

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«УФИМСКИЙ УНИВЕРСИТЕТ НАУКИ И ТЕХНОЛОГИЙ»

Институт математики, информатики и робототехники
Кафедра информационных технологий и компьютерной математики

ОТЧЕТ ОБ УЧЕБНОЙ ПРАКТИКЕ

ТИП ПРАКТИКИ

Технологическая (проектно-технологическая) практика

ОБУЧАЮЩЕГОСЯ

3 курса группы ПИ-ЗИВТ211Б

Павловский Даниил Анатольевич

Уровень высшего образования:

бакалавриат

Специальность

(направление подготовки)

Направленность (профиль)

программы

09.03.03 “Прикладная информатика”

Информационные и вычислительные технологии

Срок проведения практики

1 семестр 2023/2024 учебного года

УФА – 2023

1. МЕТОДИЧЕСКИЕ УКАЗАНИЯ

1. База практики – место прохождения практики студентом (профильная организация или УУНиТ).
2. Обучающийся – физическое лицо, осваивающее образовательную программу по направлению подготовки бакалавриата, магистратуры и специальности.
3. Вид практики – учебная, производственная или преддипломная.
4. Каждый обучающийся, находящийся на практике, обязан вести отчет по практике.
5. Отчет по практике служит основным и необходимым материалом для составления обучающимся отчета о своей работе на базе практики.
6. Заполнение отчета по практике производится регулярно, аккуратно и является средством самоконтроля. Отчет можно заполнять рукописным и (или) машинописным способами.
7. Иллюстративный материал (чертежи, схемы, тексты и т.п.), а также выписки из инструкций, правил и других материалов могут быть выполнены на отдельных листах и приложены к отчету.
8. Записи в отчете о практике должны производиться в соответствии с программой по конкретному виду практики.
9. После окончания практики обучающийся должен подписать отчет у руководителя практики, руководителя от базы практики и сдать свой отчет по практике вместе с приложениями (при наличии) на кафедру.
10. При отсутствии сведений в соответствующих строках ставится прочерк.

2. ОБЩИЕ ПОЛОЖЕНИЯ


Фамилия, инициалы, должность руководителя практики от факультета (института)	Корнилова А.А., к.х.н., доцент
Фамилия, инициалы, должность руководителя практики от кафедры	Галеева Г.Я., доцент
Полное наименование базы практики	Институт математики, информатики и робототехники
Наименование структурного подразделения базы практики	Кафедра информационных технологий и компьютерной математики
Адрес базы практики (индекс, субъект РФ, район, населенный пункт, улица, дом, офис)	450074, г. Уфа, р-н Кировский, ул. Заки Валиди, д. 32, физико-математический корпус, ауд. 511а.
Фамилия, инициалы, должность руководителя практики от профильной организации	Галеева Г.Я., доцент
Телефон руководителя практики от базы практики	89177618833

3. РАБОЧИЙ ГРАФИК (ПЛАН) ПРОВЕДЕНИЯ ПРАКТИКИ

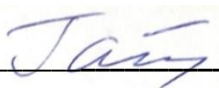
Срок проведения практики в соответствии с календарным учебным графиком: 1 семестр 2023/2024_ учебного года объемом 3 зачетных единицы.

№	Разделы (этапы) практики	Виды и содержание работ, включая самостоятельную работу обучающегося в соответствии с программой практики
1.	Подготовительный этап.	<i>Прохождение инструктажа по технике безопасности и охране труда Получение индивидуального задания по практике.</i>
2.	Основной этап.	<i>Разработка и реализация на языке Python проекта решения типовой математической задачи. Разработка для конкретной предметной области приложения средствами языка Python. Использование текстового процессора Microsoft Word для создания отчета о прохождении учебной практики.</i>
3.	Заключительный этап.	<i>Обработка результатов выполненного задания Составление отчета по практике</i>

Руководитель практики от кафедры

 / Галеева Г.Я.

Руководитель практики от профильной организации


 / Галеева Г.Я.

4. ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ


Содержание и планируемые результаты практики:

1. Овладение методикой модульного программирования, умение работать в коллективе.
2. Знакомство с практическими приложениями python для построения графиков и применением численных методов обработки табличных данных.
3. Подготовка и оформление отчета по учебной практике.


Руководитель практики от профильной
организации

 / Галеева Г.Я.
подпись И.О. Фамилия

Руководитель практики от кафедры

 / Галеева Г.Я.
подпись И.О. Фамилия

ОЗНАКОМЛЕН:
Обучающийся


 / Павловский Д. А.
подпись И.О. Фамилия

5. ИНСТРУКТАЖ ПО ОХРАНЕ ТРУДА

Инструкция о мерах пожарной безопасности в Уфимском университете науки и технологий, утвержденная приказом УУНиТ.


Правила внутреннего трудового распорядка обучающихся в Уфимском университете науки и технологий, утвержденные приказом УУНиТ.

Перед началом практики инструктаж по ознакомлению с требованиями охраны труда, техники безопасности, пожарной безопасности, а также правилами внутреннего трудового распорядка прошел:

обучающийся  / Д. А. Павловский
подпись

И.О.Фамилия

Перед началом практики инструктаж обучающегося по ознакомлению с требованиями охраны труда, техники безопасности, пожарной безопасности, а также правилами внутреннего трудового распорядка провел:


 / Г.Я. Галеева
доцент должность подпись И.О.

Фамилия


6. ДНЕВНИК РАБОТЫ СТУДЕНТА

Дата	Информация о проделанной работе, использованные источники и литература (при наличии)
02.10.2023	Прохождение инструктажа по технике безопасности по охране труда
03.10.2023	Получение индивидуального задания по практике
04.10.2023- 28.12.2023	Выполнение индивидуального задания по практике, составление отчета по практике

Руководитель практики от кафедры

 / Г.Я. Галеева
подпись И.О. Фамилия

Руководитель практики от профильной организации

 / Г.Я. Галеева
подпись И.О. Фамилия

7. ОТЧЕТ ОБУЧАЮЩЕГОСЯ О ПРАКТИКЕ

1 семестр 2023/2024 учебного года

Я, Павловский Даниил Анатольевич прошёл учебную практику во 1 семестре 2023/2024 учебного года

В соответствии с программой практики и индивидуальным заданием, я выполнял следующую работу: разработка и реализация на языке Python проекта решения типовой математической задачи (тема интерполяция и аппроксимация функций). Разработка для конкретной предметной области приложения средствами языка Python. Использование текстового процессора Microsoft Word для создания отчета о прохождении учебной практики.

В результате прохождения практики, поставленные задачи были решены в полном объеме, профессиональные компетенции (профессиональные умения, навыки и опыт профессиональной деятельности) приобретены.

Обучающийся



/ Павловский Д. А.

подпись

И.О. Фамилия

8. ЗАКЛЮЧЕНИЕ РУКОВОДИТЕЛЯ ПО ПРАКТИЧЕСКОЙ ПОДГОТОВКЕ О ПРАКТИКЕ

Обучающийся Павловский Даниил Анатольевич прошёл учебную практику в 1 семестре 2023/2024 учебного года


Перед обучающимся во время прохождения практики были поставлены следующие профессиональные задачи: получить профессиональные умения и навыки научно-исследовательской деятельности.

Краткая характеристика проделанной работы и полученных результатов: разработка и реализация проекта, самостоятельное знакомство с практическими приложениями. Поставленные цели достигнуты.

Во время прохождения практики обучающийся проявил себя как: Организованный, дисциплинированный студент с хорошим уровнем технической подготовки, недостатков не выявлено.

Рекомендации (пожелания) по организации практики: Продолжить развивать навыки и способности.

Руководитель практики от кафедры

 / Г.Я. Галеева
подпись И.О. Фамилия

9. РЕЗУЛЬТАТ ЗАЩИТЫ ОТЧЕТА

В результате прохождения практики поставленные задачи были решены в полном объеме, профессиональные компетенции (профессиональные умения, навыки и опыт профессиональной деятельности) приобретены.

Результат прохождения практики обучающимся оценивается на: «отлично»

Руководитель практики от кафедры



/ Г.Я. Галеева

подпись И.О. Фамилия

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«УФИМСКИЙ УНИВЕРСИТЕТ НАУКИ И ТЕХНОЛОГИЙ»

Институт математики, информатики и робототехники
Кафедра информационных технологий и компьютерной математики

ПРИЛОЖЕНИЕ К ОТЧЕТУ ОБ УЧЕБНОЙ ПРАКТИКЕ

Технологическая (проектно-технологическая) практика

СТУДЕНТА

3 курса группы ПИ-ЗИВТ211Б

Павловский Даниил Анатольевич

На тему:” **Интерполирование и аппроксимация функций**”

Уровень высшего образования:

бакалавриат

Специальность

(направление подготовки)

09.03.03 “Прикладная информатика”

Направленность (профиль)

Программы

Информационные и вычислительные технологии

Срок проведения практики

1 семестр 2023/2024 учебного года

Содержание

Введение	13
Теоритическая часть.	14
Интерполяция Квадратичный сплайн дефекта 2	14
Аппроксимация.	17
Аппроксимация линейной функцией	18
Аппроксимация экспоненциальной функции.	19
Реализация программы	21
Код программы	21
Результат работы программы	23
Заключение	24
Источники	25

Введение

Цель:

Создать две программы:

- 1) Интерполяция функции по методу Квадратичный сплайн дефекта 2;
- 2) Аппроксимация линейной и экспоненциальной функций.

Содержимое задания:

Задание 1. *Выберите 10 значений функции, начиная с узла, равного номеру вашего варианта.*

Задание 2. *Для выбранной таблицы выполните указанную интерполяцию и аппроксимацию.*

Задание 3. *Построить график полученной интерполяции, аппроксимации.*

Задание 4. *Составить отчет с выводом указанных методов. Сделать оценку погрешности.*

Задание 5. *Подсчитать среднеквадратичное отклонение σ , сделать сглаживание (для аппроксимации).*

Интерполяция: Квадратичный сплайн дефекта 2;

Аппроксимация: $y = a \ln(x) + b$, $y = ax + b$

Массив X: [15.0, 15.1, 15.2, 15.3, 15.4, 15.5, 15.6, 15.7, 15.8, 15.9]

Массив Y: [-0.00091, 0.624825, 1.203832, 1.677044, 1.994648, 2.12132, 2.040105, 1.754519, 1.288629, 0.685062]

Этапы выполнения задачи:

- Разбор методов решения.
- Реализация программы средствами языка Python
- Визуализация решения средствами библиотеки matplotlib

Теоретическая часть.

Интерполяция Квадратичный сплайн дефекта 2

Интерполяция — это метод нахождения новых точек данных в пределах диапазона дискретного набора известных точек данных. Другими словами, интерполяция — это метод оценки значения математической функции для любого промежуточного значения независимой переменной.

Пусть известны значения некоторой функции $y = f(x)$ в некоторых точках $x_i, i = \overline{0, n}$. и $f(x_i) = y_i, i = \overline{0, n}$.

Определение. Сплайном $S_m(x)$ называется определенная на $[a, b]$ функция, принадлежащая классу $C^l[a, b]$ (1 раз непрерывно дифференцируемых функций), такая, что на каждом промежутке $[x_{k-1}, x_k]$ ($k = \overline{1, n}$) — это многочлен m -й степени. Разность $d = m - l$ называется дефектом сплайна.

Интерполяционный сплайн 2 степени дефекта 2. (кусочно-квадратичная функция) при $n=2m$:

$$S_2(x) = \begin{cases} a_1x^2 + b_1x + c_1 & x \in [x_0, x_2] \\ a_2x^2 + b_2x + c_2 & x \in [x_2, x_4] \\ \dots \dots \dots \dots \dots \dots \\ a_mx^2 + b_mx + c_m & x \in [x_{2m-2}, x_{2m}] \end{cases}$$

Для определения $3m$ неизвестных $a_k, b_k, c_k, (k = \overline{1, m})$ составляется m трехмерных систем типа

$$\begin{cases} a_1 x_0^2 + b_1 x_0 + c_1 = y_0 \\ a_1 x_1^2 + b_1 x_1 + c_1 = y_1 \\ a_1 x_2^2 + b_1 x_2 + c_1 = y_2 \end{cases}$$

и т.д.

Аппроксимация.

Аппроксимацией (приближением) функции $f(x)$ называется нахождение такой функции (аппроксимирующей функции) $g(x)$, которая была бы близка заданной. Критерии близости функций могут быть различные.

В случае если приближение строится на дискретном наборе точек, аппроксимацию называют точечной или дискретной.

В случае если аппроксимация проводится на непрерывном множестве точек (отрезке), аппроксимация называется непрерывной или интегральной. Примером такой аппроксимации может служить разложение функции в ряд Тейлора, то есть замена некоторой функции степенным многочленом.

Наиболее часто встречающимся видом точечной аппроксимации является интерполяция – нахождение промежуточных значений величины по имеющемуся дискретному набору известных значений.

Пусть задан дискретный набор точек, называемых узлами интерполяции, а также значения функции в этих точках. Требуется построить функцию $g(x)$, проходящую наиболее близко ко всем заданным узлам. Таким образом, критерием близости функции является $g(x_i) = x_i$.

В качестве функции $g(x)$ обычно выбирается полином, который называют интерполяционным полиномом.

В случае если полином один для всей области интерполяции, говорят, что интерполяция глобальная.

В случае если между различными узлами полиномы различны, говорят о кусочной или локальной интерполяции.

Найдя интерполяционный полином, мы можем вычислить значения функции между узлами, а также определить значение функции даже за пределами заданного интервала (провести экстраполяцию).

Критерий успешности аппроксимации – среднеквадратичное отклонение $\ll 1$

Формула для вычисления среднеквадратичного отклонения:

$$\sigma = \sqrt{\frac{1}{n} \sum_{k=1}^n (y_k - f(x_k))^2}$$

Метод наименьших квадратов

Метод наименьших квадратов применяется для решения следующих задач:

1. Необходимо определить величины x_1, x_2, \dots, x_N , которые нельзя определить непосредственно, но известно, что они линейно зависимы, а коэффициенты этой зависимости можно получить в результате измерений. Таким образом, мы имеем переопределенную систему линейных алгебраических уравнений. Решение этой системы может быть получено решением задачи минимизации. Выполняя дифференцирование минимизируемой функции, приходим к линейной системе, которая будет иметь N уравнений и N неизвестных.

2. Требуется дать приближенное аналитическое описание по таблично заданным данным. Из каких-либо соображений подбирается аппроксимирующая функция, а параметры этой функции подбираются так, чтобы сумма квадратов отклонений вычисляемых значений аппроксимирующей функции от заданных была минимальной.

Для выбранной функции $g(x)$ ищутся оптимальные значения констант.

Метод наименьших квадратов (МНК) заключается в том, что рассматривается функция

$$\Phi(a_0, a_1, \dots, a_m) = \sum_{i=1}^n (F(x_i, a_0, a_1, \dots, a_m) - y_i)^2$$

Для минимального отклонения находим частные производные функции по коэффициентам a и приравниваем их к нулю.

$$\frac{\partial \Phi}{\partial a_0} = 0, \quad \frac{\partial \Phi}{\partial a_1} = 0 \quad \dots \quad \frac{\partial \Phi}{\partial a_m} = 0$$

Это нормальная система МНК

Частный случай функции

$$g(x, a_0, a_1, \dots, a_m) = a_0 + a_1 x + \dots + a_m x^m$$

$$\Phi(a_0, a_1, \dots, a_m) = \sum_{i=1}^n (a_0 + a_1 x_i + \dots + a_m x_i^m - y_i)^2$$

Нормальная система

$$\begin{aligned} \frac{1}{2} \frac{\partial \Phi}{\partial a_0} &= \sum (a_0 + a_1 x_i + \dots + a_m x_i^m - y_i) = 0 \quad \frac{1}{2} \frac{\partial \Phi}{\partial a_1} \\ &= \sum (a_0 + a_1 x_i + \dots + a_m x_i^m - y_i) x_i = 0 \quad \dots \quad \frac{1}{2} \frac{\partial \Phi}{\partial a_m} \\ &= \sum (a_0 + a_1 x_i + \dots + a_m x_i^m - y_i) x_i^m = 0 \end{aligned}$$

Из этого следует

$$\begin{aligned} \{ n a_0 + a_1 \sum x_i + a_2 \sum x_i^2 + \dots + a_m \sum x_i^m &= \sum y_i \quad a_0 \sum x_i + a_1 \sum x_i^2 + \dots + a_m \sum x_i^{m+1} \\ &= \sum y_i x_i \quad \dots \quad a_0 \sum x_i^m + a_1 \sum x_i^{m+1} + \dots + a_m \sum x_i^{2m} = \sum x_i^m y_i \end{aligned}$$

Например, для функции $y = ax + b$

Решаем полученную систему уравнений

$$\begin{cases} a \cdot \sum_{i=1}^n x_i^2 + b \cdot \sum_{i=1}^n x_i = \sum_{i=1}^n x_i \cdot y_i \\ a \cdot \sum_{i=1}^n x_i + \sum_{i=1}^n b = \sum_{i=1}^n y_i \end{cases} \Rightarrow \begin{cases} a \cdot \sum_{i=1}^n x_i^2 + b \cdot \sum_{i=1}^n x_i = \sum_{i=1}^n x_i \cdot y_i \\ a \cdot \sum_{i=1}^n x_i + n \cdot b = \sum_{i=1}^n y_i \end{cases}$$

Определяем значения коэффициентов

$$\begin{cases} a = \frac{n \cdot \sum_{i=1}^n x_i \cdot y_i - \sum_{i=1}^n x_i \cdot \sum_{i=1}^n y_i}{n \cdot \sum_{i=1}^n x_i^2 - (\sum_{i=1}^n x_i)^2} \\ b = \frac{\sum_{i=1}^n y_i - a \cdot \sum_{i=1}^n x_i}{n} \end{cases}$$

Для вычисления коэффициентов необходимо найти следующие составляющие:

$$\begin{aligned} \text{sum}x &= \sum_{i=1}^n x_i & \text{sum}y &= \sum_{i=1}^n y_i \\ \text{sum}x^2 &= \sum_{i=1}^n x_i^2 & \text{sum}xy &= \sum_{i=1}^n x_i \cdot y_i \end{aligned}$$

Тогда значения коэффициентов будут определены как

$$\begin{cases} a = \frac{n \cdot \text{sum}xy - \text{sum}x \cdot \text{sum}y}{n \cdot \text{sum}x^2 - \text{sum}x^2} \\ b = \frac{\text{sum}y - a \cdot \text{sum}x}{n} \end{cases}$$

Метод линеаризации данных по методу наименьших квадратов.

Техника линеаризации данных применяется для подгонки кривых, позволяющих при преобразовании переменных получить линейную зависимость вида $y = Ax + B$. В таблице 1 приведены основные приемы линеаризации.

Таблица замены переменной для метода линеаризации данных

№ п/п	Функция	Линеаризованная форма	Замена переменных и констант
1	$y = \frac{A}{x} + B$	$y = A \frac{1}{x} + B$	$X = \frac{1}{x}, \quad Y = y$
2	$y = \frac{A}{x+B}$	$y = -\frac{1}{B}(xy) + \frac{A}{B}$	$X = xy, \quad Y = y$
3	$y = \frac{x}{Ax+B}$	$\frac{1}{y} = A \frac{1}{x} + B$	$X = \frac{1}{x}, \quad Y = \frac{1}{y}$
4	$y = A \ln(x) + B$	$y = A \ln(x) + B$	$X = \ln(x), \quad Y = y$
5	$y = C e^{Ax}$	$\ln \ln(y) = Ax + \ln(C)$	$X = x, Y = \ln \ln(y), B = \ln(C)$
6	$y = C x^A$	$\ln \ln(y) = A \ln(x) + \ln(C)$	$X = \ln(x), Y = \ln \ln(y), B = \ln(C)$

Пусть заданы N точек с различными абсциссами $\{x_k\}$. Величина среднеквадратичной ошибки будет минимальной, когда каждая частная производная $(E_3(f))^2 = (\frac{1}{N} \sum_{k=1}^N |f(x_k) - y_k|^2)$ по неизвестным (в данном случае неизвестные A и B) будет обращаться в нуль, т.е. A и B являются решением нормальной системы уравнений вида:

$$\left\{ \left(\sum_{k=1}^N x_k \right) A + NB = \left(\sum_{k=1}^N y_k \right) \left(\sum_{k=1}^N x_k^2 \right) A + \left(\sum_{k=1}^N x_k \right) B = \left(\sum_{k=1}^N y_k x_k \right) \right.$$

Решая систему нормальных уравнений, находим искомые коэффициенты A и B .

Реализация программы

Код программы 1

```
import numpy as np
import matplotlib.pyplot as plt

# Создаем массив Data, содержащий пары значений [x, y], x и y представляют собой координаты
# точек на плоскости.
Data = [[15.0, -0.00091],
        [15.1, 0.624825],
        [15.2, 1.203832],
        [15.3, 1.677044],
        [15.4, 1.994648],
        [15.5, 2.12132],
        [15.6, 2.040105],
        [15.7, 1.754519],
        [15.8, 1.288629],
        [15.9, 0.685062],
        [16.0, 0.00091]] # X и Y

for j in range(0, 9, 2): # цикл перебирает значения j от 0 до 9 с шагом 2. Таким образом, j будет
# принимать значения 0, 2, 4, 6 и 8.
    print(j)
    # строит матрицу с коэфф в виде иксов на отрезке [x0,x2],  $ax[j]^2 + bx[j] + c$ . У с коэфф всегда
    # равен 1.
    Matrix = np.array([[Data[j][0] 2, Data[j][0], 1],
                      [Data[j + 1][0] 2, Data[j + 1][0], 1],
                      [Data[j + 2][0] 2, Data[j + 2][0], 1]])
    print(Matrix)

    # Создаем вектор Y, в котором только 3 элемента. например, для j = 0: y0,y1,y2
    Y = np.array([Data[j][1], Data[j+1][1], Data[j + 2][1]])

    # Решаем СЛАУ из 3 уравнений. И получаем коэффициенты a b c для данной СЛАУ
    coeff = np.linalg.solve(Matrix, Y)
    print(coeff)
    a = coeff[0]
    b = coeff[1]
    c = coeff[2]

    # Когда будет рассматриваться отрезок [x9,x11], будем рисовать только [x9, x10]
    if j == 8:
        x = np.linspace(Data[j][0], Data[j + 1][0], 10) # для более гладкой линии
    else:
        x = np.linspace(Data[j][0], Data[j + 2][0], 10)

    # подставляем значения нашей линии в функцию с коэфф на отрезке xj, xj+2.
    y = a * x ** 2 + b * x + c

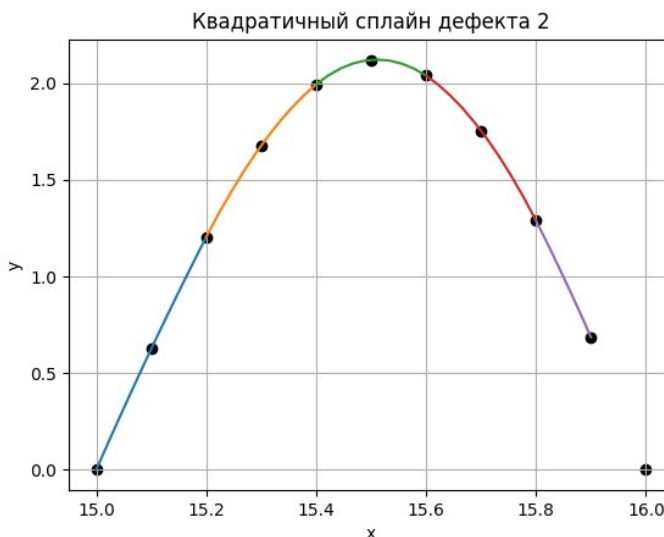
    # Строим линию на отрезке данным отрезке
    plt.plot(x, y, '-')

# Построение графика
```

```
plt.scatter([point[0] for point in Data], [point[1] for point in Data], c='k')
plt.title("Квадратичный сплайн дефекта 2")
plt.xlabel("x")
plt.ylabel("y")
plt.grid()
plt.show()
```

Результат работы программы 1

```
0
[[225.  15.  1. ]
 [228.01 15.1  1. ]
 [231.04 15.2  1. ]]
[ -2.3364  76.58299 -623.05576]
2
[[231.04 15.2  1. ]
 [234.09 15.3  1. ]
 [237.16 15.4  1. ]]
[ -7.7804  242.03432 -1880.13421601]
4
[[237.16 15.4  1. ]
 [240.25 15.5  1. ]
 [243.36 15.6  1. ]]
[ -10.39435  322.452135 -2498.644185]
6
[[243.36 15.6  1. ]
 [246.49 15.7  1. ]
 [249.64 15.8  1. ]]
[ -9.0152  279.3199 -2161.411263]
8
[[249.64 15.8  1. ]
 [252.81 15.9  1. ]
 [256.  16.  1. ]]
[ -4.02925  121.691555 -915.57597 ]
```



Код программы 2

```
import numpy as np
import matplotlib.pyplot as plt
import math

# Создаются списки X и Y, содержащие координаты точек
X = [15.0, 15.1, 15.2, 15.3, 15.4, 15.5, 15.6, 15.7, 15.8, 15.9]
Y = [-0.00091, 0.624825, 1.203832, 1.677044, 1.994648, 2.12132, 2.040105, 1.754519, 1.288629, 0.685062]

# Задается значение переменной n
n = 10

# Выполняется сдвиг массива Y, если Y[i] < 0:
min = 100
Y_C = []
for i in range(0, n):
    if Y[i] < 0:
        if Y[i] < min:
            min = Y[i]
min = abs(math.floor(min))
print('Сдвиг по Y на: ', min)

for i in range(0, n):
    Y[i] += min
```

```

# Запрашивается выбор типа аппроксимации:
print ('Сдвинутый массив:\n', Y)
print("Выберите тип аппроксимации:\n1)Логарифмическая\n2)Линейная")

# Логарифмическая аппроксимация
change = int(input())
if change == 1:
    # Ищем решение системы
    sum_ylnx = 0
    sum_lnx = 0
    sum_y = 0
    sum_ln2x = 0
    sum_powx = 0

    # инициализируется нулём и затем увеличивается путем сложения каждого элемента Y[i]
    # умноженного на np.log(X[i]).
    for i in range(0, n):
        sum_ylnx += Y[i] * np.log(X[i])

    # инициализируется нулём и затем увеличивается путем сложения значений np.log(X[i]).
    for i in range(0, n):
        sum_lnx += np.log(X[i])

    # инициализируется нулём и затем увеличивается путем сложения каждого элемента Y[i].
    for i in range(0, n):
        sum_y += Y[i]

    # инициализируется нулём и затем увеличивается путем сложения квадратов значений
    # np.log(X[i]).
    for i in range (0, n):
        sum_ln2x += pow(np.log(X[i]), 2)

    # инициализируется нулём и затем увеличивается путем сложения квадратов значений X[i] * X[i].
    for i in range(0, n):
        sum_powx += X[i] * X[i]

    # Затем, выводим суммы различных величин
    print(sum_ylnx)
    print(sum_lnx)
    print(sum_y)
    print(sum_ln2x)

    # Вычисляются коэффициенты линейной аппроксимации
    b = (n * sum_ylnx - sum_lnx * sum_y) / (n * sum_ln2x - pow(sum_lnx, 2))
    a = ((1 / n) * sum_y) - ((b / n) * sum_lnx)

    # Выводим проценты коофициента
    print("Коеф-ты:\na = ", a, "\nb = ", b)

    # Здесь вычисляем коэффициента сглаживания C
    sum_Ex = 0
    # sum_Ex инициализируется нулём и затем увеличивается путем сложения разности (Y[i] - b *
    # np.log(X[i]) - a) умноженной на X[i] для каждого элемента i.
    for i in range(0, n):

```

```

    sum_Ex += (Y[i] - b * np.log(X[i]) - a) * X[i]
C = (-sum_Ex) / (sum_powx)
print("Сглаживание C = ", C)

# Здесь вычисляется среднеквадратичное отклонение G
G = 0
sum_G = 0
# sum_G инициализируется нулём и затем увеличивается путем сложения квадратов разности
(Y[i] - b * np.log(X[i]) - a) для каждого элемента i.
# G вычисляется как квадратный корень из отношения sum_G к количеству элементов n.
for i in range(0, n):
    sum_G += pow((Y[i] - b * np.log(X[i]) - a), 2)
G = math.sqrt(sum_G / n)
print("Среднеквадратичное отклонение равно: ", G)

# Вычисляем погрешность flt
flt = 0
# flt инициализируется нулём и затем увеличивается путем сложения квадратов разности (Y[i] - (b
* np.log(X[i]) + a)) для каждого элемента i.
for i in range(0, n):
    flt += pow(Y[i] - (b * np.log(X[i]) + a), 2)
print("Погрешность: ", flt)

# Здесь вычисляем аппроксимированные значения y_ap
# В цикле for для каждого элемента i от 0 до n, вычисляется значение b * np.log(X[i]) + a + C *
X[i], которое добавляется в список y_ap
y_ap = []
for i in range(0, n):
    y_ap.append(b * np.log(X[i]) + a + C * X[i])
print(y_ap)

# Здесь вычисляются начальные значения y_start до применения сглаживания
# В цикле for для каждого элемента i от 0 до n, вычисляется значение b * np.log(X[i]) + a, которое
добавляется в список y_start.
y_start = []
for i in range(0, n):
    y_start.append(b * np.log(X[i]) + a)

# Визуализация
plt.plot(X, Y, 'ro', label='Узлы')
plt.plot(X, y_ap, 'b', label='После сглаживания')
plt.plot(X, y_start, 'y', label='y = a * ln(x) + b до сглаживания')
plt.title('Аппроксимация (Логарифмическая)')
plt.grid()
plt.xlabel('x')
plt.ylabel('y')
plt.legend()
plt.show()

# Линейная аппроксимация если выбрана 2
else:
    # Ищем решение системы
    sum_y = 0
    sum_x = 0

```

```

sum_powx = 0
sum_powy = 0
sum_xy = 0
for i in range(0, n):
    sum_y += Y[i]
for i in range(0, n):
    sum_x += X[i]
for i in range(0, n):
    sum_powx += X[i] * X[i]
for i in range(0, n):
    sum_powy += Y[i] * Y[i]
for i in range(0, n):
    sum_xy += X[i] * Y[i]
print("Суммы")
print(sum_y)
print(sum_x)
print(sum_powx)
print(sum_powy)
print(sum_xy)

# Поиск коэффициентов
a = (sum_x * sum_y - n * sum_xy) / (pow(sum_x, 2) - n * sum_powx)
b = (sum_x * sum_xy - sum_powx * sum_y) / (pow(sum_x, 2) - n * sum_powx)
print("Коеф-ты:\na = ", a, "\nb = ", b)

# Отклонение
# sum_Ex инициализируется нулём и затем увеличивается путем сложения выражения ((a * X[i] +
b) - Y[i]) * X[i] для каждого элемента i.
sum_Ex = 0
for i in range(0, n):
    sum_Ex += ((a * X[i] + b) - Y[i]) * X[i]
print("Ex", sum_Ex)
C = (-sum_Ex) / (sum_powx)
print("Сглаживание C = ", C)

# Среднеквадратичное отклонение
# sum_G инициализируется нулём и затем увеличивается путем сложения квадратов выражения
(Y[i] - (a * X[i] + b)) для каждого элемента i
G = 0
sum_G = 0
for i in range(0, n):
    sum_G += pow((Y[i] - (a * X[i] + b)), 2)
G = math.sqrt(sum_G/n)
print("Среднеквадратичное отклонение равно: ", G)

# Погрешность
# flt инициализируется нулём и затем увеличивается путем сложения квадратов выражения (Y[i] -
(a * X[i] + b)) для каждого элемента i
flt = 0
for i in range(0, n):
    flt += pow(Y[i] - (a * X[i] + b), 2)
print("Погрешность: ", flt)
print("y = a * x - b до сглаживания")

```

```

#после сглаживания
# В цикле for для каждого элемента i от 0 до n, вычисляется значение (a * X[i] + b) + C * X[i],
которое добавляется в список y_ap
y_ap = []
for i in range(0, n):
    y_ap.append((a * X[i] + b) + C * X[i])
print(y_ap)
y_start = []

# до сглаживания
# В цикле for для каждого элемента i от 0 до n, вычисляется значение a * X[i] + b, которое
добавляется в список y_start
for i in range(0, n):
    y_start.append(a * X[i] + b)

# Визуализация
plt.plot(X, Y, 'ro', label='Узлы')
plt.plot(X, y_ap, 'b', label='После сглаживания')
plt.plot(X, y_start, 'y', label='y=a*x-b до сглаживания')
plt.title('Аппроксимация(Линейная)')
plt.grid()
plt.xlabel('x')
plt.ylabel('y')
plt.legend()
plt.show()

```

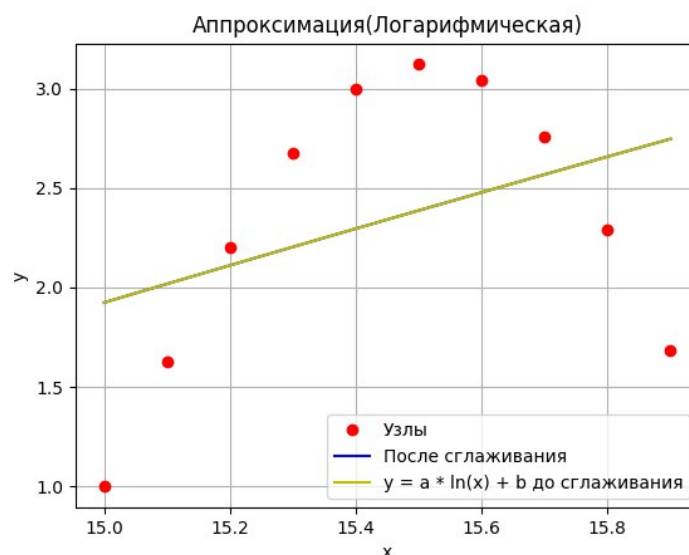
Результат работы программы 2

При выборе 1 создается график аппроксимации логарифмической

```

Сдвиг по Y на: 1
Сдвинутый массив:
[0.99909, 1.624825, 2.2038320000000002, 2.677044, 2.9946479999999998, 3.12132, 3.040105, 2.754519, 2.2886290000000002, 1.6850619999999998]
Выберите тип аппроксимации:
1)Логарифмическая
2)Линейная
64.07491752627043
27.374361410387355
23.389073999999997
74.93902409441682
Коеф-ты:
a = -36.31089531295811
b = 14.11897875297731
Сглаживание C = 6.113052956944653e-06
Среднеквадратичное отклонение равно: 0.6232913172747293
Погрешность: 3.88492066190069
[1.9240996341943086, 2.0179144029677354, 2.111109931866809, 2.203694342158089, 2.2956755963840454, 2.3870615024724855, 2.4778597177138773, 2.5680777526115373, 2.657722974609627, 2.7468026117035724]

```

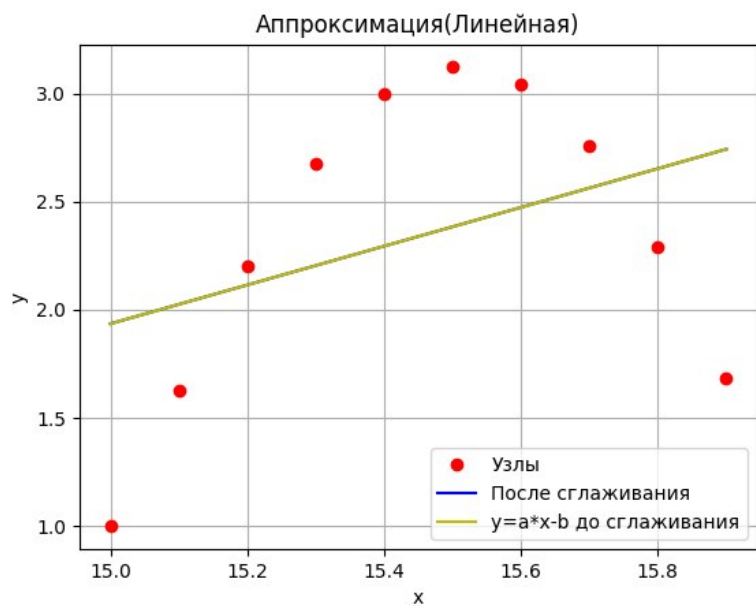


При выборе 2 создается график аппроксимации линейной

```

Сдвиг по Y на: 1
Сдвинутый массив:
[0.99909, 1.624825, 2.2038320000000002, 2.677044, 2.9946479999999998, 3.12132, 3.040105, 2.754519, 2.2886290000000002, 1.6850619999999998]
Выберите тип аппроксимации:
1)Логарифмическая
2)Линейная
.
Суммы
23.389073999999997
154.5
2387.85
59.27910233306
362.1006766
Коеф-ты:
a = 0.8963433939394291
b = -11.509598036364574
Ex -6.091482873671339e-11
Сглаживание C = 2.5510324658882842e-14
Среднеквадратичное отклонение равно: 0.625411313011597
Погрешность: 3.9113931044288974
y = a * x - b до сглаживания
[1.9355528727272453, 2.0251872121211902, 2.1148215515151354, 2.2044558909090823, 2.2940902303030275, 2.383724569696972, 2.4733589090909174, 2.5629932484848625, 2.6526275878788095, 2.742261927272754]

```



Заключение

В результате мы смогли реализовать интерполяцию квадратичного сплайна дефекта 2 и аппроксимацию линейной и экспоненциальной функций на языке python, с помощью библиотеки matplotlib. Получили коэффициенты a, b, c и построили график интерполяции и аппроксимации функций.

Источники

1. 1. Калиткин Н.Н. Численные методы. [Электронный ресурс] / Н.Н. Калиткин. - М.: Питер, 2001. С.504.
2. Галеева Г.Я., Маликова Л.Е., Фазылов А.Р. Учебное пособие по численным методам. – Уфа, 2013.
3. Численное интегрирование [Электронный ресурс] - Режим доступа: http://ru.wikipedia.org/wiki/Численное_интегрирование
4. Семакин И.Г. Основы программирования. [Текст] / И.Г. Семакин, А.П. Шестаков. - М.: Мир, 2006. С.346.
5. Численное интегрирование функции [Электронный ресурс] - Режим доступа: <https://math.semestr.ru/optim/numerical-integration>