заключается в поиске такой функции $y = g(x)$.

Приближение функции кубическим сплайном — пример задачи интерполяции.

Сплайн k-го порядка — функция, проходящая через все узлы (x_i, y_i), $i = \overline{0, n}$, являющаяся многочленом k-ой степени на каждом частичном отрезке разбиения $[x_i, x_{i+1}]$, $x_i = a + ih$, $h = \frac{(b-a)}{n}$, $x_i \in [a, b]$ и имеющая первые p непрерывных на $[a, b]$ производных. $d = k - p$ — дефект сплайна.

Наиболее употребительны сплайны третьего порядка с дефектом $d = 1$ (кубические сплайны).

На каждом частичном отрезке разбиения кубический сплайн описывается

$$S_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3$$

$$x \in [x_i, x_{i+1}], \quad i = \overline{0, n-1}$$

На частные многочлены накладываются условия:

1. Сплайн проходит через все узлы

$$S_i(x_i) = y_i, \quad i = \overline{0, n-1}; \quad S_{n-1}(x_n) = y_n$$

2. Условие гладкости на краях

$$S_0''(x_0) = 0; \quad S_{n-1}''(x_n) = 0$$

3. Непрерывность сплайна и его первых двух производных в промежуточных узлах

$$S_{i-1}'(x_i) = S_i'(x_i);$$

$$S_{i-1}''(x_i) = S_i''(x_i);$$

$$i = \overline{0, n-1}$$

Эти условия позволяют выразить коэффициенты a_i, b_i, d_i и приводят к трехдиагональной СЛАУ относительно коэффициента c_i:

$$a_i = y_i, \quad i = \overline{0, n-1};$$

$$b_i = \frac{y_{i+1} - y_i}{h} - \frac{h}{3}(c_{i+1} + 2c_i), \quad i = \overline{0, n-2};$$

$$b_{n-1} = \frac{y_{n-1} - y_{n-1}}{h} - \frac{2h}{3}c_{n-1};$$

$$d_i = \frac{c_{i+1} - c_i}{3h}, \quad i = \overline{0, n-2};$$

$$d_{n-1} = -\frac{c_n}{3h}$$

СЛАУ с трехдиагональной матрицей относительно коэффициента c_i:

$$c_{i-1} + 4c_i + c_{i+1} = \frac{3(y_{i+1} - 2y_i + y_{i-1})}{h^2}, \quad i = \overline{1, n-1};$$

$$c_0 = c_n = 0,$$

где $h = x_{i+1} - x_i$, $i = \overline{0, n-1}$ - постоянный шаг интерполяции.

3 Реализация

```
import copy


def solve(A, B):
    n = len(A)

    for i in range(n):
        max_row = i
        for k in range(i + 1, n):
            if abs(A[k][i]) > abs(A[max_row][i]):
                max_row = k

        A[i], A[max_row] = A[max_row], A[i]
        B[i], B[max_row] = B[max_row], B[i]

        for k in range(i + 1, n):
            c = -A[k][i] / A[i][i]
            for j in range(i, n):
                if i == j:
                    A[k][j] = 0
                else:
                    A[k][j] += c * A[i][j]
            B[k] += c * B[i]

    X = [0 for _ in range(n)]
    for i in range(n - 1, -1, -1):
        X[i] = B[i] / A[i][i]
        for k in range(i - 1, -1, -1):
            B[k] -= A[k][i] * X[i]

    return X
```

Листинг 1: Метод Гаусса

```
import csv
from math import sqrt

def count_f(x):
    return x * sqrt(x + 1)

def count_spline(x, A, B, C, D, X, i):
    return A[i] + B[i] * (x - X[i]) + C[i] * (x - X[i]) ** 2 + D[i] * (x - X[i]) ** 3

def solve(A, B):
```

```
    n = len(A)

    for i in range(n):
        max_row = i
        for k in range(i + 1, n):
            if abs(A[k][i]) > abs(A[max_row][i]):
                max_row = k

        A[i], A[max_row] = A[max_row], A[i]
        B[i], B[max_row] = B[max_row], B[i]

        for k in range(i + 1, n):
            c = -A[k][i] / A[i][i]
            for j in range(i, n):
                if i == j:
                    A[k][j] = 0
                else:
                    A[k][j] += c * A[i][j]
            B[k] += c * B[i]

    X = [0 for _ in range(n)]
    for i in range(n - 1, -1, -1):
        X[i] = B[i] / A[i][i]
        for k in range(i - 1, -1, -1):
            B[k] -= A[k][i] * X[i]

    return X

def count_c(A, B, C, D):
    n = len(D)
    matrix = [[0.0] * n for _ in range(n)]
    for i in range(n):
        if i > 0:
            matrix[i][i - 1] = A[i - 1]
        matrix[i][i] = B[i]
        if i < n - 1:
            matrix[i][i + 1] = C[i]
    return solve(matrix, D)

def write_to_csv(filename, headers, data):
    with open(filename, 'w', newline='') as file:
        writer = csv.writer(file)
        writer.writerow(headers)
        writer.writerows(data)

def main():
```

```python
    a, b = 0, 3
    n = 32
    h = (b - a) / n
    X = [a + i * h for i in range(n)]
    Y = list(map(count_f, X))
    coef_a = [0] * (n-1)
    coef_b = [0] * n
    coef_c = [0] * (n-1)
    coef_d = [0] * n
    coef_b[0] = 1.0
    coef_b[n-1] = 1.0
    for i in range(1, n-1):
        coef_a[i - 1] = h
        coef_b[i] = 4 * h
        coef_c[i] = h
        coef_d[i] = 3 * ((Y[i+1] - Y[i]) / h - (Y[i] - Y[i-1]) / h)
    C = count_c(coef_a, coef_b, coef_c, coef_d)

    A = [Y[i] for i in range(n)]
    B = [((Y[i + 1] - Y[i]) / h) - h / 3 * (C[i + 1] + 2 * C[i])
         for i in range(n - 1)]
    B.append(((Y[n-1] - Y[n-2]) / h) - (2 * h * C[n-2] / 3))
    D = [(C[i + 1] - C[i]) / (3 * h) for i in range(n - 1)]
    D.append(C[n-1] / (3 * h))
    headers = ["i", "X", "A", "B", "C", "D"]
    data = []
    for i in range(n):
        row = [i, X[i], A[i], B[i], C[i], D[i]]
        data.append(row)

    write_to_csv('output.csv', headers, data)

    x = float(input("Координаты точки: "))
    print("Значение функции:", count_f(x))
    if x == 3:
        k = n - 1
    else:
        k = 0
        while True:
            if k < n - 1:
                if X[k] <= x <= X[k+1]:
                    break
            else:
                break
            k += 1
    c_s = count_spline(x, A, B, C, D, X, k)
```

```
    print(f"Интерполированное значение: {c_s}")
    print(f"Погрешность: {abs(c_s - count_f(x))}")

    x_vals = [a + (i - 0.5) * h for i in range(n)]
    y_vals = [count_f(x) for x in x_vals]
    spline_vals = [count_spline(x_vals[i], A, B, C, D, X, i)
                   for i in range(len(x_vals))]
    p = [abs(spline_vals[i] - y_vals[i]) for i in range(len(x_vals))]

    additional_headers = ["i", "x_i", "f(x_i)", "S(x_i)", "inaccuracy"]
    additional_data = []
    for i in range(len(x_vals)):
        row = [i, x_vals[i], y_vals[i], spline_vals[i], p[i]]
        additional_data.append(row)

    write_to_csv('additional_output.csv', additional_headers, additional_data)

if __name__ == "__main__":
    main()
```

Листинг 2: Сплайн-интерполяция

4 Результаты

Для заданных узлов интерполяции (x_i, y_i) построен кубический сплайн с коэффициентами, представленными на рисунке 1.

i	X	A	B	C	D
0	0.0	0.0	1.027035644942393	0.0	2.137814838067372
1	0.09375	0.09804609685946197	1.0834038096179974	0.6012604232064483	-0.6994529497155418
2	0.1875	0.2043233879784691	1.1776975318966287	0.4045392810989522	0.07928781158838147
3	0.28125	0.31835338376264355	1.2556392436972978	0.4268389781081845	-0.11480915888832202
4	0.375	0.4397264774834465	1.3326443574734568	0.39454890217084393	-0.05111524750117072
5	0.46875	0.5680874832909761	1.4052745113155176	0.38017273881113967	-0.058655556206838905
6	0.5625	0.703125	1.4750103177941838	0.3636758636279662	-0.04876022279744675
7	0.65625	0.8445634100028399	1.5419138722873855	0.3499620509661843	-0.04482203542498967
8	0.75	0.9921567416492215	1.6063499258313627	0.337355853502906	-0.04030336061216606
9	0.84375	1.1456838799312727	1.6685414620970165	0.3260205333307343	-0.03675393387647949
10	0.9375	1.3049447725382863	1.7287012142306455	0.3156834894279744	-0.03360464514011537
11	1.03125	1.4697573839168574	1.7870058085191103	0.30623218298231697	-0.03088910232990136
12	1.125	1.6399552204252408	1.8436098840754556	0.2975446229520322	-0.0285052378465021
13	1.21875	1.8153852977527594	1.8986478979279307	0.2895275248077035	-0.026406687969804093
14	1.3125	1.9959064552541033	1.9522380387364213	0.2821006438161961	-0.024546862091465133
15	1.40625	2.1813879455843304	2.004484677736656	0.2751968388529715	-0.022890526144722954
16	1.5	2.3717082451262845	2.0554805262267566	0.2687588783747682	-0.02140821396275842
17	1.59375	2.5667540432147713	2.1053083415304292	0.2627378181977424	-0.020075738266699023
18	1.6875	2.7664193774399064	2.154042341687427	0.2570915168102333	-0.018873281120001195
19	1.78125	2.970604889278129	2.2017493657473146	0.25178340649523295	-0.017783100295980167
20	1.875	3.1792171795899695	2.248489864125335	0.24678190953698853	-0.01679468475042572
21	1.96875	3.392168247584703	2.2943186435617027	0.2420584044509313	-0.01588260588744392
22	2.0625	3.609375	2.3392858147488296	0.2375914215450877	-0.015086968261135722
23	2.15625	3.8307588197073468	2.3834364053675854	0.23334821172164327	-0.014211397440414133
24	2.25	4.056245184896988	2.4268144804844454	0.2293512561915268	-0.013948762557893643
25	2.34375	4.285763331541346	2.4694500513825997	0.2254281667221192	-0.011678795700815413
26	2.4375	4.51924595307392	2.5114098956469793	0.22214350543126488	-0.017150679541300957
27	2.53125	4.756628932219553	2.5526095877321233	0.21731987681027398	0.006045907138725721
28	2.625	4.9978511007231905	2.593516478201184	0.2190202881930406	-0.0779516363278365
29	2.71875	5.242854023387155	2.6325274168263917	0.19709639047583657	0.2379225774555737
30	2.8125	5.491581803371866	2.6757563392508654	0.2640121153852167	-0.9387097435918815
31	2.90625	5.74398090616522	2.6757563392508654	0.0	0.0

Рисунок 1 : Коэффициенты кубического сплайна

Значения функции и результаты интерполяции в точках x'_i представлены на рисунке 2.

i	x_i	f(x_i)	S(x_i)	inaccuracy
0	-0.046875	-0.04576318169476555	-0.04836248399993875	0.002599302305173197
1	0.046875	0.047961051397690135	0.04865471357708007	0.0006936621793899381
2	0.140625	0.1501875658356597	0.14999953003418787	0.0001880358014718364
3	0.234375	0.26039632081979264	0.2604449978073209	4.8676987528251114e-05
4	0.328125	0.37814537813115356	0.37813096667526136	1.4411455892204827e-05
5	0.421875	0.5030538757472174	0.5030566223997397	2.7466525223118765e-06
6	0.515625	0.6347896630063895	0.6347880059753811	1.6570310084063067e-06
7	0.609375	0.7730600996847589	0.7730597733514025	3.263333563374715e-07
8	0.703125	0.9176050642597163	0.917604500414398	5.638453183243186e-07
9	0.796875	1.0681915430330937	1.0681911381449456	4.048881481200084e-07
10	0.890625	1.224609375	1.2246090047727192	3.7022728083258016e-07
11	0.984375	1.386667857780923	1.386667541570963	3.1620995999404045e-07
12	1.078125	1.5541930059224316	1.5541927276301473	2.782922843280744e-07
13	1.171875	1.727025310920644	1.7270250662201894	2.44700454565816e-07
14	1.265625	1.905017892348	1.905017675484669	2.1686333084858234e-07
15	1.359375	2.0880349575931096	2.0880347645210873	1.9307202236973353e-07
16	1.453125	2.2759505078343265	2.275950335082364	1.727519625127627e-07
17	1.546875	2.46864724247181	2.4686470872191704	1.5525263963311886e-07
18	1.640625	2.666015625	2.6660154849137294	1.400862705835948e-07
19	1.734375	2.8679530813297807	2.8679529543856512	1.2694412943048405e-07
20	1.828125	3.0743633076320784	3.0743631924150177	1.1521706078099214e-07
21	1.921875	3.2851556694039323	3.285155563637088	1.0576684417173965e-07
22	2.015625	3.500244677029345	3.500244582806852	9.422249291901608e-08
23	2.109375	3.719549525888466	3.7195494299388443	9.594962158132603e-08
24	2.203125	3.9429936912538395	3.942993638433832	5.2820007478970865e-08
25	2.296875	4.170504569943536	4.170504388823322	1.81120213937902e-07
26	2.390625	4.40201316208282	4.402013488969328	3.26886508084101e-07
27	2.484375	4.637453787437157	4.637452244580307	1.5428568502784401e-06
28	2.578125	4.876763831678822	4.876769290329071	5.4586502491460465e-06
29	2.671875	5.119883518682143	5.11986286857667	2.0650105473585256e-05
30	2.765625	5.366755705543016	5.366832513839488	7.680829647149778e-05
31	2.859375	5.617325697513307	5.618554827762836	0.0012291302495288647

Рисунок 2 : Значения функции и сплайна в точках x_i'

5 Вывод

В ходе лабораторной работы был изучен метод аппроксимации функции при помощи кубической сплайн-интерполяции. На основе заданной функции и узлов интерполяции был построен сплайн третьего порядка. По итогам тестирования было определено, что значения заданной функции и сплайна практически не отличаются, что подтверждает корректность работы данного метода. Таким образом, кубическая сплайн-интерполяция показывает высокую точность в узлах, однако следует учитывать некоторые погрешности в промежуточных точках между узлами.