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)»

Факультет: Информатика и системы управления

Кафедра: Теоретическая информатика и компьютерные технологии

Лабораторная работа № 3

«Сплайн-интерполяция»

по дисциплине «Численные методы»

Вариант 9

Работу выполнил

студент группы ИУ9-62Б

Егоров Алексей

Москва, 2022

1. Цель работы

Целью данной работы является изучение приближения функция путем сплайн-интерполяции, построение сплайна третьего порядка на основе заданных точек (узлов интерполяции) и вычисление абсолютной погрешности с аппроксимирующей функцией.

2. Постановка задачи

Дано: значения функции в заданных точках (таблица 1), аппроксимирующая функция: $f(x) = \frac{1}{0.22352220596602404 ln(x) + 0.08902416312224844}$

Найти: коэффициенты сплайна на каждом участке разбиения, абсолютную погрешности с аппроксимирующей функцией.

Таблица 1. Значение функции в заданных точках

i	0	1	2	3	4	5	6	7	8
x_i	1	1, 5	2	2, 5	3	3, 5	4	4, 5	5
y_i	9. 44	5. 16	4. 43	3. 31	3.48	3. 2	2. 34	2. 13	2. 18

3. Теоретические сведения

На каждом отрезке разбиения сплайн третьего порядка задается следующим образом:

$$S_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3,\ i = \overline{0, n - 1},\ x \in [x_i, x_{i+1}]$$

Сплайн должен проходить через все узлы интерполяции. Также сплайн и первые две его производные непрерывны в промежуточных узлах. И $S''_0(x_0) = 0,\ S''_{n-1}(x_n) = 0$

Эти условия приводят к трехдиагональной СЛАУ относительно коэффициентов c_i:

$$c_{i-1} + 4c_i + c_{i+1} = \frac{3(y_{i+1} - 2y_i + y_{i-1})}{h^2},\ i = \overline{1, n-1},\ c_0 = c_n = 0$$

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

Систему следует решать методом прогонки.

Остальные коэффициенты выражаются через c_i следующим образом:

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

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

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

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

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

Абсолютная погрешность между аппроксимирующей функцией и сплайном в промежуточных узлах рассчитывается следующим образом:

$$\Delta = \left| f(x_{\Delta}) - S_i(x_{\Delta}) \right|, \ x_{\Delta} = x_0 + (i + 0.5) * h, \ i = \overline{0, n - 1}$$

4. Практическая реализация

Листинг 1. Вычисление коэффициентов сплайна и абсолютной погрешности между сплайном и аппроксимирующей функцией

```
package main

import (
 "fmt"
 "math"
)

const a, b float64 = 0.22352220596602404, 0.08902416312224844
const SIZE = 9

func direct(bs, as, cs, ds []float64, size int) (alpha, beta
[]float64) {
 alpha = append(alpha, -cs[0]/bs[0])
 beta = append(beta, ds[0]/bs[0])
 var div float64
 for i := 1; i < size-1; i++ {
   div = as[i-1]*alpha[i-1] + bs[i]
   alpha = append(alpha, -cs[i]/div)
   beta = append(beta, (ds[i]-as[i-1]*beta[i-1])/div)
 }
 div = as[size-2]*alpha[size-2] + bs[size-1]
 beta = append(beta, (ds[size-1]-as[size-2]*beta[size-2])/div)
 return alpha, beta
}

func reverse(alpha, beta []float64, size int) (xs []float64) {
 xs = make([]float64, size)
```

```
 xs[size-1] = beta[size-1]
 for i := size - 2; i >= 0; i-- {
   xs[i] = alpha[i]*xs[i+1] + beta[i]
 }
 return xs
}

func f(x float64) float64 {
 return 1 / (a*math.Log(x) + b)
}

func main() {
 l, r := 1.0, 5.0
 h := (r - l) / float64(SIZE-1)
 xs := []float64{1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5, 5.0}
 ys := []float64{9.44, 5.16, 4.43, 3.31, 3.48, 3.2, 2.34, 2.13, 2.18}

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

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

 as := []float64{}
 for i := 1; i < SIZE-2; i++ {
   as = append(as, 1)
 }

 cs := []float64{}
 for i := 1; i < SIZE-2; i++ {
   cs = append(cs, 1)
 }

 alpha, beta := direct(bs, as, cs, ds, SIZE-2)
 res := reverse(alpha, beta, SIZE-2)
 res = append([]float64{0}, res...)

 coefA := ys
```

```
  coefB := make([]float64, 0, SIZE-1)
  for i := 0; i < SIZE-2; i++ {
    coefB = append(coefB, (ys[i+1]-ys[i])/h-h/3*(res[i+1]+2*res[i]))
  }
  coefB = append(coefB, (ys[SIZE-1]-ys[SIZE-2])/h-2.0/3*h*res[SIZE-2])

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

  for i := 0; i < SIZE-1; i++ {
    newX := l + (float64(i+1)-0.5)*h
    newY := f(newX)
    val := coefA[i] + coefB[i]*(newX-xs[i]) +
res[i]*math.Pow(newX-xs[i], 2) + coefD[i]*math.Pow(newX-xs[i], 3)
    fmt.Printf(
      "xi: %f, yi: %f, yi*: %f, |yi-yi*|: %f\n",
      newX,
      newY,
      val,
      math.Abs(newY-val),
    )
  }
}
```

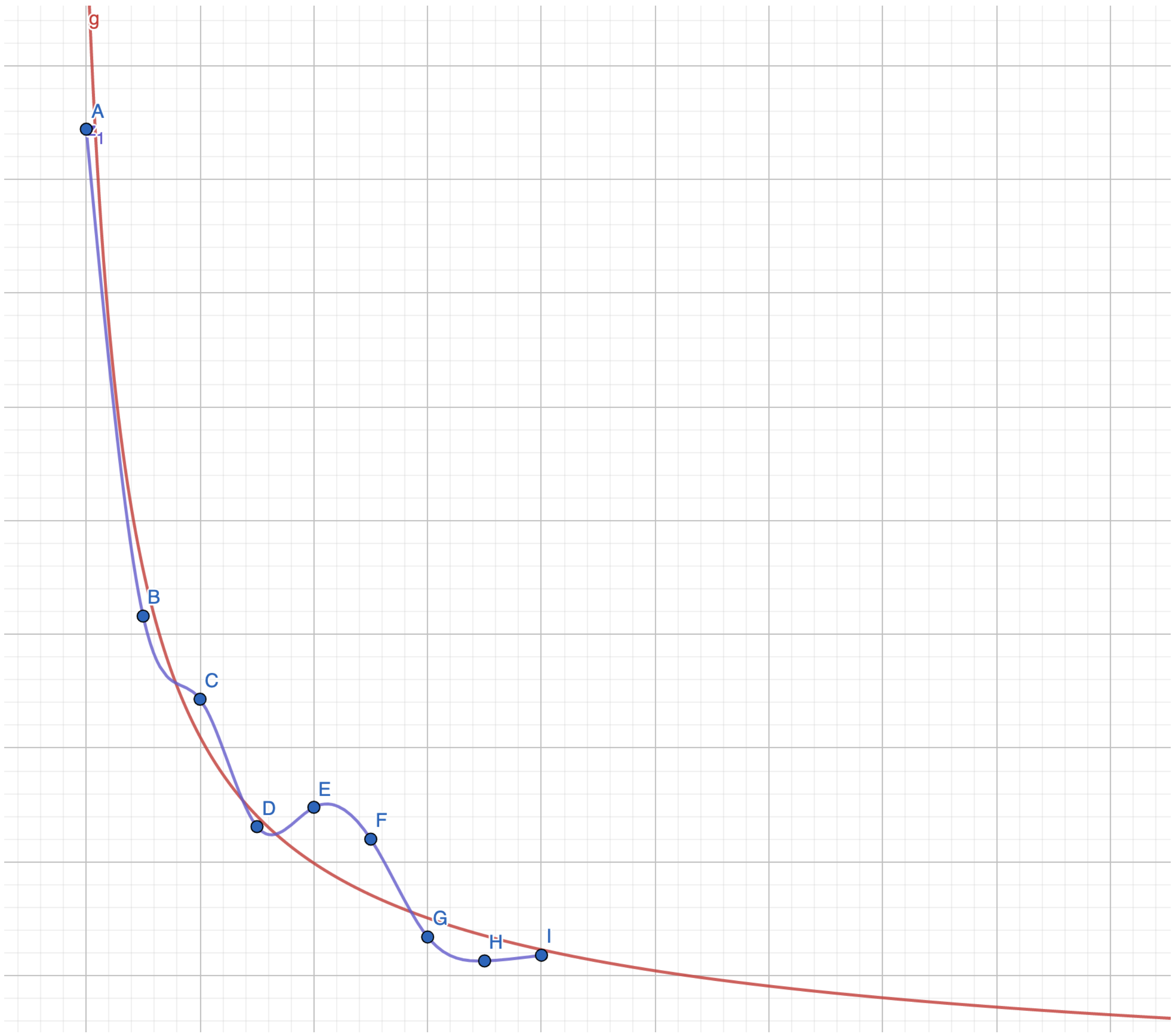

Рисунок 1. Построенные в GeoGebra сплайн и аппроксимирующая функция

```
anyegorov@0000NBA06KRMD6M lab3 % go run main.go
xi: 1.250000, yi: 7.199336, yi*: 6.922997, |yi-yi*|: 0.276339
xi: 1.750000, yi: 4.670481, yi*: 4.594758, |yi-yi*|: 0.075723
xi: 2.250000, yi: 3.699797, yi*: 3.862970, |yi-yi*|: 0.163173
xi: 2.750000, yi: 3.173199, yi*: 3.285861, |yi-yi*|: 0.112663
xi: 3.250000, yi: 2.837043, yi*: 3.468585, |yi-yi*|: 0.631542
xi: 3.750000, yi: 2.601011, yi*: 2.751050, |yi-yi*|: 0.150040
xi: 4.250000, yi: 2.424579, yi*: 2.155964, |yi-yi*|: 0.268616
xi: 4.750000, yi: 2.286738, yi*: 2.148845, |yi-yi*|: 0.137893
anyegorov@0000NBA06KRMD6M lab3 %
```

Рисунок 2. Абсолютная погрешность сплайна и аппроксимирующей функции

5. Вывод

В ходе лабораторной работы был изучен метод приближения функции путем сплайн-интерполяции. Был построен сплайн третьего порядка на основе заданных точек (узлов интерполяции) и вычислена абсолютная погрешность с аппроксимирующей функцией. Сделан вывод о более точном приближении функции методом сплайн-интерполяции по сравнению с аппроксимацией методом наименьших квадратов.