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 № 3

«Приближение функции кубическими сплайнами»

по курсу «Численные методы»

Выполнила:

студент группы ИУ9-61Б

Яровикова Анастасия

Проверила:

Домрачева А. Б.

Москва, 2023

1. Цель

Целью данной работы является изучение приближения заданной функции путем сплайн-интерполяции, построение сплайна третьего порядка на основе заданных точек (узлов интерполяции) и вычисление значения сплайна третьего порядка в серединах частичных отрезков между узлами интерполяции.

2. 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 конечным набором точек:

x_i	1	1.5	2	2.5	3	3.5	4	4.5	5
y_i	3.33	2.30	1.60	1.27	1.18	0.99	1.41	0.80	1.12

3. 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 -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

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

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

На частные многочлены накладываются условия:

1. Сплайн проходит через все узлы

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

2. Условие гладкости на краях

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

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:

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

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

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

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

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

СЛАУ с трехдиагональной матрицей относительно коэффициента c_i:

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

$$c_0 = c_n = 0,$$

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

4. Реализация

Листинг 1. Сплайн-интерполяция

```
package main

import (
	"fmt"
	"math"
)

const SIZE = 8

func f(x float64) float64 {
	return math.Exp(x)
}
```

```
func direct(b, a, c, d []float64, size int) (alpha, beta []float64) {
	alpha = append(alpha, -c[0] / b[0])
	beta = append(beta, d[0] / b[0])
	var y float64
	for i := 1; i < size - 1; i++ {
		y = a[i - 1] * alpha[i - 1] + b[i]
		alpha = append(alpha, -c[i] / y)
		beta = append(beta, (d[i] - a[i - 1] * beta[i - 1]) / y)
	}
	y = a[size - 2] * alpha[size - 2] + b[size - 1]
	beta = append(beta, (d[size - 1] - a[size - 2] * beta[size - 2]) / y)
	return alpha, beta
}

func reverse(alpha, beta []float64, size int) (x []float64) {
	x = make([]float64, size)
	x[size - 1] = beta[size - 1]
	for i := size - 2; i >= 0; i-- {
		x[i] = alpha[i] * x[i + 1] + beta[i]
	}
	return x
}

func main() {
	var l, r, h float64
```

```
    l , r = 1.0, 5.0
    h = (r - l) / SIZE

    xs := []float64{1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5, 5.0}
    ys := []float64{3.33, 2.30, 1.60, 1.27, 1.18, 0.99, 1.41, 0.80, 1.12}

    fmt.Println("table for f(x):")
    for i := 0; i <= SIZE; i++ {
        fmt.Printf("%.1f; %.16f\n", xs[i], ys[i])
    }

    d := []float64{}
    for i := 1; i < SIZE; i++ {
        d = append(d, 3 * (ys[i + 1] - 2 * ys[i] + ys[i - 1]) / (h * h) )
    }

    b := []float64{}
    for i := 1; i < SIZE; i++ {
        b = append(b, 4)
    }

    a := []float64{}
    for i := 1; i < SIZE - 1; i++ {
        a = append(a, 1)
    }

    c := []float64{}
```

```
    for i := 1; i < SIZE - 1; i++ {
        c = append(c, 1)
    }

    alpha, beta := direct(b, a, c, d, SIZE - 1)
    coefC := reverse(alpha, beta, SIZE - 1)
    coefC = append([]float64{0}, coefC...)
    coefC = append(coefC, 0)

    coefA :=  make([]float64, 0, SIZE)
    for i := 1; i <= SIZE; i++ {
        coefA = append(coefA, ys[i - 1])
    }

    coefB := make([]float64, 0, SIZE + 1)
    for i := 1; i <= SIZE; i++ {
        coefB = append(coefB, (ys[i] - ys[i - 1]) / h
- (h / 3) * (coefC[i] + 2 * coefC[i - 1]))
    }

    coefD := make([]float64, 0, SIZE)
    for i := 1; i <= SIZE; i++ {
        coefD = append(coefD, (coefC[i] - coefC[i -
1]) / ( 3 * h))
    }

    for i := 0; i < SIZE; i++ {
        varX := l + float64(i) * h
        varY := ys[i]
```

```
        s := coefA[i] + coefB[i] * (varX - xs[i]) + coefC[i] * math.Pow(varX - xs[i], 2) + coefD[i] * math.Pow(varX - xs[i], 3)
        fmt.Printf("x: %.1f, y: %.16f, y*: %.16f, |y-y*|: %.16f\n", varX, varY, s, math.Abs(varY - s))
    }

    for i := 0; i < SIZE; i++ {
        varX := l + (float64(i + 1) - 0.5) * h
        s := coefA[i] + coefB[i] * (varX - xs[i]) + coefC[i] * math.Pow(varX - xs[i], 2) + coefD[i] * math.Pow(varX - xs[i], 3)
        fmt.Printf("x: %.2f, y*: %.16f\n", varX, s)
    }
}
```

5. Результаты

Для заданных узлов интерполяции (x_i, y_i) построен кубический сплайн с коэффициентами:

```
a: [3.330000 2.300000 1.600000 1.270000 1.180000 0.990000 1.410000 0.800000 ]
b: [-2.196596 -1.786808 -1.036173 -0.248501 -0.489821 0.527787 -0.241327 -0.702478 ]
c: [0.000000 0.819577 0.681694 0.893649 -1.376289 3.411506 -4.949735 4.027434 ]
d: [0.546384 -0.091922 0.141303 -1.513292 3.191863 -5.574161 5.984779 -2.684956 ]
```

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
1	3.3300000000000001
1.25	2.7893882317746685
1.5	2.2999999999999998
1.75	1.9030853046759939

2	1.6000000000000001
2.25	1.3857705495213550
2.5	1.2700000000000000
2.75	1.2400824972385862
3	1.1799999999999999
3.25	1.0213994615243005
3.5	0.9900000000000000
3.75	1.2480696566642120
4	1.4099999999999999
4.25	1.1338219118188511
4.5	0.8000000000000000
4.75	0.8341426960603830
5	1.1200000000000001

6. Вывод

В ходе выполнения лабораторной работы был изучен метод приближения функции с помощью интерполяции кубическим сплайном, был построен сплайн третьего порядка на основе заданных узлов интерполяции и найдены значения функции в серединах частичных отрезков между узлами интерполяции. В результате тестирования сделан вывод о точности данного метода приближения функции: значения функции и значения кубического сплайна совпадают в узлах интерполяции, а в серединах между узлами интерполяции присутствует погрешность результат вследствие увеличения погрешности при реализации арифметических операций на ЦВМ. Однако катастрофического роста погрешности не наблюдается.