



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

---

ФАКУЛЬТЕТ Информатики и систем управления

КАФЕДРА Теоретической информатики и компьютерных технологий

## **Лабораторная работа № 1**

«Приближение функции кубическими сплайнами»

по курсу «Численные методы»

Выполнила:

студент группы ИУ9-61Б

Окутин Денис

Проверила:

Домрачева А. Б.

Москва, 2024

## 1. Цель

Работа направлена на исследование аппроксимации заданной функции методом сплайн-интерполяции, построение сплайна третьего порядка, используя заданные точки (узлы интерполяции), и вычисление его значений в серединах частичных отрезков между узлами интерполяции.

### Постановка задачи

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

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

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

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

1. Для заданных узлов  $(x_i, y_i)$  построить кубический сплайн (распечатать массивы a, b, c, d).
2. Вычислить значения  $f(x)$  в серединах частичных отрезков между узлами интерполяции, т.е. в точках  $x_i^* = a + \left(i - \frac{1}{2}\right)h$ ,  $h = \frac{(b-a)}{n}$ .

**Индивидуальный вариант:**  $y = f(x)$  задана функцией:  $y = \ln(2x)$  на отрезке  $[0.5, 2 \cdot e]$ .

## 2. Основные теоретические сведения

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

$g(x_i) = f(x_i)$ ,  $i = \overline{0, n}$ . 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. Чем выше дефект, тем грубее сплайн.

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

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

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  $c_i$ :

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

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

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

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

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

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

### 3. Реализация

Листинг 1. Сплайн-интерполяция

```
import math

SIZE = 8

def f(x):
    return math.log(2 * x, math.e)

def forward(a, b, c, d):
    if b[0] == 0 or b[len(b) - 1] == 0:
        raise Exception("invalid b data, cant
calculate forward")
    if abs(c[0]) / abs(b[0]) > 1 or abs(a[len(b) - 1
- 1]) / abs(b[len(b) - 1]) > 1:
        raise Exception("invalid matrix data, cant
calculate forward")
```

```

    y = [0.0] * len(b)
    alpha = [0.0] * len(b)
    beta = [0.0] * len(b)
    y[0] = b[0]
    alpha[0] = -c[0] / b[0]
    beta[0] = d[0] / b[0]

    n = len(b) - 1
    for i in range(len(b)):
        if i > 1 and i < n and abs(b[i]) < abs(a[i -
1]) + abs(c[i]):
            raise Exception("invalid matrix data,
cant calculate forward")

        if i == 0:
            continue
        elif i == n:
            y[n] = b[n] + a[n - 1] * alpha[n - 1]
            beta[n] = (d[n] - a[n - 1] * beta[n - 1])
/ y[n]
        else:
            y[i] = b[i] + alpha[i - 1] * a[i - 1]
            alpha[i] = -c[i] / y[i]
            beta[i] = (d[i] - a[i - 1] * beta[i - 1])
/ y[i]

    return alpha, beta

def backward(alpha, beta):

```

```

x = [0] * len(beta)
n = len(beta) - 1
x[n] = beta[n]
for i in range(n - 1, -1, -1):
    x[i] = alpha[i] * x[i + 1] + beta[i]

return x

def main():
    l, r = 0.5, 2 * math.e
    h = (r - l) / SIZE
    xs = []
    ys = []
    i = l
    while i < r:
        xs.append(i)
        ys.append(f(i))
        i += h

    if len(xs) < SIZE + 1:
        xs.append(r)
        ys.append(f(r))

    for i in range(SIZE + 1):
        print(xs[i], ";", ys[i])

    print()

    d = []

```

```

    for i in range(1, SIZE):
        d.append(3 * (ys[i + 1] - 2 * ys[i] + ys[i -
1])) / (h * h))

    b = [4] * (SIZE - 1)
    a = [1] * (SIZE - 2)
    c = [1] * (SIZE - 2)
    alpha, beta = forward(a, b, c, d)
    coefC = backward(alpha, beta)
    coefC.insert(0, 0)
    coefC.append(0)

    coefA = []
    for i in range(SIZE):
        coefA.append(ys[i])

    coefB = []
    for i in range(1, SIZE + 1):
        coefB.append((ys[i] - ys[i - 1]) / h - (h /
3) * (coefC[i] + 2 * coefC[i - 1]))

    coefD = []
    for i in range(1, SIZE + 1):
        coefD.append((coefC[i] - coefC[i - 1]) / (3 *
h))

    for i in range(SIZE):
        var_x = l + i * h
        var_y = ys[i]
        s = coefA[i] + coefB[i] * (var_x - xs[i]) +

```

```

coefC[i] * ((var_x - xs[i]) ** 2) + coefD[i] * (
    (var_x - xs[i]) ** 3)
    print(f'x: {var_x}, y: {var_y}, y*: {s}, |y-
y*|: {math.fabs(var_y - s)}')

print()
for i in range(SIZE):
    var_x = l + (i + 0.5) * h
    var_y = f(var_x)
    s = coefA[i] + coefB[i] * (var_x - xs[i]) +
coefC[i] * ((var_x - xs[i]) ** 2) + coefD[i] * (
    (var_x - xs[i]) ** 3)
    print(f'x: {var_x}, y: {var_y} y*: {s}, |y-
y*|: {math.fabs(var_y - s)}')

if __name__ == '__main__':
    main()

```

#### 4. Результаты

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

```

a: [0.0, 0.8038567757524159, 1.24365932103139, 1.5480778530710773, 1.7811304570373452, 1.9700039163314722, 2.128808383036289, 2.265815957825284]
b: [1.4469415609562306, 1.0142124601269649, 0.5424840603998891, 0.4340171109673103, 0.33446095217964084, 0.2794091989640917, 0.23820157835493477, 0.20592840497501266]
c: [0, -0.701263682031674, -0.0632007227831963, -0.11257652952562108, -0.04076024756327823, -0.04045444839373594, -0.026324993295680766, -0.025975636248021457, 0]
d: [-0.378813404501962, 0.3446732044881057, -0.026672160460298938, 0.03447271497473664, 0.004486683313484033, 0.007632545541136592, 0.00018871807567042865, 0.01403169656912753]

```

Рисунок 1. Коэффициенты сплайна.

Значения функции в узлах интерполяции и в серединах частичных отрезков между узлами интерполяции представлены в таблице 1.



Таблица 1 – Результаты программы

$x_i$	$y_i$
0.5	0.0
0.8	0.48061615238210714
1.1	0.8038567757524159
1.42	1.047743946552493
1.73	1.24365932103139
2.04	1.4074079634121703
2.35	1.5480778530710773
2.65	1.6713780356208636
2.96	1.7811304570373452
3.27	1.8800197223577078
3.58	1.9700039163314722
3.89	2.0525552001207563
4.2	2.128808383036289
4.51	2.199656721994441
4.8	2.265815957825284
5.12	2.327868443915036
5.43	2.3862943611198904

Значения погрешности представлены на рисунке 2 (последняя колонка).

```

x: 0.8085352285573806, y: 0.48061615238210714 y*: 0.43530643088315474, |y-y*|: 0.045309721498952404
x: 1.425605685672142, y: 1.047743946552493 y*: 1.060144255798603, |y-y*|: 0.01240030924611002
x: 2.0426761427869033, y: 1.4074079634121703 y*: 1.4042350558097594, |y-y*|: 0.003172907602410957
x: 2.6597465999016645, y: 1.6713780356208636 y*: 1.6722833006056734, |y-y*|: 0.000905264984809806
x: 3.2768170570164257, y: 1.8800197223577078 y*: 1.8798135379996197, |y-y*|: 0.00020618435808805735
x: 3.893887514131187, y: 2.0525552001207563 y*: 2.052584650344619, |y-y*|: 2.945022386269258e-05
x: 4.510957971245948, y: 2.199656721994441 y*: 2.1998015231620487, |y-y*|: 0.00014480116760751116
x: 5.128028428360709, y: 2.327868443915036 y*: 2.327291521665632, |y-y*|: 0.0005769222494040882

```

Рисунок 2. Результаты интерполяции в промежуточных точках.

## **5. Вывод**

В ходе лабораторной работы был изучен метод приближения функции при помощи интерполяции кубическим сплайном. Был построен сплайн третьего порядка на основе заданных узлов интерполяции и вычислены значения функции в серединах частичных отрезков между узлами интерполяции. После проведения тестирования был сделан вывод о точности данного метода приближения функции: значения функции и значения кубического сплайна совпадают в узлах интерполяции, но в серединах между узлами интерполяции наблюдается погрешность из-за возрастания погрешности при выполнении арифметических операций на компьютере, чем ближе значение было к краевым, тем выше была погрешность.