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 № 2

«Аппроксимация методом наименьших квадратов.

Двухпараметрические модели»

по дисциплине «Численные методы»

Вариант 9

Работу выполнил

студент группы ИУ9-62Б

Егоров Алексей

Москва, 2022

Цель работы

Целью данной работы является изучение создания аппроксимирующей функции на основе априорных данных о ней, а также оценка ошибки с помощью среднеквадратичного отклонения.

Задание

1. Построить графики таблично заданной функции и функции $z(x)$.
2. Найти значение x_a, x_g, x_h, y_a, y_g, y_h, $z(x_a)$, $z\left(x_g\right)$, $z(x_h)$, δ_1, …, δ_9, $\delta_k = \min \delta_i$.
3. Составить систему уравнений для определения a и b и решить её.
4. Найти среднеквадратичное отклонение Δ.

Индивидуальный вариант

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

Реализация

1. С использованием сайта GeoGebra на координатной плоскости были изображены заданные точки $(x_i, y_i), i = 0, \dots, n$ и проведена гладкая монотонная кривая, аппроксимирующую эту зависимость (рисунок 1).

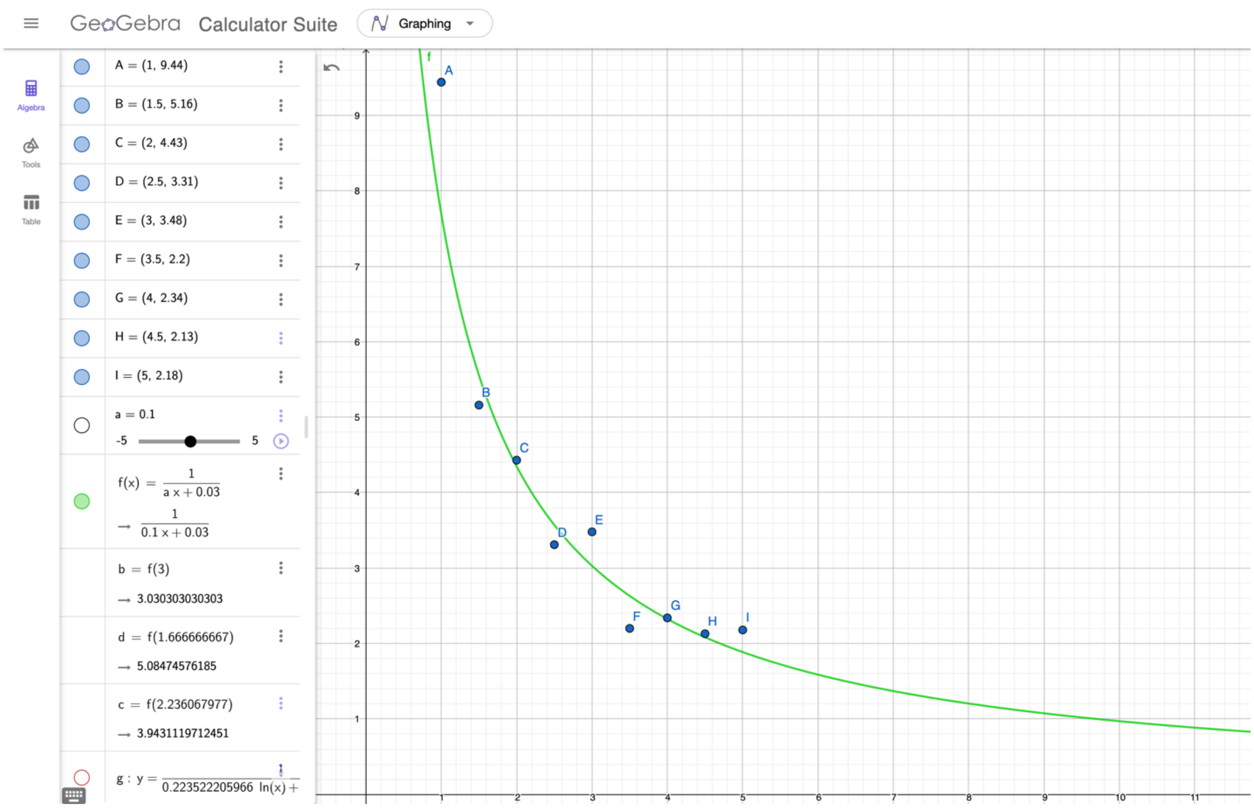
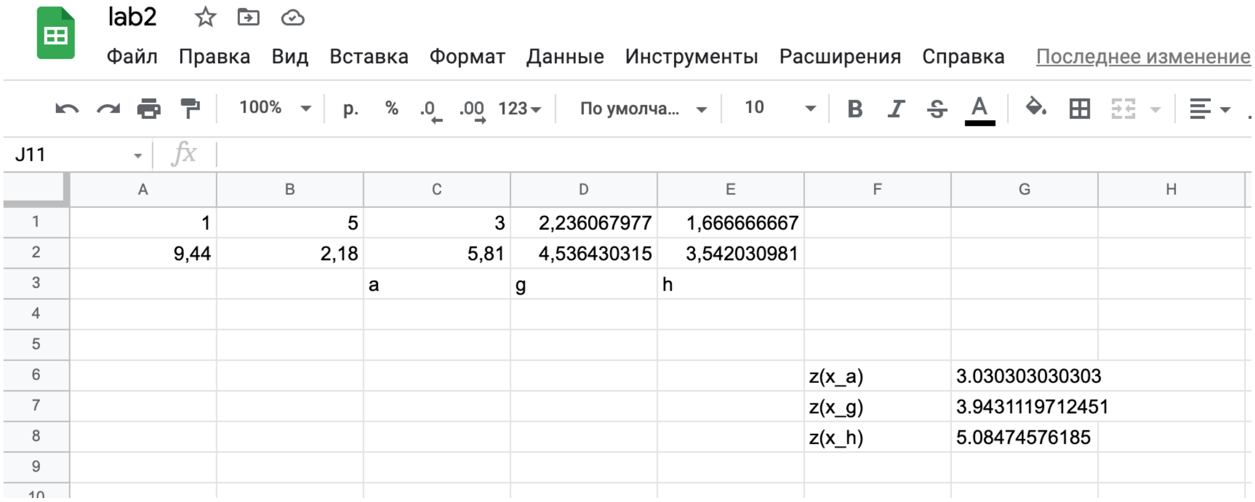


Рисунок 1 – заданные точки и получившийся график в GeoGebra

2. С использованием Google Таблиц (рисунок 2) и GeoGebra (рисунок 1), были вычислены значения величин $x_a = \frac{x_0+x_n}{2}$, $x_g = \sqrt{x_0 x_n}$, $x_h = \frac{2}{\frac{1}{x_0}+\frac{1}{x_n}}$, $y_a = \frac{y_0+y_n}{2}$, $y_g = \sqrt{y_0 y_n}$, $y_h = \frac{2}{\frac{1}{y_0}+\frac{1}{y_n}}$, $z(x_a)$, $z(x_g)$, $z(x_h)$. Так же были вычислены значения следующих величин с помощью языка программирования Python:

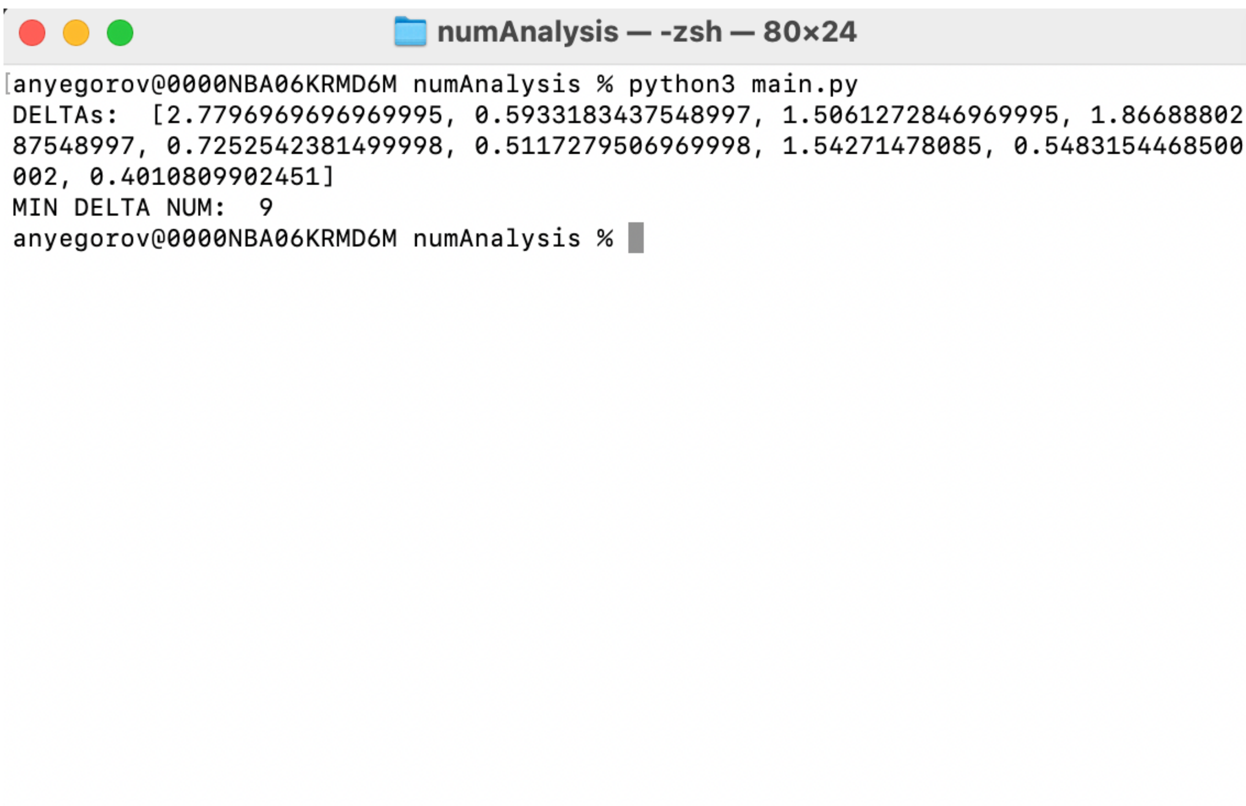
$\delta_1 = \lvert z(x_a) - y_a \rvert$,	$\delta_2 = \lvert z(x_g) - y_g \rvert$,	$\delta_3 = \lvert z(x_a) - y_g \rvert$,
$\delta_4 = \lvert z(x_g) - y_a \rvert$,	$\delta_5 = \lvert z(x_h) - y_a \rvert$,	$\delta_6 = \lvert z(x_a) - y_h \rvert$,
$\delta_7 = \lvert z(x_h) - y_h \rvert$,	$\delta_8 = \lvert z(x_h) - y_g \rvert$,	$\delta_9 = \lvert z(x_g) - y_h \rvert$

Минимальной оказалась величина δ_9 (рисунок 3).



	A	B	C	D	E	F	G	H
1	1	5	3	2,236067977	1,666666667			
2	9,44	2,18	5,81	4,536430315	3,542030981			
3			a	g	h			
4								
5								
6						z(x_a)	3.030303030303	
7						z(x_g)	3.9431119712451	
8						z(x_h)	5.08474576185	
9								

Рисунок 2. Расчеты в Google Таблицах



```
anyegorov@0000NBA06KRMD6M numAnalysis % python3 main.py
DELTAs:  [2.7796969696969995, 0.5933183437548997, 1.5061272846969995, 1.86688802
87548997, 0.7252542381499998, 0.5117279506969998, 1.54271478085, 0.5483154468500
002, 0.4010809902451]
MIN DELTA NUM:  9
anyegorov@0000NBA06KRMD6M numAnalysis %
```

Рисунок 3. Расчет δ_i

3. Для определения коэффициентов a и b перейдем к обратным величинам:

$$\frac{1}{z_9(x)} = \frac{1}{alnx + b} = alnx + b$$

Минимизируется величина:

$$F(a, b) = \sum_{i=0}^{n} \left(alnx + b - \frac{1}{y_i} \right)^2$$

$$\frac{\partial F}{\partial a} = 2 \sum_{i=0}^{n} \left(a\,(\ln x_i)^2 + b \ln x_i - \frac{\ln x_i}{y_i} \right)$$

$$\frac{\partial F}{\partial b} = 2 \sum_{i=0}^{n} \left(a \ln x_i + b - \frac{1}{y_i} \right)$$

$$\begin{cases} \dfrac{\partial F}{\partial a} = 0 \\ \dfrac{\partial F}{\partial b} = 0 \end{cases} \Rightarrow \begin{cases} \displaystyle\sum_{i=0}^{n} \left(a\,(\ln x_i)^2 + b \ln x_i - \frac{\ln x_i}{y_i} \right) = 0 \\ \displaystyle\sum_{i=0}^{n} \left(a \ln x_i + b - \frac{1}{y_i} \right) = 0 \end{cases}$$

Система была решена с помощью Python, полученные значения a и b представлены на Рисунке 3. Полученная функция $z_9(x)$ построена в GeoGebra (Рисунок 4). Также с помощью Python были вычислены значения точек в $z_9(x_i)$ $i = 0, \dots, n$ и найдено среднеквадратичное отклонение Δ (Рисунок 3).

```
anyegorov@0000NBA06KRMD6M numAnalysis % python3 main.py
a =  0.22352220596602404
b =  0.08902416312224844
sd =  4.0588455287777725
anyegorov@0000NBA06KRMD6M numAnalysis %
```

Рисунок 3 – расчеты a и b

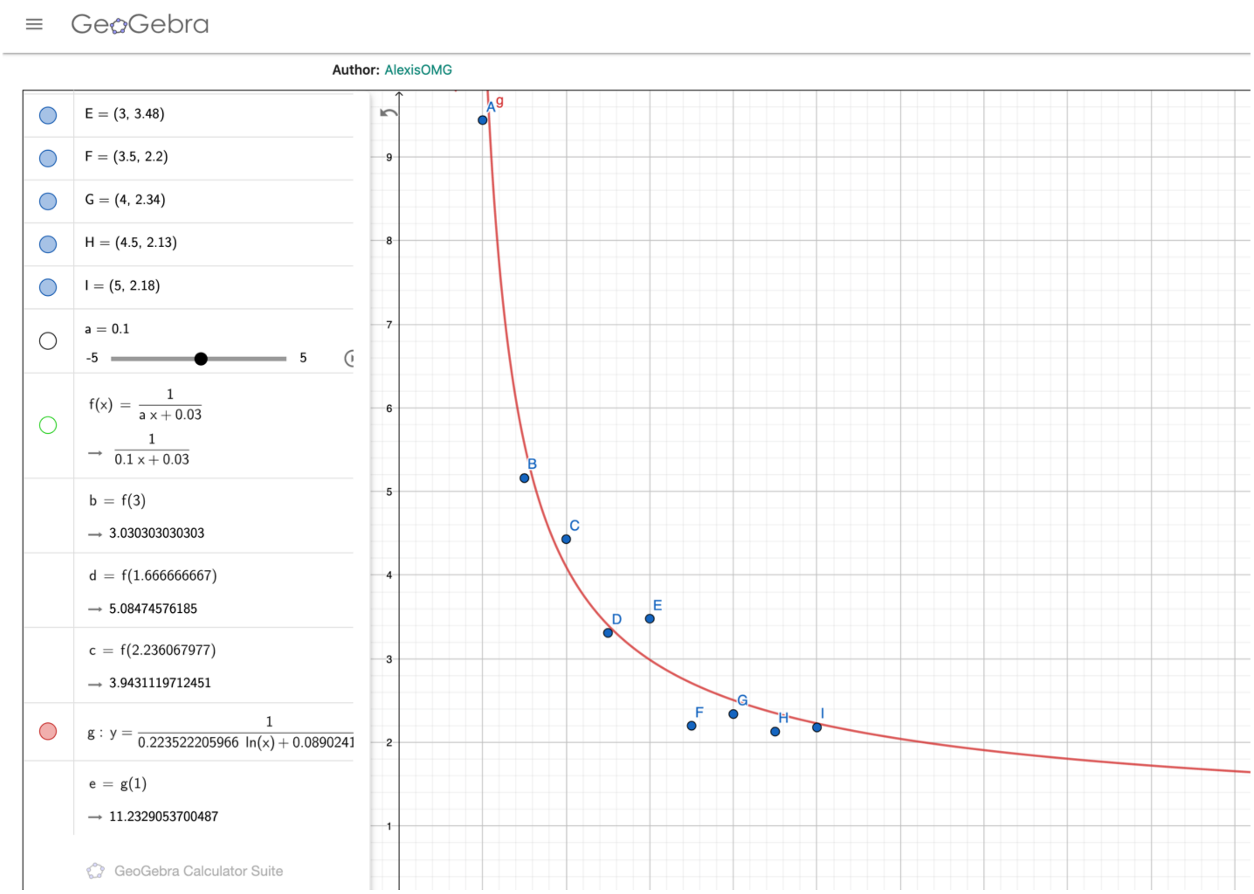


Рисунок 4 – график аппроксимированной функции в GeoGebra

Листинг программы на Python

```
import math


z_x_a = 3.030303030303
z_x_g = 3.9431119712451
z_x_h = 5.08474576185

y_a = 5.81
y_g = 4.536430315
y_h = 3.542030981

deltas = [
     abs(z_x_a - y_a),
     abs(z_x_g - y_g),
     abs(z_x_a - y_g),
     abs(z_x_g - y_a),
     abs(z_x_h - y_a),
     abs(z_x_a - y_h),
     abs(z_x_h - y_h),
     abs(z_x_h - y_g),
     abs(z_x_g - y_h),
```

```
]

print('DELTAs: ', deltas)

min_delta = min(deltas)

for i in range(len(deltas)):
    if deltas[i] == min_delta:
        print('MIN DELTA NUM: ', i+1)

xs = [1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5, 5.0]
ys = [9.44, 5.16, 4.43, 3.31, 3.48, 3.2, 2.34, 2.13, 2.18]

n = 9

sum1 = sum([math.log(xi)**2 for xi in xs])
sum2 = sum([math.log(xi) for xi in xs])
sum3 = sum([1/yi for yi in ys])
sum4 = sum([math.log(xi)/yi for (xi, yi) in zip(xs, ys)])

a = (sum4-sum3*sum2/n)/((n*sum1-sum2**2)/n)
b = (sum3-a*sum2)/n

print('a = ', a)
print('b = ', b)

func = lambda x: 1/(a*math.log(x)+b)

z_ys = [func(xi) for xi in xs]

sd = sum([(z_yi - yi)**2 for (z_yi, yi) in zip(z_ys, ys)])

print('sd = ', sd)
```

Вывод

В ходе лабораторной работы был изучен способ создания аппроксимирующей функции на основе априорных данных о ней. Предполагаемая и получившаяся функция различаются. Среднеквадратичное отклонение позволяет оценить размер получившийся ошибки.