

**«Санкт-Петербургский государственный электротехнический университет
«ЛЭТИ» им. В.И. Ульянова (Ленина)»
(СПбГЭТУ «ЛЭТИ»)**

Направление	09.03.04 Программная инженерия
Профиль	Разработка программно-информационных систем
Факультет	КТИ
Кафедра	МО ЭВМ

К защите допустить

Зав. кафедрой

Кринкин К.В.

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
БАКАЛАВРА**

**Тема: РАЗРАБОТКА И ПРОГРАММНАЯ РЕАЛИЗАЦИЯ
АППРОКСИМАЦИИ ВРЕМЕННЫХ РЯДОВ С ПОМОЩЬЮ В-
СПЛАЙНОВ**

Студентка

_____ *подпись*

Нгуен Тхи Тху Зуен

Руководитель

Д.Т.Н.,
профессор

_____ *подпись*

Середа А.-В. И.

Консультанты

К.Т.Н.

_____ *подпись*

Заславский М.М.

Санкт-Петербург

2021

ЗАДАНИЕ НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ

Утверждаю
Зав. кафедрой МО ЭВМ
к.т.н. Кринкину К. В.
« » _____ 2021 г.

Студентка Нгуен Т.Т.З. Группа 7304

Тема работы: Разработка и программная реализация аппроксимации временных рядов с помощью В-сплайнов

Место выполнения ВКР: Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В.И. Ульянова (Ленина)

Исходные данные (технические требования):

ОС Windows, технологии программирования выбираются исполнителем

Разработка программного модуля, позволяющий осуществлять аппроксимацию временных рядов с помощью В-сплайнов.

Содержание ВКР:

Введение, Обзор литературы по теме работы, Аппроксимация данных с помощью В-сплайнов, Разработка программного модуля, Проведение вычислительных экспериментов и анализ полученных результатов, Специальные вопросы обеспечения безопасности, заключение.

Перечень отчетных материалов: пояснительная записка, иллюстративный материал.

Дополнительные разделы: Безопасность жизнедеятельности (каф. БЖД)

Дата выдачи задания

« 22 » апреля _____ 2021 г.

Дата представления ВКР к защите

« 3 » июня _____ 2021 г.

Студентка

Нгуен Т.Т.З.

Руководитель д.т.н., профессор
(Уч. степень, уч. звание)

Середа А.-В. И.

КАЛЕНДАРНЫЙ ПЛАН ВЫПОЛНЕНИЯ ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ

Утверждаю

Зав. кафедрой МО ЭВМ

к.т.н. Кринкину К. В.

« » 2021 г.

Студентка Нгуен Т.Т.3.

Группа 7304

Тема работы: Разработка и программная реализация аппроксимации временных рядов с помощью В-сплайнов

№ п/п	Наименование работ	Срок выполнения
1	Обзор литературы по теме работы	22.04 – 25.04
2	Изучение метода аппроксимации данных с помощью В-сплайнов	25.04 – 27.04
3	Разработка программного модуля	27.04 – 10.05
4	Проведение вычислительных экспериментов и анализ полученных результатов	10.05 – 15.05
5	Специальные вопросы обеспечения безопасности жизнедеятельности	15.05 – 17.05
6	Оформление пояснительной записки	17.05 – 25.05
7	Оформление иллюстративного материала	25.05 – 30.05
8	Предзащита	01.06

Студентка

Нгуен Т.Т.3.

Руководитель

Д.Т.Н., профессор
(Уч. степень, уч. звание)

Середа А.-В. И.

подпись

РЕФЕРАТ

Выпускная квалификационная работа содержит пояснительную записку объёмом 55 страниц, 14 рисунков, 5 таблиц, 22 источников информации.

РАЗРАБОТКА И ПРОГРАММНАЯ РЕАЛИЗАЦИЯ АППРОКСИМАЦИИ ВРЕМЕННЫХ РЯДОВ С ПОМОЩЬЮ В-СПЛАЙНОВ

Ключевые слова: аппроксимация, интерполяция, В-сплайн.

Объектом исследования является аппроксимация временных рядов с помощью В-сплайна.

Предметом исследования является представление аппроксимации данных путем исключения случайных величин при сохранении точности важных характеристик исходных.

Цель данной исследовательской работы, таким образом, разработать и реализовать программный модуль, позволяющий осуществлять аппроксимацию временных рядов с помощью В-сплайнов.

Для достижения цели необходимо решить следующие задачи:

- В первой части даны задачи аппроксимации данных и обзор методов существующих подходов к решению;
- Во второй части изучен метод аппроксимации данных с помощью В-сплайнов;
- В третьей части рассмотрен и реализован программный модуль;
- В четвертой части рассмотрено проведение вычислительных экспериментов и анализ полученных результатов;
- В пятой части рассмотрены специальные вопросы безопасности жизнедеятельности.

ABSTRACT

In this project report, the methods of processing interpolation and data approximation are utilized along with the development of the time series approximation method with B-spline. To achieve the project aim and objectives, the first chapter provides an overview of the existing interpolation and approximation methods. Then the method of data approximation using B-splines is studied. An application, which produces graphics based on the given data with the ability to adjust parameters and coefficients by using the methods of interpolation and approximation is developed and implemented. Computational experiments and analysis of the results are conducted to test the application.

СОДЕРЖАНИЕ

РЕФЕРАТ	4
ABSTRACT	5
ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ	8
ВВЕДЕНИЕ	9
1. ОПИСАНИЕ ОСНОВНЫХ МЕТОДОВ ИНТЕРПОЛЯЦИИ И АППРОКСИМАЦИИ ДАННЫХ	11
1.1. Полиномиальная интерполяция и аппроксимация данных	11
1.1.1. Аппроксимации данных с помощью полиномов	11
1.1.2. Интерполяция полиномами в форме многочлена Лагранжа	13
1.1.3. Интерполяция полиномами в форме многочлена Ньютона	14
1.2. Кусочно-полиномиальная интерполяция (кубический сплайн)	16
1.3. Выводы по разделу	20
2. АППРОКСИМАЦИИ ДАННЫХ С ПОМОЩЬЮ В-СПЛАЙНОВ	22
2.1. Подход к построению В-сплайнов	22
2.2. Задача аппроксимации таблично заданных функций	24
2.3. Выводы по разделу	32
3. РАЗРАБОТКА ПРОГРАММНОГО МОДУЛЯ	33
3.1. Краткое описание	33
3.2. Описание алгоритмов методов	33
3.2.1. Полиномиальная интерполяция и аппроксимация	33
3.2.2. Кубический сплайн-интерполяция	33
3.2.3. Аппроксимация В-сплайнами	34
3.4. Общая архитектура решения	35
3.5. Используемые технологии	36

3.6. Интерфейс пользователя приложения программы	37
3.6.1. Сценарии использования:	38
3.6.2. Графическая схема	39
3.7. Выводы по разделу	39
4. ПРОВЕДЕНИЕ ВЫЧИСЛИТЕЛЬНЫХ ЭКСПЕРИМЕНОВ И АНАЛИЗ ПОЛУЧЕННЫХ РЕЗУЛЬТАТОВ	40
4.1. Основные характеристики ПК	40
4.2. Тестирование программы	41
4.3. Анализ полученных результатов	44
5. СПЕЦИАЛЬНЫЕ ВОПРОСЫ ОБЕСПЕЧЕНИЯ БЕЗОПАСНОСТИ ЖИЗНЕДЕЯТЕЛЬНОСТИ	46
5.1. Оценка аспектов эргономики программного обеспечения	46
5.2. Оценка удобства и пригодности при использовании Python	49
5.3.1. Принципы организации диалога для выполнения задания.....	51
5.3.2. Дополнительные принципы проектирования	52
5.4. Выводы по разделу	54
ЗАКЛЮЧЕНИЕ	55
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	56

ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

В настоящей пояснительной записке применяют следующие термины с соответствующими определениями:

ВКР – выпускная квалификационная работа

СЛУ/СЛАУ - Система линейных уравнений или система линейных алгебраических уравнений

Python (в русском языке встречаются названия пито́н^[1] или пайтон) - высокоуровневый язык программирования

GUI (англ. graphical user interface) - графический интерфейс пользователя

Макет UI(англ. wireframe) - макет интерфейса пользователя

ПЭВМ - персональная электронно-вычислительная машина

CPU (англ. Central Processing Unit) - центральное обрабатывающее устройство

RAM (англ. Random-access memory) - память с произвольным доступом

ИСО (англ. International Organization for Standardization) - Международная организация по стандартизации

ВВЕДЕНИЕ

Актуальность. Как известно, временной ряд — это собранный в разные моменты времени статистический материал о значении каких-либо параметров исследуемого процесса. В настоящее время временные ряды являются важной формой данных в различных областях промышленности и исследований, и представления временных рядов вызывает большой интерес. В последние десятилетия интеллектуальный анализ данных временных рядов привлек внимание и стал быстро развиваться. Временные ряды по своей природе являются часто интеграционными, т.е. значения временного ряда в некоторые моменты времени зависят от значений ряда в предыдущие моменты времени. Свойства временных рядов, такие как большой объем данных, случайные данные, тренд, различные сглаживания, сезонность, медленно и быстро меняющиеся компоненты, циклическая нерегулярность, приводят к тому, что представления данных временных рядов становится все более сложными.

В настоящее время, после 30 лет интенсивной разработки В-сплайны стали стандартными инструментами в компьютерной графике, геометрическом моделировании, автоматизированном проектировании и многих других междисциплинарных областях. А также решается задача аппроксимации данных, которые предоставляют более точное отображение по сравнению с характеристиками исходных данных.

Цель данной исследовательской работы, таким образом, разработать и реализовать программный модуль, позволяющий осуществлять аппроксимацию временных рядов с помощью В-сплайнов.

Для достижения цели необходимо решить следующие **задачи**:

1. Постановка задачи аппроксимации данных и обзор существующих методов для её решения ;
2. Изучить метод аппроксимации данных с помощью В-сплайнов;
3. Реализовать программный модуль;

4. Провести вычислительные эксперименты и проанализировать полученные результаты;
5. Выявить специальные вопросы безопасности жизнедеятельности.

Объектом исследования являются аппроксимации временных рядов с помощью В-сплайна.

Предметом исследования является представление аппроксимации данных путем исключения случайных величин при поддержании точности важных характеристик исходных данных.

Практическая значимость решения заключается в том, что программный модуль, представляющий данные, даёт возможность обеспечить лучший обзор отслеживаемых данных и получить более эффективный результат работы.

1. ОПИСАНИЕ ОСНОВНЫХ МЕТОДОВ ИНТЕРПОЛЯЦИИ И АППРОКСИМАЦИИ ДАННЫХ

Численные вычисления представляют собой важнейшую роль в естественных и технических науках. В настоящее время цифровые гуманитарные науки или биотехнологии, независимо от того, разработаны новые материалы или созданы системы искусственного интеллекта, включают в себя некоторое количество численных вычислений. Благодаря отображению данных можно получить обзор характеристик числовых рядов. С помощью алгоритмов интерполяции и аппроксимации данных представление становится более простым. В этом разделе рассматривается задача интерполяции и аппроксимации данных.

Интерполяция (интерполяция) - это построение функции на основе ее известных значений в заданных точках^[2]. Совершенно очевидно, что можно построить бесконечный набор функций, проходящих через зафиксированные точки. Полиномиальная интерполяция предполагают, что полиномы рассматривают как интерполирующие функции.

Аппроксимация данных – это метод, основанный на замене полученных данных аналитической функцией, которая наиболее приближенно проходит или совпадает в узловых точках с исходными значениями.

1.1. Полиномиальная интерполяция и аппроксимация данных

1.1.1. Аппроксимации данных с помощью полиномов

Пусть набор из n точек (x_i, y_i) на промежутке $[a, b]$ заданы в $n + 1$ узлах, и необходимо аппроксимировать эти данные некоторой функцией $f(x)$.

Представим аппроксимирующую функцию в виде суммы произведений коэффициентов c_0, c_1, \dots, c_m и базисных функций g_0, g_1, \dots, g_m :

$$f(x) = c_0 g_0(x) + c_1 g_1 + c_2 g_2(x) + \dots + c_{m-1} g_{m-1}(x) + c_m g_m(x) \quad (1.1.1)$$

Во многих случаях в качестве аппроксимирующей функции выбирают степенной полином:

$$f(x) = P_m(x) = c_0 + c_1x + c_2x^2 + \dots + c_{m-1}x^{m-1} + c_mx^m \quad (1.1.2)$$

В этом случае система базисных функций представляет собой последовательность $1, x, x^2, x^3, \dots, x^{m-1}, x^m$.

В результате получаем систему линейных уравнений:

$$y_i = c_0 + c_1 \cdot x_i + c_2 \cdot x_i^2 + \dots + c_m \cdot x_i^m, i = 0, 1, \dots, n \quad (1.1.3)$$

которая имеет $n+1$ уравнений и $m+1$ неизвестных (коэффициенты $c_i, i=0,1,2,\dots,m$). Если $n>m$, система (1.1.3) является переопределенной и ее решение ищется в смысле метода наименьших квадратов (МНК).

В матричной форме система (1.1.3) имеет вид:

$$\begin{pmatrix} 1 & x_0 & x_0^2 & \dots & x_0^{m-1} & x_0^m \\ 1 & x_1 & x_1^2 & \dots & x_1^{m-1} & x_1^m \\ 1 & x_2 & x_2^2 & \dots & x_2^{m-1} & x_2^m \\ 1 & x_3 & x_3^2 & \dots & x_3^{m-1} & x_3^m \\ \vdots & \vdots & \vdots & \dots & \vdots & \vdots \\ 1 & x_{n-1} & x_{n-1}^2 & \dots & x_{n-1}^{m-1} & x_{n-1}^m \\ 1 & x_n & x_n^2 & \dots & x_n^{m-1} & x_n^m \end{pmatrix} \cdot \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \\ c_{m-1} \\ c_m \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_{n-1} \\ y_n \end{pmatrix} \quad \text{или} \quad Ac = y \quad (1.1.4)$$

Для решения системы можно, в частности, использовать первую трансформацию Гаусса, позволяющую перейти к решению системы вида:

$$A^T Ac = A^T y \quad (1.1.5)$$

с квадратной неособенной матрицей.

Таким образом, в результате вычислений СЛУ по формуле (1.1.5) для нахождения коэффициентов $c_i, i = 0, 1, 2, \dots, m$. Затем вычислить (1.1.3), чтобы получить полиномиальную аппроксимацию.

Замечание:

В случае $n = m$ (число неизвестных равно числу уравнений), то есть на промежутке $[a, b]$ заданы в $n+1$ узлах одномерной сетки x_0, x_1, \dots, x_n и соответствующие $n+1$ значения y_0, y_1, \dots, y_n существует уникальный многочлен степени не выше n , который интерполирует данные $\{(x_0, y_0), \dots, (x_n, y_n)\}$. Тогда полином (1.1.2) имеет вид:

$$f(x) = P_n(x) = c_0 + c_1x + \dots + c_{n-1}x^{n-1} + c_nx^n \quad (1.1.6)$$

Степени n , значения которых в заданных точках $x_k, k = 0, 1, \dots, n$, совпадают со значениями функции $f(x)$ в этих точках.

Для любой непрерывной функции $f(x)$ сформулированная задача имеет единственное решение. Действительно, для нахождения коэффициентов c_0, c_1, \dots, c_n получаем СЛАУ:

$$y_i = f(x_i) = c_0 + c_1 x_i + c_2 x_i^2 + \dots + c_n x_i^n, \quad i = 0, 1, \dots, n, \quad (1.1.7)$$

Решение системы уравнений (1.1.7) можно записать различным образом. Наиболее употребительна запись интерполяционного многочлена в форме Лагранжа.

1.1.2. Интерполяция полиномами в форме многочлена Лагранжа

Интерполяционный многочлен Лагранжа — многочлен минимальной степени, принимающий данные значения в заданном наборе точек. Они были предложены потому, что давали некоторые преимущества в каких-то частных случаях: например, если сетка равномерна или заранее известно количество узлов, используемых для построения полинома, и т.д. Это позволяет представить многочлен $L_n(x)$ в виде линейной комбинации следующим образом:

$$L_n(x) = \sum_{k=0}^n c_k(x) f(x_k) \quad (1.1.8)$$

Многочлен $L_n(x)$, удовлетворяющий условиям:

$$f(x_i) = L_n(x_i), \quad i = 0, 1, \dots, n \quad (1.1.9)$$

называется интерполяционным многочленом для функции $f(x)$, построенным по узлам $\{x_i\}_0^n$. Найдем явное выражение для коэффициентов $c_k(x)$. Из условий интерполирования (1.1.9) получаем $\sum_{k=0}^n c_k(x_i) f(x_k) = f(x_i), i = 0, 1, \dots, n$.

Эти соотношения будут выполнены, если на функции $c_k(x)$ наложить условия

$$c_k(x_i) = \begin{cases} 0, & i \neq k \\ 1, & i = k, i = 0, 1, \dots, n, \end{cases}$$

которые означают, что каждая из функций $c_k(x), k = 0, 1, \dots, m$, имеет не менее n нулей на $[a, b]$.

Поскольку $L_n(x)$ — многочлен степени n , коэффициенты $c_k(x)$ интерполяционного многочлена (1.1.8) находятся по формулам:

$$c_k(x) = \frac{\prod_{j \neq i} (x - x_j)}{\prod_{j \neq i} (x_i - x_j)} \quad (1.1.10)$$

Полиномы определяются по формуле:

$$L_n(x) = \sum_{i=0}^n \frac{\prod_{j \neq i} (x - x_j)}{\prod_{j \neq i} (x_i - x_j)} \quad (1.1.11)$$

В результате вычислений управления (1.1.11) будет определен многочлен. При этом порядок многочлена n всегда равен количеству узлов в сетке на единицу ($n = m + 1$).

1.1.3. Интерполяция полиномами в форме многочлена Ньютона

Алгебраический интерполяционный многочлен единственный, но существует много форм его записи. Далее будет приведена форма Ньютона; она удобна тем, что пригодна на произвольной неравномерной сетке и позволяет увеличивать или уменьшать число узлов в ходе расчета, а также апостериорно оценивать достигнутую точность^[3].

Определим многочлен (1.1.6) по его разделенным разностям:

$$\begin{aligned} P(x) &= P(x_0) + (x - x_0)P(x, x_0); \\ P(x, x_0) &= P(x_0, x_1) + (x - x_1)P(x, x_0, x_1); \\ P(x, x_0, x_1) &= P(x_0, x_1, x_2) + (x - x_2)P(x, x_0, x_1, x_2), \dots; \\ P(x, x_0, x_1, \dots, x_n) &= 0. \end{aligned}$$

Подставив здесь каждую последующую формулу в предыдущую, мы производим выражение многочлена (1.1.6) через его разделенные разности:

$$f(x) = P_n(x) = P(x_0) + \sum_{i=1}^n (x - x_0)(x - x_1) \dots (x - x_{i-1})P(x_0, x_1, \dots, x_i) \quad (1.1.12)$$

Для расчетов по формуле (1.1.12) исходная таблица составляется из $n+1$ узлов сетки x_0, x_1, \dots, x_n . В таб. 1.1 приведены разделенные разности многочлена в виде треугольной таблицы.

Таблица 1.1: Разделенные разности

		Разделенные разности			
		1	2	...	n
x_0	$P(x_0)$	$P(x_0, x_1)$ $P(x_1, x_2)$ \vdots $P(x_n, x_{n-1})$	$P(x_0, x_1, x_2)$ \vdots $P(x_{n-2}, x_{n-1}, x_n)$...	$P(x, x_0, x_1, \dots, x_n)$ $= 0$
x_1	$P(x_1)$				
x_2	$P(x_2)$				
\vdots	\vdots				
x_{n-1}	$P(x_{n-1})$				
x_n	$P(x_n)$				

Для окончательных вычислений достаточно сохранить лишь одну верхнюю строку разделенных разностей всех порядков, входящих в формулу (1.1.12). Остальная часть таблицы является промежуточным результатом и служит только для вычисления этой строки[3].

Замечание:

Поскольку порядок полинома Лагранжа увеличивается с увеличением числа узлов во время интерполяции, ненадежность разделенных разностей высокого порядка скажется на погрешности самой интерполяции, и в начале и в конце последовательности могут возникнуть шумы. Рассмотрим погрешность интерполирования полинома.

Остаточный член интерполяционной формулы. Заменяя функцию $f(x)$ интерполяционным многочленом $P_n(x)$, мы допускаем погрешность

$$R_n(x) = f(x) - P_n(x) \quad (1.1.12)$$

которая называется погрешностью интерполирования или, что то же самое, остаточным членом интерполяционной формулы.

Если $f(x)$ является $n + 1$ раз непрерывно дифференцируемых на отрезке (a, b) и есть многочлен степени не выше n , который интерполирует $f(x)$ в $n + 1$ различных узлов $\{x_i\}_0^n$ в этого интервала, то для каждого x в этом интервале существует ξ такое, что

$$R_n(x) = |f(x) - P_n(x)| = \frac{f^{(n+1)}(\xi)}{(n+1)!} \prod_{i=0}^n (x - x_i) \quad (1.1.13)$$

где $\xi(x)$ - точка, расположенная между $a = x_0$ и $b = x_n$. Ошибка может быть количественно измерена с помощью 2-норма $R_n(x)$:

$$\epsilon = \|R_n(x)\|_2 = \left(\int_a^b R_n^2(x) dx \right)^{\frac{1}{2}} \quad (1.1.14)$$

1.2. Кусочно-полиномиальная интерполяция (кубический сплайн)

Для построения графиков функции по точкам между каждой парой соседних узлов строился свой полином третьей степени, а в узлах соседние полиномы «склеивались» так, чтобы обеспечить максимально возможную гладкость интерполяции. Этот прием - интерполирование функции кусочно-полиномиальной функцией получил название сплайн-интерполяции^[3].

Современная теория сплайн-интерполяции и прикладной математический аппарат для использования сплайнов с целью интерполяции и аппроксимации функций начали интенсивно развиваться с середины XX века. В настоящее время этот описанный прием быстро распространился и был обобщен на случай полинома большей степени. При решении задач интерполяции преимущество сплайнов перед интерполяционными полиномами заключается в гарантии устойчивости вычислительного процесса и сходимости сплайн-интерполяции при увеличении количества узлов сетки. В настоящем разделе рассмотрена задача интерполирования, таблично заданная на отрезке $[a, b]$ функции $f(x)$, с использованием кубических сплайнов, которые широко применяются для интерполяции функций [4].

1.2.1. Постановка задачи

Пусть значения функции $f(x)$ на промежутке $[a, b]$ заданы в узлах одномерной сетки

$$\eta_x = \{x_i / x_i = x_{i-1} + h_i, h_i > 0, i = 1, 2, 3, \dots, n; x_0 = a, x_n = b\}, \quad (1.2.1)$$

где $x_i, i = 0, 1, 2, \dots$;

n – узлы одномерной сетки η_x ;

$h_i, i = 1, 2, 3, \dots, n$ – шаг сетки η_x . Если $h_i = h \forall i = 1, 2, 3, \dots, n$, сетка η_x равномерная.

Если сплайн в заданных точках совпадает с табулированной функцией, то такой сплайн называется интерполяционным (или интерполяцией). В случае табличного задания функции на отрезке $[a, b]$ часто возникает проблема восстановления значений функции $f(x)$ в произвольной точке $x \in (a, b)$ между узлами сетки η_x .

1.2.2. Сплайн-интерполяция таблично заданной функции

Пусть на отрезке $[a, b]$ задана одномерная сетка $\eta_x = \{x_i / x_i = x_{i-1} + h_i, h_i > 0, i = 1, 2, 3, \dots, n; x_0 = a, x_n = b\}$, в узлах x_i известны значения $y_i = f(x_i), i = 0, 1, 2, \dots, n$ – соответствующие значения функции $f(x)$. Очевидно, что все узлы сетки различны и упорядочены по возрастанию.

Кубическим сплайном будем называть функцию $S(x)$, заданную на отрезке $[a, b]$ и удовлетворяющую следующим условиям:

1. $S(x_i) = y_i, i = 0, 1, 2, \dots, n$.
2. $S(x), S'(x)$ и $S''(x)$ непрерывны на отрезке $[a, b]$.

В этом случае на каждом интервале интерполяционный сплайн $[x_{i-1}, x_i], i = 1, 2, \dots, n$ $S(x)$ представляет собой полином 3-й степени. Его удобно записать в следующем виде::

$$S_i(x) = a_i + b_i(x - x_i) + (c_i / 2)(x - x_i)^2 + (d_i / 6)(x - x_i)^3, \quad (1.2.2)$$

здесь $x \in [x_{i-1}, x_i], i = 1, 2, \dots, n$; коэффициенты a_i, b_i, c_i, d_i свои для каждого интервала и определяют из условий в узлах.

Существует условие естественного кубического сплайна: для вычисления первой и второй производных многочлена, требуется непрерывность этих производных (т.е. создается гладкость сплайна) во всех точках, включая узлы. Таким образом: $S''_n(a) = S''_n(b) = 0$.

На рис. 1 изображен график кубического «естественного» сплайна по трем точкам.

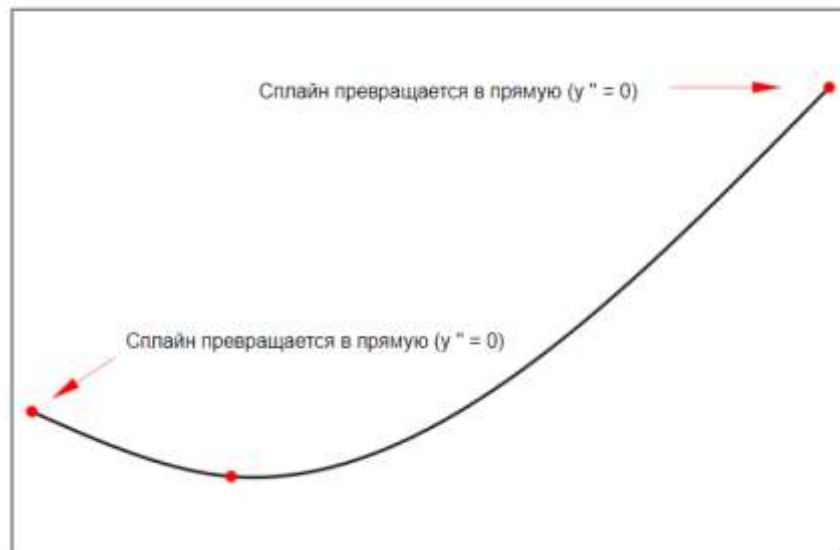


Рисунок 1 – Интерполяция кубическими «естественными» сплайнами по трем точкам.

Для построения кубического сплайна на отрезке $[a, b]$ требуется найти коэффициенты $a_i, b_i, c_i, d_i, i = 1, 2, \dots, n$ – всего $4n$ неизвестных величин. При нахождении их в соответствии с определением кубического сплайна имеются условия:

- совпадения значений $S(x)$ и таблично заданной функции $f(x)$ в узлах сетки η_x : $S(x_i) = y_i, i = 0, 1, 2, \dots, n$;
- непрерывности функции $S(x)$, ее первой и второй производных.

Эти условия сводятся к необходимости обеспечения непрерывности $S(x)$ и ее производных во всех внутренних узлах $x_i, i = 1, 2, \dots, n - 1$ сетки η_x , так как в этих точках наблюдается изменение аналитического задания кусочно-полиномиальной функции $S(x)$, в остальных точках отрезка $[a, b]$ такая функция непрерывна по определению^[4].

Следовательно, для n интервалов существует $4n$ неизвестных (по 4 коэффициента для каждой кубики) $a_i, b_i, c_i, d_i, i = 1, 2, \dots, n$ имеется только $(n + 1) + 3(n - 1) = 4n - 2$ условий.

Коэффициенты $a_i, i = 1, 2, \dots, n$ определяются из соотношений

$$a_i = y_i, i = 1, 2, \dots, n. \quad (1.2.3)$$

Для нахождения коэффициентов $c_i, i = 1, 2, \dots, n$ необходимо решить трехдиагональную систему линейных уравнений, приводящую к следующему виду:

$$\begin{cases} c_{i-1} \frac{h_i}{6} + c_i \frac{(h_i + h_{i+1})}{3} + c_{i+1} \frac{h_{i+1}}{6} = \frac{y_{i+1} - y_i}{h_{i+1}} - \frac{y_i - y_{i-1}}{h_i}, i = 1, 2, \dots, n-1, \\ c_0 = 0, c_n = 0. \end{cases} \quad (1.2.4)$$

Это система $n - 1$ уравнения для $n + 1$ неизвестных коэффициентов c_n . Очевидно, что эта система имеет матрицу с диагональным преобладанием, у нее единственное решение, для нахождения которого может быть использован метод Прогонки (см. раздел 1.2.3).

После того как коэффициенты $c_i, i = 1, 2, \dots, n$ будут найдены, коэффициенты $d_i, i = 1, 2, \dots, n$ могут быть определены по формуле:

$$d_i = (c_i - c_{i-1}) / h_i, i = 1, 2, \dots \quad (1.2.5)$$

Наконец, находим коэффициенты $b_i, i = 1, 2, \dots, n$ по формуле:

$$b_i = (a_i - a_{i-1}) / h_i + c_i h_i / 2 - d_i h_i^2 / 6, i = 1, 2, \dots, n. \quad (1.2.6)$$

Таким образом, в результате вычислений будет построен интерполяционный кубический сплайн $S(x)$.

1.2.3. Описание метода прогонки при использовании решения трехдиагональной системы

Пусть трехдиагональная система $Ax = b, b \in R^n$, где A – трехдиагональная матрица, определитель которой порядка n может быть вычислен из трехчленного рекуррентного соотношения^[5].

Система запишется в виде следующим образом:

$$\begin{cases} b_1 x_1 + c_1 x_2 & = d_1 \\ a_2 x_1 + b_2 x_2 + c_2 x_3 & = d_2 \\ c_3 x_2 + d_3 x_3 + e_3 x_4 & = d_3 \\ \dots & \dots & \dots \\ a_{n-1} x_{(n-2)} + b_{n-1} x_{n-1} + c_{n-1} x_n & = d_{n-1} \\ a_n x_{n-1} + b_n x_n & = d_n \end{cases}$$

Метод прогонки состоит из двух этапов: прямой прогонки и обратной прогонки. На первом этапе определяются прогоночные коэффициенты, а на втором – находят неизвестные $x_1, x_2, x_3, \dots, x_n$. Выпишем формулы метода Прогонки в порядке их применения.

а. Прямой ход метода прогонки (вычисление вспомогательных величин):

$$\alpha_1 = -\frac{c_1}{b_1}; \quad \beta_1 = \frac{d_1}{b_1}$$

...

$$\alpha_i = -\frac{c_i}{a_i \alpha_i} + b_i;$$

$$\beta_i = \frac{\alpha_i - a_i \beta_{i-1}}{a_i \alpha_{i-1} + b_i}, i = 2, 3, \dots, n-1$$

б. Обратный ход метода прогонки (нахождение решения):

$$x_n = \frac{\alpha_n - a_n \beta_{n-1}}{c_n a_n + b_n}$$

$$x_i = a_{i+1} x_{i+1} + b_{i+1}, i = n-1, \dots, 1$$

1.3. Выводы по разделу

В основном, для интерполяции табулированной функции обычно используют только небольшое число узлов сетки, ближайших к искомому значению аргумента x . Однако за пределами выбранных узлов эта формула стремительно теряет точность. Этим самым она оказывается пригодной лишь в малом промежутке. Для других промежутков требуется составить аналогичную формулу, но по новой группе узлов, т.е. с другими численными коэффициентами.

Аппроксимация данных с помощью полиномов приведена, но при добавлении новых узлов должна быть полностью пересчитана для всех данных.

Многочлены Лагранжа и Ньютона определяют наименьшую степень полиномиальных кривых, которые проходят через данный набор (x_i, y_i) [3]. Однако по мере увеличения порядка полиномов любые небольшие возмущения в координатах приводят к большим выбросам в конечных точках,

что известно в литературе как феномен (явление) Рунге^[6]. Более высокий порядок уменьшает точность, а также увеличивает сложность алгоритма, требуется много времени, память и даже работает не стабильно, наблюдаются сбои ЭВМ.

Для набора из нескольких точек полиномиальная интерполяция с высокой степенью из-за явления Рунге, в этом случае возможность выполнения линейной интерполяции между каждой из точек. Функция кубического сплайна (третьего порядка) рассмотрена, эти сплайны гладко связаны друг с другом в координатах (x_i, y_i) , и, поскольку их непрерывные первая и вторая производные существуют повсюду, общая сгенерированная кривая является гладкой. Кубический сплайн-интерполяция, хотя и строится сразу на большом отрезке, может обеспечить высокую точность лишь за счет использования большого числа коэффициентов.

Таким образом данное решение должно требовать разработку других метод, в следующей главе предложим метод аппроксимации В-сплайнами.

2. АППРОКСИМАЦИИ ДАННЫХ С ПОМОЩЬЮ В-СПЛАЙНОВ

В-сплайн — сплайн-функция, имеющая наименьший носитель для заданной степени, порядка гладкости и разбиения области определения с заданными пользователем узлами. А также он обладает желательной локальностью возможно производить локальные изменения кривой без полного пересчета.

Как правило, аппроксимация кривой В-сплайном вместо интерполяции предпочтительна для плотных и зашумленных данных для создания кривых. В данной главе, аппроксимация кривой В-сплайнами изучена, особенно когда необходимо аппроксимировать плотные и зашумленные точки данных. В этом случае сглаживание и размещения повторенных узлов используются для предварительной обработки данных в существующих приложениях, чтобы облегчить и улучшить производительность аппроксимации кривой.

2.1. Подход к построению В-сплайнов

Для задания В-сплайнов одной переменной необходимо наличие массива узлов некоторой одномерной сетки w_x , считая ее равномерной с шагом $h > 0$:

$$w_x = \{x_i: x_i = x_{i-1} + h, \quad i = 0, 1, \dots, n-1, x_0 = a, x_{n-1} = b\}$$

В-сплайн нулевого порядка определяется согласно формуле:

$$B_i^0(x) = \begin{cases} 1, & x_i \leq x \leq x_{i+1} \\ 0, & \text{otherwise} \end{cases} \quad (2.1)$$

Тогда получаем, что В-сплайны k -го порядка ($k \geq 1$) могут быть определены рекуррентно, согласно по формуле:

$$B_i^k(x) = \frac{(x-x_i)B_i^{k-1}(x)}{x_{i+k}-x_i} + \frac{(x_{i+k+1}-x)B_{i+1}^k(x)}{x_{i+k+1}-x_{i+1}} \quad (2.2)$$

В-сплайн-функции могут быть представлены через последовательность узлов, как однородных или неоднородных. В-сплайновая кривая является однородной, если расстояние между всеми узлами одинаково на реальной прямой. Если кривая однородна, активная часть всех В-сплайнов формирует одинаковую форму на каждом интервале.

Предполагая w_x равномерной сеткой с шагом $h>0$, выведем, в соответствии с (2.2), формулы для В-сплайнов различных порядков.

В-сплайн первого порядка ($k=1$) :

$$B_i^1(x) = \frac{x-x_i}{h} B_i^0(x) + \frac{x_{i+2}-x}{h} B_{i+1}^0(x) \quad (2.3)$$

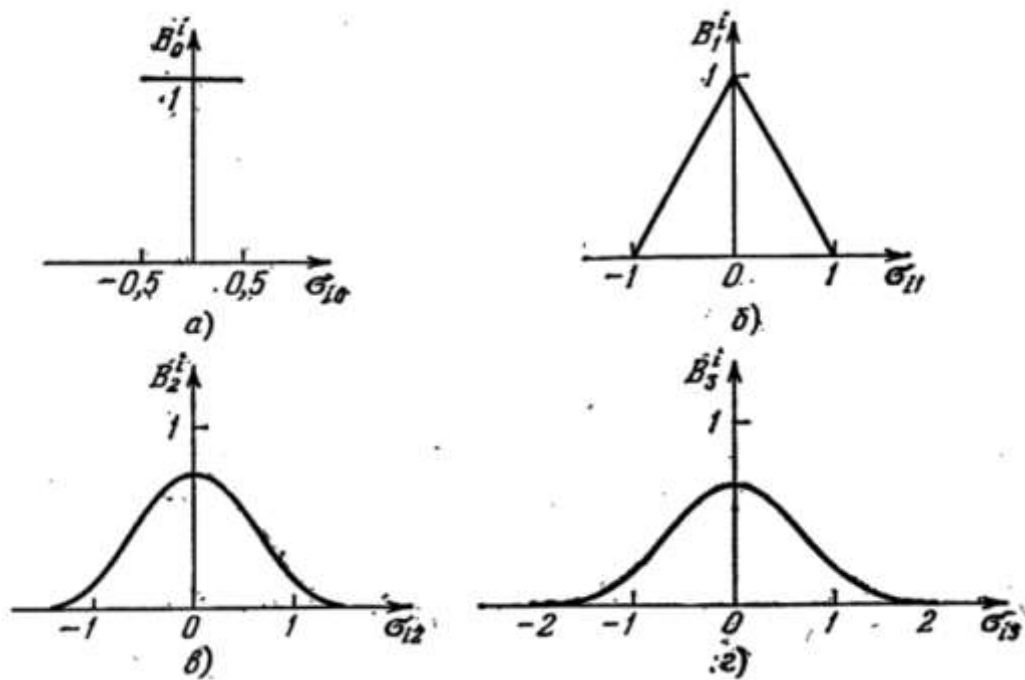
В-сплайн второго порядка ($k=2$) :

$$B_i^2(x) = \frac{x-x_i}{2h} B_i^1(x) + \frac{x_{i+3}-x}{2h} B_{i+1}^1(x) \quad (2.4)$$

В-сплайн третьего порядка ($k=3$) :

$$B_i^3(x) = \frac{x-x_i}{3h} B_i^2(x) + \frac{x_{i+4}-x}{3h} B_{i+1}^2(x) \quad (2.5)$$

Рассмотрим на рис.2 зависимость В-сплайнов от разных порядков:



(а) нулевого порядка. (б) 1-ого порядка.(в) 2-ого порядка. (г) 3-его порядка

Рисунок 2 – Графическое представление В-сплайнов

Кривая кубического В-сплайна может быть определена в форме четырех кусочно-полиномиальной функции третьего порядка путем объединения в пяти узлах. Реализуя формулу (2.5), получаем:

$$B_i^3(x) = \begin{cases} 0 & , x \leq x_i \\ \frac{(x - x_i)^3}{6h^3} & , x \in [x_i, x_{i+1}] \\ \frac{h^3 + 3h^2(x - x_{i+1}) + 3h(x - x_{i+1})^2 - 3(x - x_{i+1})^3}{6h^3} & , x \in [x_{i+1}, x_{i+2}] \\ \frac{h^3 + 3h^2(x_{i+3} - x) + 3h(x_{i+3} - x)^2 - 3(x_{i+3} - x)^3}{6h^3} & , x \in [x_{i+2}, x_{i+3}] \\ \frac{(x_{i+4} - x)^3}{6h^3} & , x \in [x_{i+3}, x_{i+4}] \\ 0 & , x \geq x_{i+4} \end{cases}$$

На рис. 3 представлена иллюстрация кубической функции В-сплайна которая состоит из четырех интервалов, т.е. четырех кубических полиномов для аппроксимации требуется не менее пяти узлов.

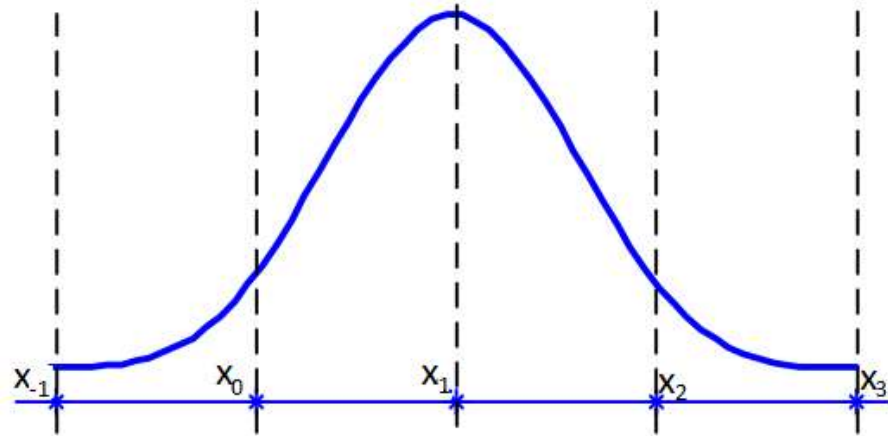


Рисунок 3 – Кривая кубического В-сплайна четырьмя кубическими многочленами, построенная из 5 точек.

2.2. Задача аппроксимации таблично заданных функций

Подстановка задачи

Пусть значения функции $f(x)$ на промежутке $[a, b]$ заданы n узлов одномерной сетки

$$w_x = \{ x_i: x_i = x_{i-1} + h, \quad i = 0, 1, \dots, n-1, x_0 = a, x_{n-1} = b \}$$

Заданные в узлах сетки значения $y_i = f(x_i) + \varepsilon_i$, где ε_i - случайная ошибка измерений $M(\varepsilon_i) = 0, i = \overline{0, n-1}$

Требуется аппроксимировать значения y_i дважды непрерывно дифференцируемой на отрезке $[a, b]$ функцией $S(x, \lambda^*)$, где $\lambda^* \in E_m$, удовлетворяющей условию:

$$\sum_{i=0}^{n-1} [S(x_i, \lambda^*) - y_i]^2 = \min_{\lambda \in E_m} \sum_{i=0}^{n-1} [S(x_i, \lambda) - y_i]^2$$

Формирования расширенной сетки

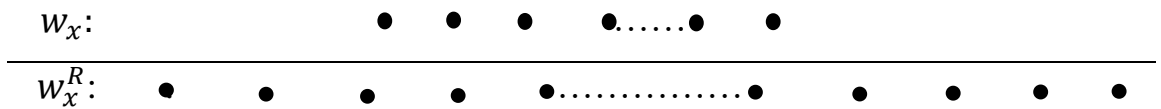
Введем в рассмотрение разреженную сетку w_x^R В-сплайнов третьего порядка:

$$w_x^R = \left\{ \begin{array}{l} x_i^R: x_0^R = a; x_{i+1}^R = x_i^R + h_R, i = -3, -2, -1, 0, 1, \dots, q + 3; \\ x_{q-1}^R < b; x_q^R \geq b \end{array} \right\}$$

Здесь q - количество узлов разреженной сетки на промежутке $[a, b]$;

$h_R = R \cdot h$; $R > 1$ - коэффициент, определяющий большую или меньшую разреженность сетки w_x^R по сравнению с сеткой w_x .

Пример расположения узлов сеток w_x и w_x^R :



Вследствие того, что коэффициент R (h_R) определяет соотношение между узлами и количеством узлов разреженной сетки на промежутке, кубические В-сплайны основаны на различиях узлов, форма кривой функции зависит от значения коэффициента R (интервала h_R), но не зависит от конкретных значений узлов. Следовательно, значение коэффициента R и также интервал h_R можно рассматривать как параметр для управления размером сетки кубического В-сплайна. На рис. 4 представлена иллюстрация кубических В-сплайнов, которые определяют различными значениями интервала h_R от $1/12$ до $1/6$.

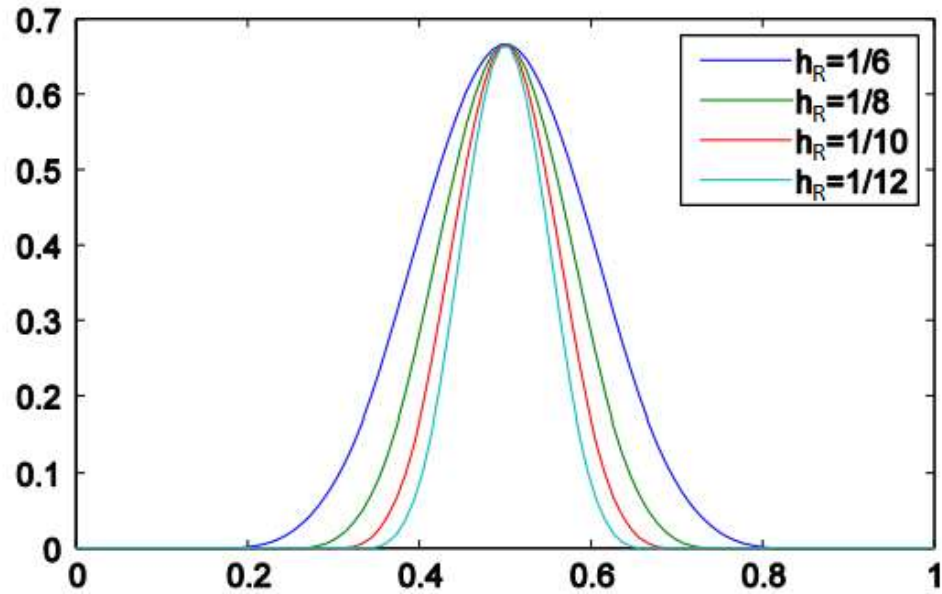


Рисунок 4 – Иллюстрация кубических В-сплайнов определяющих с различными значениями интервала h_R

Узлы $\{x_{-3}^R, x_{-2}^R, x_{-1}^R\}$ и $\{x_q^R, x_{q+1}^R, x_{q+2}^R, x_{q+3}^R\}$ являются дополнительными узлами, которые располагаются в начальной и конечной основной сетке соответственно. Так как каждый кубический В-сплайн отличен от нуля на интервале ограниченном пятью последовательными узлами, узел x_{q-1}^R является последним базовым узлом, а узлы $\{x_q^R, x_{q+1}^R, x_{q+2}^R, x_{q+3}^R\}$ в расчетах фактически не участвуют и носят чисто формальный характер.

В следствии множество базовых узлов определено: $\{x_i^R, i = -3, -2, -1, 0, \dots, q-1\}$, с которого из этих начинается задание каждого из В-сплайнов. Например, базовым узлом для кубического В-сплайна $B_1^3(x)$ является x_1^R . На рис. 3 представлена кривая кубического В-сплайна, построенная из 5 узлов с базовым узлом x_1^R

На основе множества базовых узлов зададим семейство В-сплайнов третьего порядка:

$$B_{-3}^3(x), B_{-2}^3(x), B_{-1}^3(x), B_0^3(x), B_1^3(x), \dots, B_{q-2}^3(x), B_{q-1}^3(x) \quad (2.7)$$

Всего $m = q+3$ В-сплайнов, каждый из них отличен от нуля на интервале ограниченном пятью последовательными узлами расширенной сетки w_x^R :

- Кубический В-сплайн B_{-3}^3 отличен от нуля на интервале (x_{-3}^R, x_1^R) ;

- Кубический В-сплайн B_{-2}^3 отличен от нуля на интервале (x_{-2}^R, x_2^R) ;
- Продолжая аналогичным образом, получаем, что кубический В-сплайн B_{q-1}^3 отличен от нуля на интервале (x_{q-1}^R, x_{q+3}^R)

В результате, сформированное семейство В-сплайнов покрывает весь отрезок $[a, b]$ сдвинутыми на один шаг расширенной сетки копиями кубического В-сплайна. На рис.5 показана иллюстрация расположения В-сплайнов третьего порядка вдоль одномерной сетки.

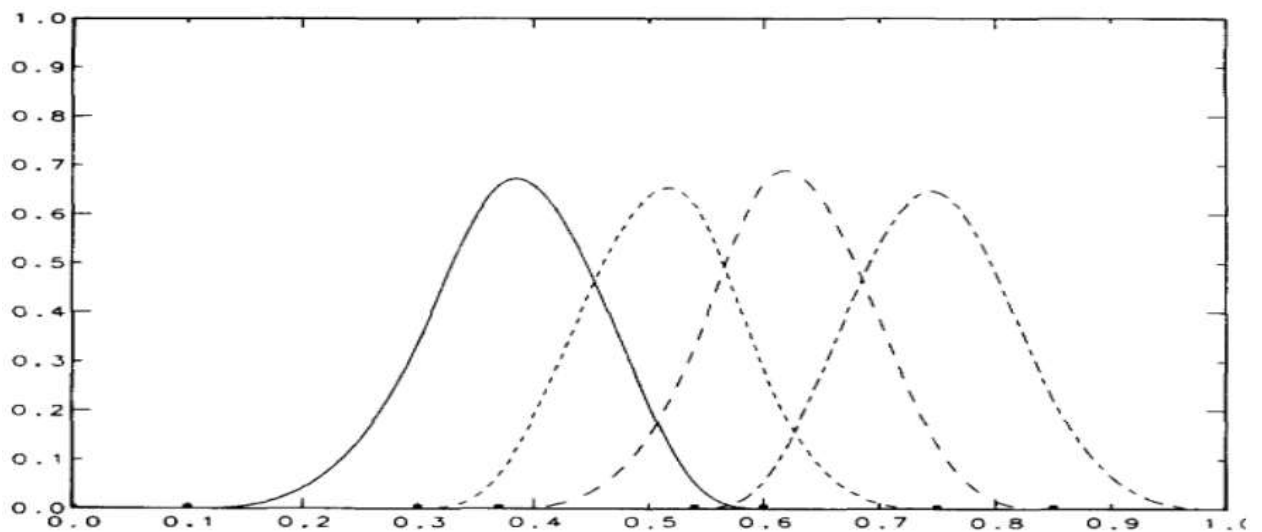


Рисунок 5 – Иллюстрация расположения В-сплайнов третьего порядка
вдоль одномерной сетки

Пример формирования расширенной сетки

Рассмотрим пример формирования расширенной сетки В-сплайна третьего порядка ($k = 3$) для сетки, заданной 10 точками ($n = 10$) за шагом $h = 1$, задавая для расширенной сетки $R=1.35$ и $x_0^R = x_0$.

Заданная сетка: $w_x = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$

Значения $y = \{10, -25, 32, -7, -42, 49, 22, 49, -10, 16\}$

$h=1, R=1.35 \rightarrow h_R = 1.35$

Введем в рассмотрение сетку w_x^R :

$w_x^R = \{-4.05, -2.7, -1.35, 0, 1.35, 2.7, 4.05, 5.4, 6.75, 8.1, 9.45, 10.8, 12.15, 13.5\}$

Таблица 2.1: Расширенная сетка

q	-3	-2	-1	0	1	2	3	4	5	6	7	8	9	10
x_q^R	-4.05	-2.7	-1.35	0	1.35	2.7	4.05	5.4	6.75	8.1	9.45	10.8	12.15	13.5

Увидим, что $x_6^R = 8.1$ является последним базовым узлом, таким образом семейство В-сплайнов, состоящее из $m = 10$ В-сплайнов покрывает весь отрезок $[0, 9]$, получается:

$$B^3(x) = \{B_q^3(x), q = -3, -2, -1, 0, 1 \dots 5, 6\}$$

Таблица 2.2: Семейство В-сплайнов определяется на ограниченных интервалах

В-сплайн определен	Определяется на интервале
$B_{-3}^3(x)$	$[x_{-3}^R, x_1^R] = [-4.05, 1.35]$
$B_{-2}^3(x)$	$[x_{-2}^R, x_2^R] = [-2.7, 2.7]$
$B_{-1}^3(x)$	$[x_{-1}^R, x_3^R] = [-1.35, 4.05]$
$B_0^3(x)$	$[x_0^R, x_4^R] = [0, 5.4]$
$B_1^3(x)$	$[x_1^R, x_5^R] = [1.35, 6.75]$
$B_2^3(x)$	$[x_2^R, x_6^R] = [2.7, 8.1]$
$B_3^3(x)$	$[x_3^R, x_7^R] = [4.05, 9.45]$
$B_4^3(x)$	$[x_4^R, x_8^R] = [5.4, 10.08]$
$B_5^3(x)$	$[x_5^R, x_9^R] = [6.75, 12.15]$
$B_6^3(x)$	$[x_6^R, x_{10}^R] = [8.1, 13.5]$

Из примера мы видим, что в результате, сформированное семейство В-сплайнов (2.7) покрывает весь отрезок $[a, b]$ сдвинутыми на один шаг расширенной сетки w_x^R копиями кубического В-сплайна.

Определение значений В-сплайнов

Дальше определим функцию $S(x, \lambda)$ как линейную комбинацию семейства В-сплайнов (3):

$$S(x, \lambda) = \sum_{i=-3}^{q-1} \lambda_i B_i^3(x) \quad (2.8)$$

Для получения конкретной аппроксимирующей функции $S(x, \lambda^*)$ будем искать решение линейной системы уравнений:

$$S(x_j, \lambda) = y_j, j = \overline{0, n-1} \quad (2.9)$$

Запишем (2.9) в более подробном виде системы линейных алгебраических уравнений (СЛАУ):

$$\begin{cases} \lambda_{-3}B_{-3}^3(x_0) + \lambda_{-2}B_{-2}^3(x_0) + \lambda_{-1}B_{-1}^3(x_0) + \lambda_0B_0^3(x_0) + \dots + \lambda_{q-1}B_{q-1}^3(x_0) = y_0 \\ \lambda_{-3}B_{-3}^3(x_1) + \lambda_{-2}B_{-2}^3(x_1) + \lambda_{-1}B_{-1}^3(x_1) + \lambda_0B_0^3(x_1) + \dots + \lambda_{q-1}B_{q-1}^3(x_1) = y_1 \\ \lambda_{-3}B_{-3}^3(x_2) + \lambda_{-2}B_{-2}^3(x_2) + \lambda_{-1}B_{-1}^3(x_2) + \lambda_0B_0^3(x_2) + \dots + \lambda_{q-1}B_{q-1}^3(x_2) = y_2 \\ \vdots \\ \lambda_{-3}B_{-3}^3(x_{n-2}) + \lambda_{-2}B_{-2}^3(x_{n-2}) + \lambda_{-1}B_{-1}^3(x_{n-2}) + \lambda_0B_0^3(x_{n-2}) + \dots + \lambda_{q-1}B_{q-1}^3(x_{n-2}) = y_{n-2} \\ \lambda_{-3}B_{-3}^3(x_{n-1}) + \lambda_{-2}B_{-2}^3(x_{n-1}) + \lambda_{-1}B_{-1}^3(x_{n-1}) + \lambda_0B_0^3(x_{n-1}) + \dots + \lambda_{q-1}B_{q-1}^3(x_{n-1}) = y_{n-1} \end{cases}$$

Как правило, $q + 3 = m < n$, и система (2.9) является переопределенной СЛАУ. В силу отсутствия точного решения переопределенных систем, на практике принято вместо него отыскивать вектор, наилучшим образом удовлетворяющий всем уравнениям, то есть минимизирующий норму невязки системы в какой-нибудь степени. Наиболее часто минимизируют квадрат отклонений от оцениваемого решения. Для этого применяют так называемый метод наименьших квадратов. Это значит необходимо искать обобщенное решение системы (2.9), доставляющее минимум суммы квадратов невязок:

$$\sum_{i=0}^{n-1} [S(x_i, \lambda^*) - y_i]^2 = \min_{\lambda \in E_m} \sum_{i=0}^{n-1} [S(x_i, \lambda) - y_i]^2 \quad (2.10)$$

Для достижения этой цели записана система (2.9) в матричной форме и использован метод, основанный на первой трансформации Гаусса:

$$A \cdot \lambda = y \quad (2.11)$$

где $A_{n,m}$ - матрица коэффициентов В-сплайнов на каждом значении узла расширенной сетки, т.е. примет вид:

$$A_{(n,m)} = \begin{pmatrix} B_{-3}^3(x_0) & B_{-2}^3(x_0) & B_{-1}^3(x_0) & B_0^3(x_0) & \dots & B_{q-2}^3(x_0) & B_{q-1}^3(x_0) \\ B_{-3}^3(x_1) & B_{-2}^3(x_1) & B_{-1}^3(x_1) & B_0^3(x_1) & \dots & B_{q-2}^3(x_1) & B_{q-1}^3(x_1) \\ B_{-3}^3(x_2) & B_{-2}^3(x_2) & B_{-1}^3(x_2) & B_0^3(x_2) & \dots & B_{q-2}^3(x_2) & B_{q-1}^3(x_2) \\ B_{-3}^3(x_3) & B_{-2}^3(x_3) & B_{-1}^3(x_3) & B_0^3(x_3) & \dots & B_{q-2}^3(x_3) & B_{q-1}^3(x_3) \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ B_{-3}^3(x_{n-2}) & B_{-2}^3(x_{n-2}) & B_{-1}^3(x_{n-2}) & B_0^3(x_{n-2}) & \dots & B_{q-2}^3(x_{n-2}) & B_{q-1}^3(x_{n-2}) \\ B_{-3}^3(x_{n-1}) & B_{-2}^3(x_{n-1}) & B_{-1}^3(x_{n-1}) & B_0^3(x_{n-1}) & \dots & B_{q-2}^3(x_{n-1}) & B_{q-1}^3(x_{n-1}) \end{pmatrix}$$

здесь $m = q+3$, $\lambda \in E_m$, $y \in E_n$

Как известно, в каждом узле исходной сетки отличными от нуля будут четыре – значения четырех кубических многочленов, определенных на интервале ограниченном пятью последовательными узлами (на рис.3). В результате строка матрицы A не обязательно будет не больше четыре отличных от нуля элемента. Другими словами, она содержит большое число нулевых элементов, расположение которых соответственно определяются В-сплайны вне ограниченного интервала. А в системе (2.11) матрица записана в виде разреженной матрицы для экономии компьютерной памяти, сокращения объема вычислений и уменьшения времени.

Таким образом получается, что матрица A имеет вид разреженной матрицы:

$$A_{(n,m)} = \begin{pmatrix} B_{-3}^3(x_0) & B_{-2}^3(x_0) & B_{-1}^3(x_0) & & & & \\ B_{-3}^3(x_1) & B_{-2}^3(x_1) & B_{-1}^3(x_1) & B_0^3(x_1) & & & \\ & B_{-2}^3(x_2) & B_{-1}^3(x_2) & B_0^3(x_2) & B_1^3(x_2) & & \\ & & \ddots & \ddots & \ddots & \ddots & \\ & & & & B_{q-4}^3(x_{n-2}) & B_{q-3}^3(x_{n-2}) & B_{q-2}^3(x_{n-2}) \\ & & & & B_{q-4}^3(x_{n-1}) & B_{q-3}^3(x_{n-1}) & B_{q-2}^3(x_{n-1}) & B_{q-1}^3(x_{n-1}) \end{pmatrix}$$

В следствии полученной матрицы рассмотрим решение СЛАУ (2.11) с помощью первой трансформации Гаусса:

$$\tilde{A} = A^T \cdot A - \text{симметричная и положительно определенная;}$$

$$\tilde{y} = A^T \cdot y$$

В результате преобразования системы (2.11) запишем к виду:

$$\tilde{A} \cdot \lambda = \tilde{y} \quad (2.12)$$

Система (2.12) является системой с квадратной, симметричной и положительно определенной матрицей. Ее решение может быть найдено, в частности, методом квадратных корней^[8] (схема Холецкого). В этом случае матрица преобразуется к виду:

$$\tilde{A} = L \cdot L^T, \quad (2.13)$$

$$\text{где } L - \text{левая треугольная матрица: } L \cdot \beta = \tilde{y} \quad (2.14)$$

$$\text{а } L^T - \text{правая треугольная матрица: } L^T \cdot \lambda = \beta \quad (2.15)$$

λ^* - решение системы (2.15) будет обобщенным решением исходной СЛАУ (2.11), а $S(x, \lambda^*)$, определяемая согласно формуле (2.9) – искомой дважды непрерывно дифференцируемой сплайн-функцией, аппроксимирующей заданные в узлах сетки значения $y_i, i = \overline{0, n-1}$.

Замечание к решению СЛАУ

- Этот алгоритм построения разворачивающихся В-сплайновых может быть использован для решения интерполяционной задачи. При $m = n$ (число неизвестных равно числу уравнений) система (2.9) будет системой с квадратной неособенной матрицей. Такая система имеет единственное решение, а результат ее решения позволит осуществить интерполяцию данных. При этом метод квадратных корней можно и не применять.

- При $m > n$ система (2.9) будет неопределенной СЛАУ (число уравнений меньше числа неизвестных) и будет иметь бесчисленное множество точных решений. В этой ситуации решение системы будет нормальным решением исходной СЛАУ. Для нахождения нормального решения с помощью второй трансформации Гаусса преобразуем эту систему в промежуточную систему с квадратной симметричной матрицей. Тогда система оказывается совместной и имеет единственное решение.

2.3. Выводы по разделу

В данном разделе были изучены основные В-сплайнов; рассмотрена задача аппроксимация таблично заданных функций с использованием кубических В-сплайнов; были приведены методы решения СЛАУ в различных случаях, чтобы помочь процессу нахождения коэффициентов В-сплайна на каждом значении узла сетки.

Это обеспечивает простой линейный алгоритм для построения В-сплайновых кривых третьего порядка, которые определяются на пяти последовательных узлах и аппроксимируют заданную исходную данных. Это означает, что на перемещение любой точки в заданных данных влияют только несколько смежных сегментов этой точки, а не весь набор кривых. Таким образом, кривые В-сплайна обладают многими преимуществами по сравнению с другими кривыми. Этот результат полезно использовать для решения задач аппроксимации, основанных на заданной сети, например, построение графическое отображение заданной сети и оценки погрешности аппроксимации в узловых точках сетки.

В следующей главе создан программный модуль с интерфейсом, позволяющий осуществлять аппроксимацию исходных данных с помощью В-сплайнов.

3. РАЗРАБОТКА ПРОГРАММНОГО МОДУЛЯ

3.1. Краткое описание

Программный модуль разработан на языке Python. Приложение должно предоставлять возможность: ввода исходных данных, выбора метода решаемой задачи (интерполяция или аппроксимация), задания параметров расчета для выбранного метода, отображения полученных результатов и запоминания информации о решенной задаче (в текстовой файл *.json* и в графической форме как рисунок).

3.2. Описание алгоритмов методов

В вычислительной практике, как правило, используется интерполяционный полином с использованием полинома Лагранжа или Ньютона для сетки с большим количеством узлов (выше 5-6, тогда требуется много времени и память ЭВМ). Программный модуль реализуется следующим методами:

- Полиномиальная аппроксимация методом наименьших квадратов;
- Интерполяция кубическими сплайнами;
- Аппроксимация В-сплайнами.

3.2.1. Полиномиальная интерполяция и аппроксимация

Полиномиальная аппроксимация методом наименьших квадратов (МНК)

- На отрезке $[a, b]$ задать набор из n точек (x_i, y_i) ;
- Составить и вычислить СЛУ по формуле (1.1.3) для нахождения коэффициентов $c_i, i = 0, 1, 2, \dots, m$;
- Вычислить (1.1.5), чтобы получить полиномиальную аппроксимацию.

3.2.2. Кубический сплайн-интерполяция

Алгоритм построения интерполяционный кубический сплайн $S(x)$, содержащий следующие шаги:

- На отрезке $[a, b]$ задать одномерную сетку η_x и значения $y_i = f(x_i)$ в узлах сетки $x_i, i = 0, 1, 2, \dots, n$. Задать $x^* \in (a, b)$.

- ii. Положить $a_i = y_j, i = 0, 1, 2, \dots, n$;
- iii. Составить и решить трехдиагональную систему (1.2.4) методом прогонки* для определения значения коэффициентов $c_i, i = 0, 1, 2, \dots, n$;
- iv. Определить значения коэффициентов d_i и $b_i, i = 1, 2, 3, \dots, n$;
- v. Определить значение индекса $0 < k \leq n$ из условия $x^* \in [x_{k-1}, x_k]$;
- vi. Вычислить $S(x^*)$ по формуле (1.2.2). Процесс завершен: $S(x^*)$ – результат интерполяции табличных данных в точку $x^* \in (a, b)$.

3.2.3. Аппроксимация В-сплайнами

Для реализации метода и построении программы созданы алгоритм, содержащий следующие шаги:

- i. На отрезке $[a, b]$ задать одномерную сетку w_x и значения $y_i = f(x_i)$ в узлах сетки $x_i, i = 0, 1, 2, \dots, n$;
- ii. Создать разреженную сетку w_x^R с заданным значением коэффициента R ;
- iii. Вычислить значений В-сплайнов и затем составить матрицу коэффициентов в системе (2.11);
- iv. Решить СЛУ для нахождения решения λ^* , доставляющее минимум суммы квадратов невязок (2.10);
- v. Вычислить $S(x, \lambda^*)$, определяемая согласно формуле (2.9) - результат аппроксимации.

3.3. Описание алгоритмов работы

Анализ рядов данных, практически невозможно представить без их визуализации в виде графического представления рядов. Программа была построена для графического представления этих данных.

На рис.6 изображен общий алгоритм обработки данных:

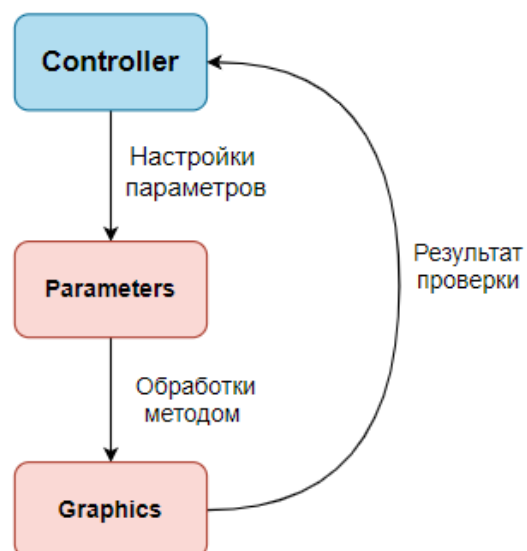


Рисунок 6 – Общий алгоритм обработки данных

3.4. Общая архитектура решения

Выбранное решение подразумевает приложение, взаимодействующее с интерфейсом для обработки данных и получения результатов. На рис.7 представлена обобщенная архитектура решения:

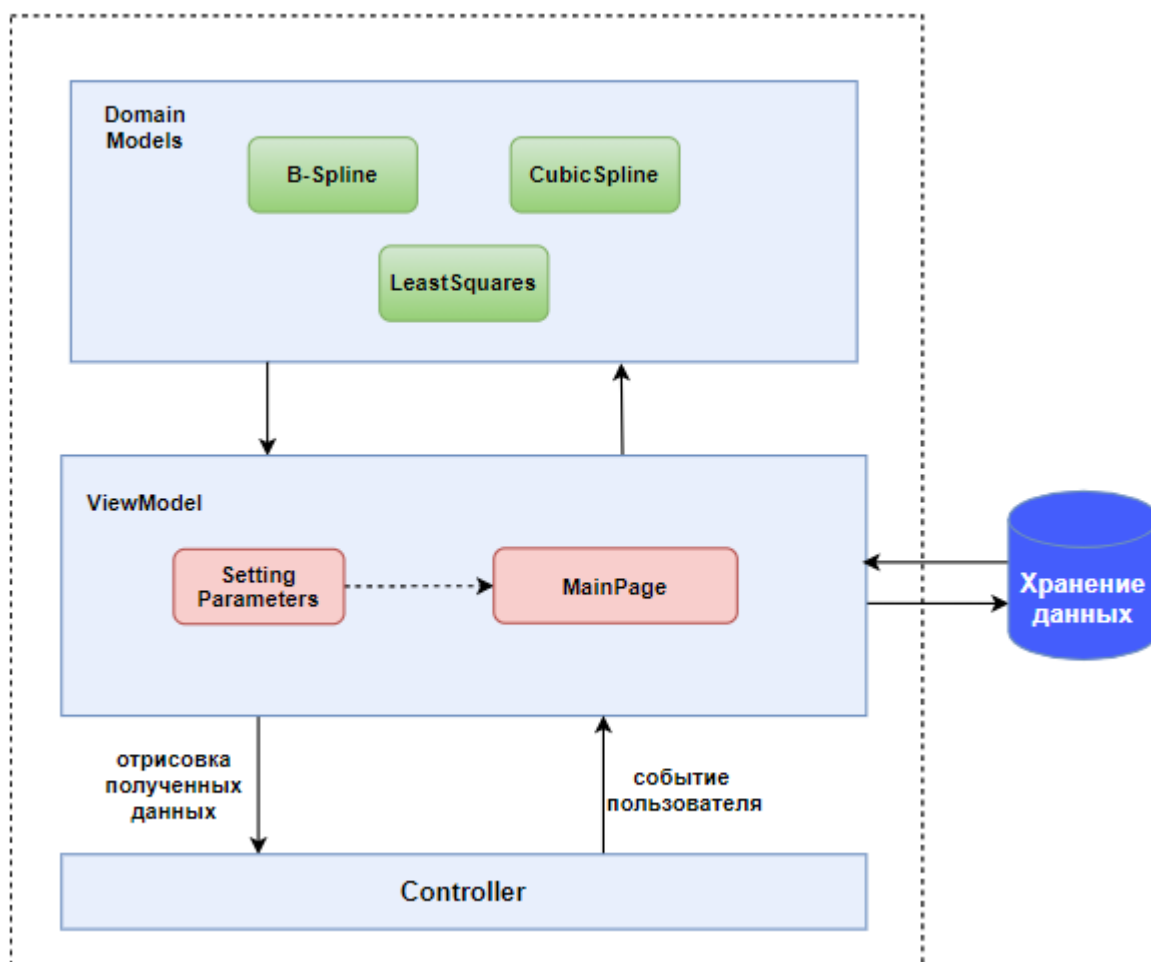


Рисунок 7 – Обобщенная архитектура решения

Более подробно архитектура будет рассмотрена после описания используемых технологий.

3.5. Используемые технологии

В качестве языка разработки выбран основной язык Python. Он является кроссплатформенным (поддерживает такие ОС как Windows, macOS и Linux) и может использоваться для создания приложений и предоставляет несколько стандартных библиотек разработки приложений, рассмотрим некоторые из них:

- Numpy - фундаментальный пакет для научных вычислений с Python^[8].
- Math - модуль обеспечивает доступ к математическим функциям, определенным стандартом C^[9].

- Random - модуль реализует генераторы псевдослучайных чисел для различных распределений^[10].
- Time - модуль предоставляет различные функции, связанные со доступом по времени и преобразования^[11].
- Dataclasses - модуль предоставляет декоратор и функции для автоматического добавления сгенерированных специальных методов, таких как пользовательские классы^[12].
- Matplotlib - комплексная библиотека для создания статических, анимированных и интерактивных визуализаций в Python^[13].
- Tkinter – это стандартный интерфейс Python, предназначенный для работы с библиотекой Tk GUI^[14]. Библиотека Tk содержит компоненты графического интерфейса пользователя (graphical user interface – GUI).

3.6. Интерфейс пользователя приложения программы

В Python есть довольно много GUI фреймворков (graphical user interface), однако только Tkinter встроен в стандартную библиотеку языка. Tkinter имеет несколько преимуществ, поскольку он кроссплатформенный, поэтому один и тот же код можно использовать в Windows, macOS и Linux. Следовательно, приложение сконструировано с использованием популярного графического фреймворка Python Tkinter.

Небольшое приложение создано для представления графического отображения полученных результатов с задания и управления параметров расчета для каждого метода.

На рис.8, 9 представлено окно приложения и макет интерфейса с графом переходов, показанный взаимодействия между другими окнами.

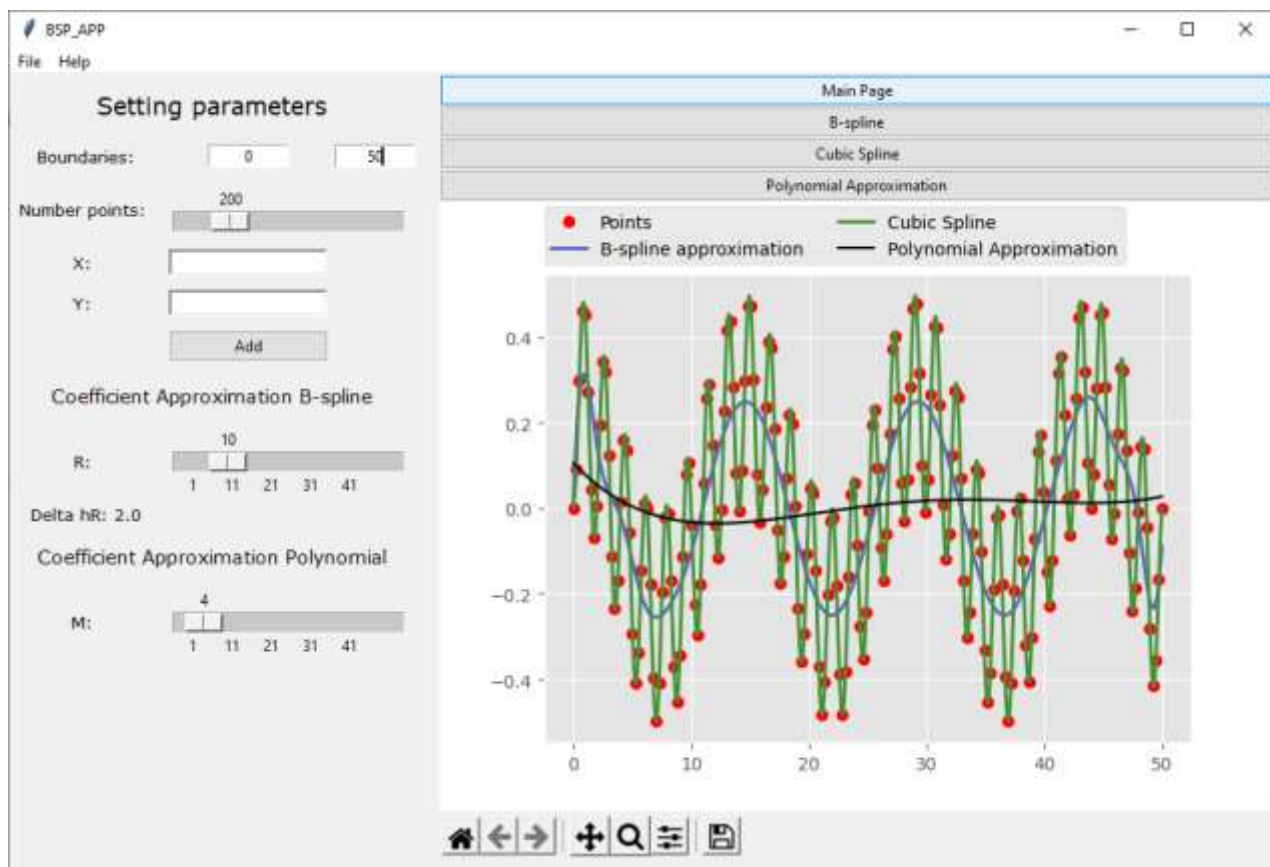


Рисунок 8 – Окно приложения.

3.6.1. Сценарии использования:

Практическое применение решения заключается в построении графического отображения и просмотре полученных результатов анализа, поэтому существуют следующие сценарии использования:

- Действующее лицо: Пользователь.
- Краткое описание: Пользователь запускает программу и выбирает метод обработки, а также задания параметров расчета для выбранного метода.
- Основной сценарий:
 - i. Пользователь запускает программу командой `python app.py`.
 - ii. Пользователь продолжает выбирать методы обработки.
 - iii. При выполнении и просмотре результата:
 - a. Пользователь производит установить параметры в «*Setting parameters*».
 - b. Пользователь сохранит данные: File → Save (Диалоговое окно сохранения файла вызывается для введения названия файла).

с. Пользователь открывает файл: File → Open (Диалоговое окно выбора файлов вызывается).

- Альтернативный сценарий: Пользователь закрывает программу: File → Exit.

3.6.2. Графическая схема

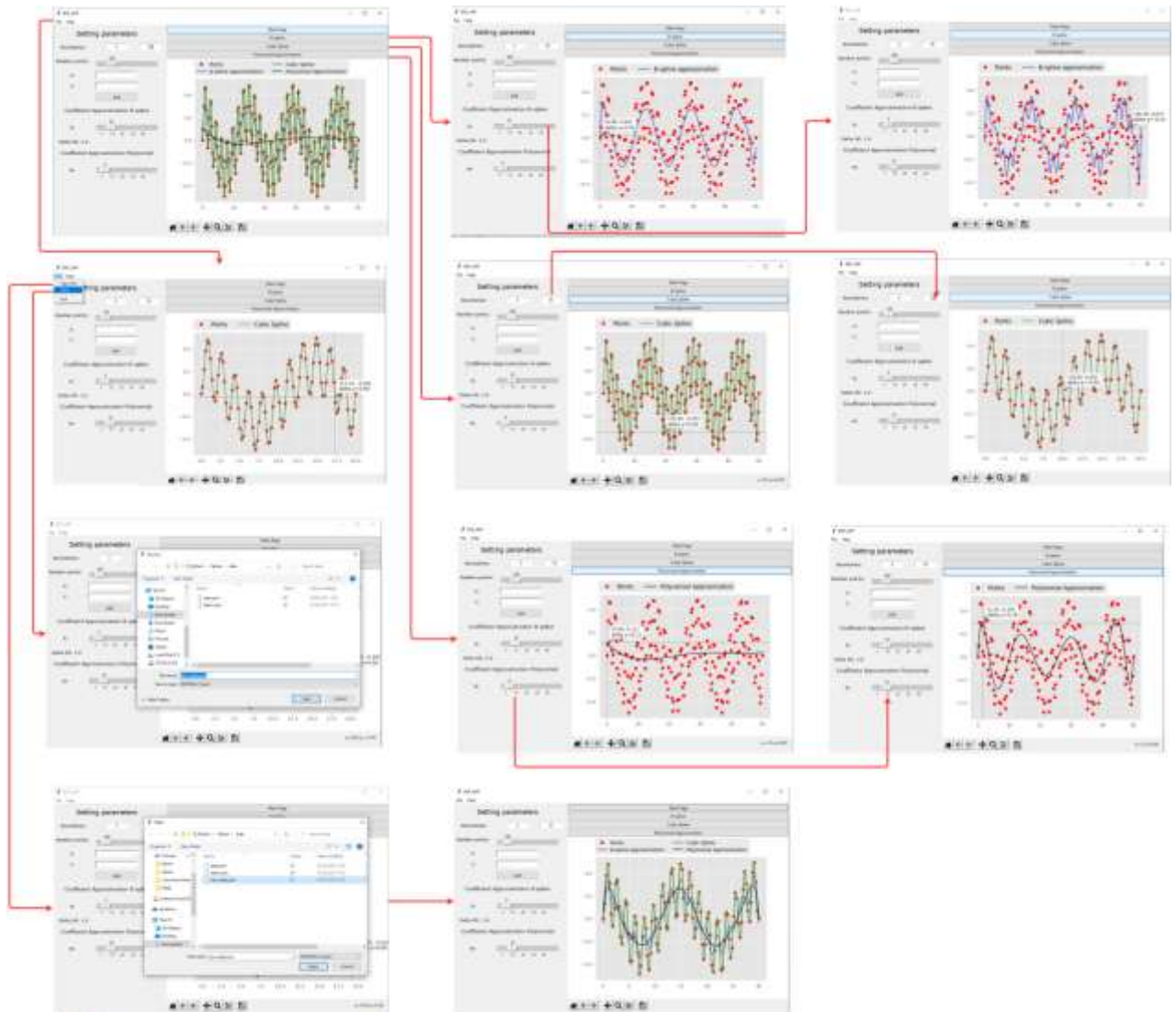


Рисунок 9 – Макет интерфейса с графом переходов (Макет UI)

3.7. Выводы по разделу

Программный модуль реализовано на языке Python в виде интерфейс-приложение. Исходный код решения расположен в открытом репозитории на GitHub^[15].

4. ПРОВЕДЕНИЕ ВЫЧИСЛИТЕЛЬНЫХ ЭКСПЕРИМЕНОВ И АНАЛИЗ ПОЛУЧЕННЫХ РЕЗУЛЬТАТОВ

В этой главе, эксперимент проверяется на персональном компьютере (ПК), на рис.10 показаны основные характеристики ПК. На рис.11 и рис.12 представлены результаты корректного запуска и выполнения программы с различными настройками параметров. А также в процессе эксперимента измеряется время, необходимое для выполнения алгоритма и получения результата каждого метода обработки данных. После эксперимента, в таб. 4.1 и 4.2 приведены результаты средних времени, по этим результатам изучаем эффект метода, а также оптимизацию алгоритма.

4.1. Основные характеристики ПК

- тактовая частота центрального процессора (CPU): Inter® Core™ i5-5200 CPU @ 2.20 ГГц;
- количество ядер центрального процессора: 4 ядра;
- объем оперативной памяти компьютера (ОЗУ) : 6GB RAM
- объем памяти видеокарты: 1 ГБ;
- размер свободного пространства на жестком диске или SSD: 256 ГБ;
- версия Windows: Windows 10 Pro 64-bit.

На рис.10 представлены основные характеристики компьютера

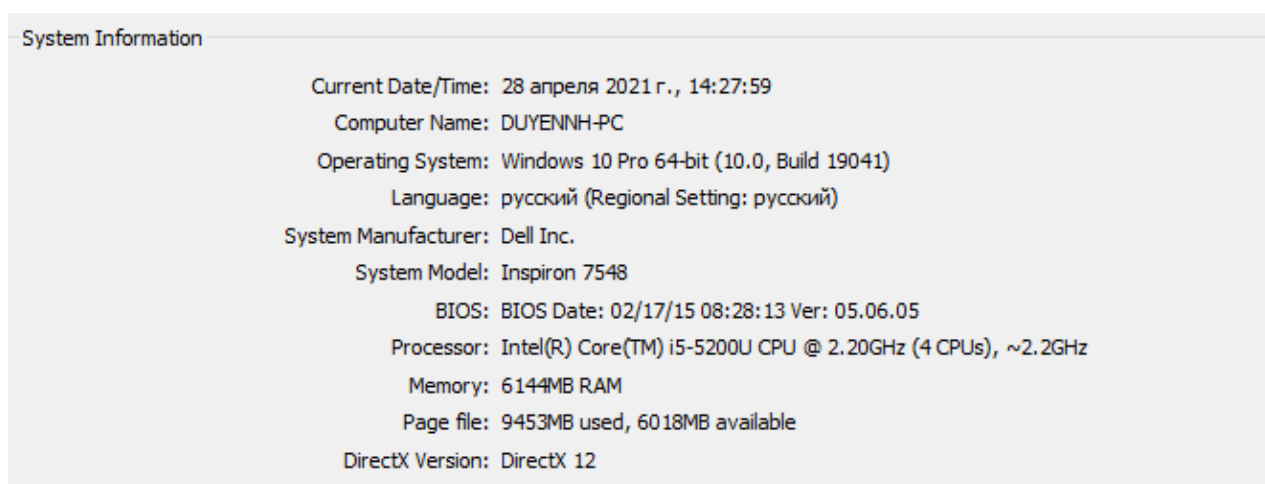


Рисунок 10 – Основные характеристики компьютера

4.2. Тестирование программы

Было проведены вычислительные эксперименты на заданные исходными данными:

- Заданная сетка: $x \in [0, 40]$ заданную 200 точками ($n = 200$)
- Функция значения $f(x) = \frac{1}{2} \sin\left(\frac{\pi}{2}x\right) \cdot \cos\left(2x - \frac{\pi}{2}\right)$
- Коэффициент аппроксимации В-сплайнами: $R = 5$
- Порядка аппроксимирующей полинома: $M = 5$

На рис.11 изображен результат обработки разными методами и на рис.12 проведены результат тот исходными данными после задании параметров в настройки приложения.

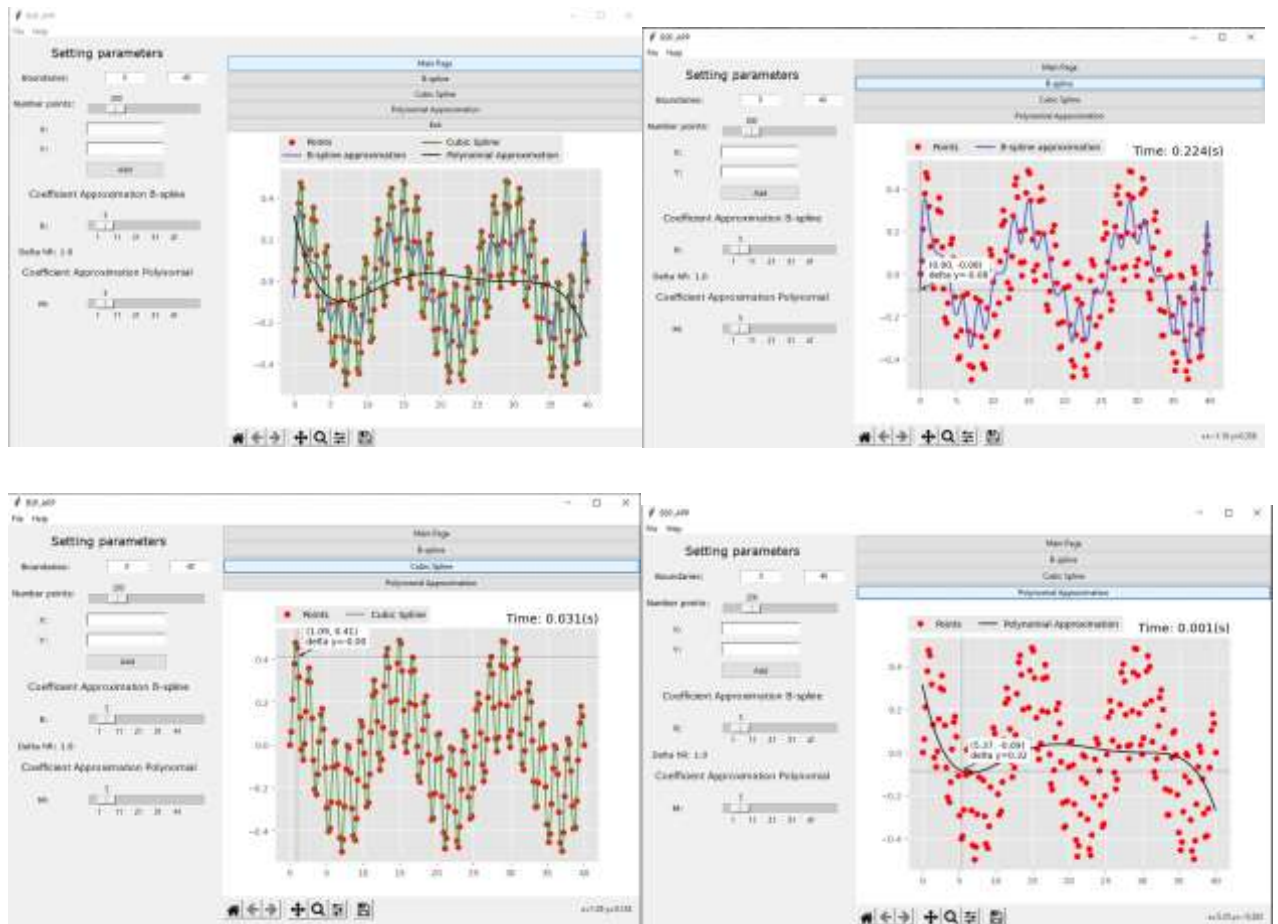


Рисунок 11 – Запуск приложения.

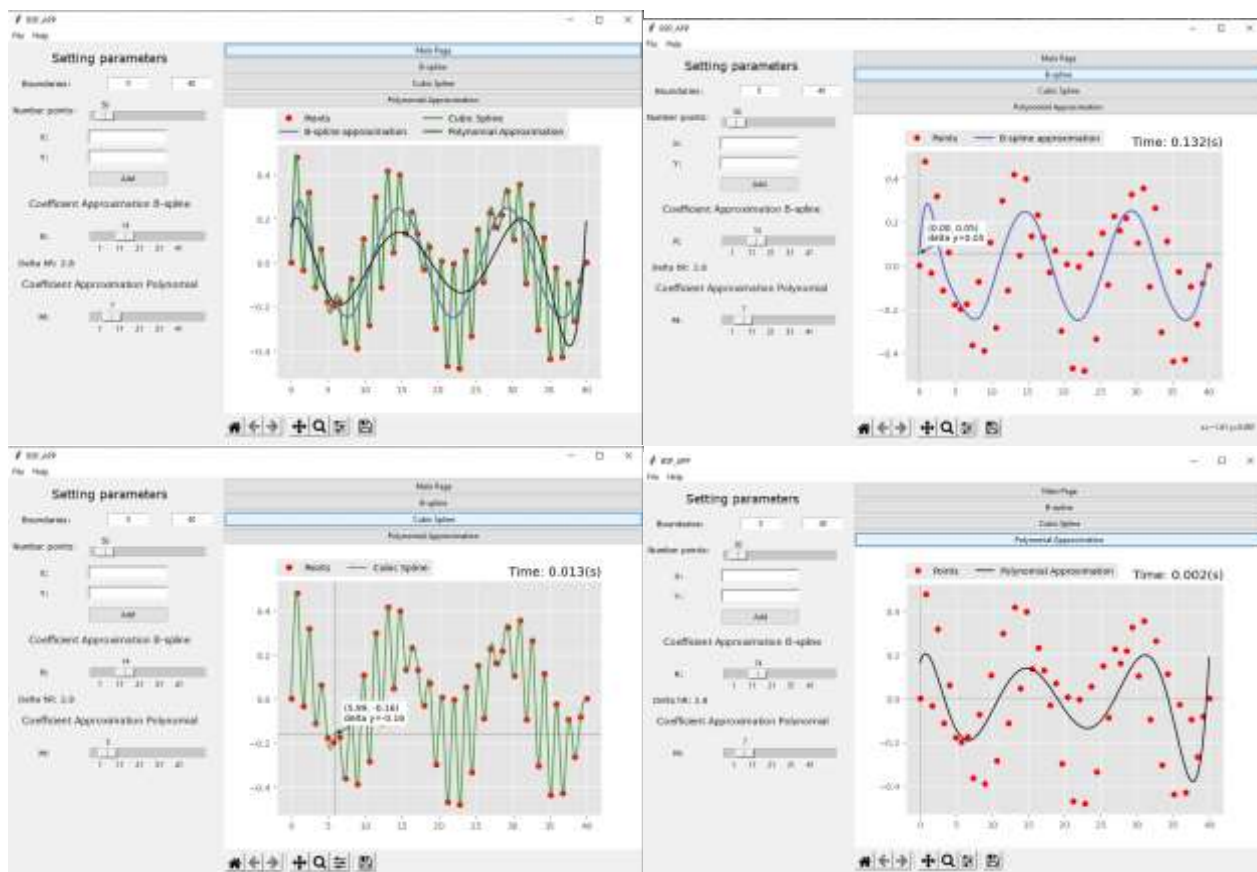


Рисунок 12– Результаты после задания параметров в окно настройки приложения

Рассмотрим таб. 4.1 и 4.2 проведенные результаты измерения сравнение времени, необходимое для выполнения и получения результата в случае использования интерполяционных полиномов (многочлен Лагранжа), интерполяционных сплайнов и В-сплайнов.

Таблица 4.1: Среднее время в случае использования методов интерполяции

Метод интерполяции	Интерполяция полиномами	Кубический сплайн	Интерполяция В-сплайнами
100 узлов	28,8362	0,0112	0,245
	32,9814	0,0216	0,276
	22,8362	0,0129	0,314
<i>Среднее время</i>	<i>28,218 с</i>	<i>0,01205 с</i>	<i>0,2783 с</i>
500 узлов	-	0,036	1,791
	-	0,032	0,975
	-	0,019	1,877
<i>Среднее время</i>	-	<i>0,029 с</i>	<i>1,547 с</i>

1000 узлов	-	0,051	2,735
	-	0,042	2,728
	-	0,062	3,045
<i>Среднее время</i>	-	<i>0,052 с</i>	<i>2,836 с</i>
5000 узлов	-	0,638	54,135
	-	0,357	48,562
	-	0,403	53,721
<i>Среднее время</i>	-	<i>0.466 с</i>	<i>52,139 с</i>

Следовательно, предлагаем следующее сравнение времени, необходимое для выполнения и получения результата в случае использования аппроксимационных В-сплайнов с разными уровнями коэффициента R , определяющий большую или меньшую разреженность узлов сетки.

Таблица 4.2: Среднее время в случае использования аппроксимационных В-сплайнов

Метод аппроксимации	Аппроксимация В-сплайнами		
	$R = 5$	$R = 50$	$R = 100$
100 узлов	0,097	0,034	-
	0,062	0,053	-
	0,098	0,055	-
<i>Среднее время</i>	<i>0,0857 с</i>	<i>0,0473 с</i>	-
500 узлов	0,278	0,068	0,039
	0,335	0,062	0,051
	0,264	0,078	0,058
<i>Среднее время</i>	<i>0,2923 с</i>	<i>0,0693 с</i>	<i>0,0493 с</i>
1000 узлов	0,457	0,133	0,139
	0,651	0,127	0,093
	0,734	0,244	0,142
<i>Среднее время</i>	<i>0,6140 с</i>	<i>0,1680 с</i>	<i>0,1247 с</i>
5000 узлов	3,498	0,609	0,673
	2,781	0,758	0,491
	3,389	0,687	0,519
<i>Среднее время</i>	<i>3,2227 с</i>	<i>0,6847 с</i>	<i>0,5610 с</i>

10000 узлов	5,163	1,689	1,226
	4,509	2,107	1,195
	5,272	2,232	1,344
<i>Среднее время</i>	<i>4,9813 с</i>	<i>2,0093 с</i>	<i>1,255 с</i>

4.3. Анализ полученных результатов

Из таб. 4.1 очевидно, что в случае решения задач большой размерности (рост степени интерполяционного полинома с увеличением числа узлов сетки) использование интерполяционных полиномов со слишком большой степенью потребует недопустимо длительного времени даже на современных ЭВМ. В вычислительной практике, как правило, полиномы выше 5-6-го порядков вообще стараются не использовать из-за большой вероятности резких скачков значений функции на интервалах между узлами сетки.

В этом случае с помощью интерполяционных кубических сплайнов устраняется основной недостаток полиномиальной интерполяции – рост степени интерполяционного полинома при увеличении количества узлов сетки, он используется и оказывается более эффективной, чем попытки подобрать один интерполяционный полином для отрезка $[a, b]$. Полиномы третьего порядка используются в каждом подсегменте отрезка $[a, b]$ и объединяются настолько гладко, насколько это необходимо по всему отрезку. Таким образом, этот метод оказался более эффективным, а также более быстрым временем для получения результатов.

Очевидно, что в таблице 4.1 показано, что самым быстрым способом интерполяции данных является интерполяция кубическими сплайнами, однако при этом осуществляется только интерполяция данных, то есть строится функция, проходящая через заданные фиксированные точки, а не аппроксимации, но с помощью В-сплайнов не только задача интерполяции, но и задача аппроксимации данных.

В таблице 4.2 отказалось сравнение время в случае использования метода аппроксимации с помощью В-сплайнов. Видим, что чем большей разреженность (R роста), тем меньше количество узлов разреженной сетки, а

также уменьшается время, необходимое для выполнения алгоритма. В зависимости от требований аппроксимации задачи коэффициент R может быть соответствующим образом настроен, обеспечивая при этом точность и сохранение характеристик рядов данных.

Из экспериментальных результатов мы обнаруживаем, что кубический В-сплайн обладает свойством локального управления, это означает, что на каждый узел или точку влияет только несколько функций. Это локальное свойство отличается от других методов, таких как глобальный в случае использования кубических сплайнов, а также с помощью полиномов.

5. СПЕЦИАЛЬНЫЕ ВОПРОСЫ ОБЕСПЕЧЕНИЯ БЕЗОПАСНОСТИ ЖИЗНЕДЕЯТЕЛЬНОСТИ

В данной главе рассматриваются аспекты эргономики программного обеспечения, разработанного в данной выпускной квалификационной работе. В детали, рассмотрим эргономику и требования к пользовательскому интерфейсу, а также оценку удобства использования программы. Для взаимодействия с пользователем создан интерфейс для обработки и отображения информации языком программирования Python.

5.1. Оценка аспектов эргономики программного обеспечения

Согласно ИСО 6385:2004, эргономика (изучение факторов, влияющих на человека) (ergonomics study of human factors) – научная дисциплина, изучающая взаимодействие человека с другими элементами системы, предполагающая использование теории, принципов, данных и методов для обеспечения благополучия человека и оптимизации общей производительности системы^{[16][17]}.

Эргономисты применяют теорию, принципы, данные и методы эргономики на протяжении всего процесса проектирования для обеспечения для поддержания здоровья человека и оптимизации общей производительности системы. Этим образом, эргономика программного обеспечения рассматривает вопросы применения эргономики к программно-техническим аспектам интерактивных систем.

Объектом изучения эргономики является система «человек - машина». Предметом изучения эргономики является взаимодействие оператора с техническими средствами (машинами). Общей целью эргономики является обеспечение комфортных условий для эффективной работы человека, эффективное функционирование систем «человек — машина».

Эргономичность взаимодействия человека и техники предусматривает единство свойств как:

- Управляемость;

- Обслуживаемость;
- Усвояемость;
- Обитаемость.

Управляемость, обслуживаемость и усвояемость это такие свойства технологии, которые обеспечивают ее включение в деятельность человека-оператора по управлению, обслуживанию и усвоению техники. Свойство обитаемость отражает условия функционирования техники, при которых сохраняется здоровье людей, использующих ее, поддерживается их нормальная работоспособность и хорошее самочувствие. Одним из видов взаимодействия «человек-техника» является взаимодействие с ПЭВМ в процессе работы с программными приложениями. Под системой взаимодействия «человек - компьютер» подразумевается комплекс, который включает в себя пользователя, автоматизированное рабочее место и интерактивную коммуникационную среду, предназначенную для реализации функциональности программного приложения. Одним из проблем, возникающих перед разработчиками программных приложений, является реализация интерактивного процесса взаимодействия между пользователями и компьютерным оборудованием, обеспечивая при этом помощь в обеспечении достаточного уровня удобства использования и доступности пользователя и программного приложения.

Интерактивная система – сочетание компонентов аппаратного и программного обеспечения, которое получает информацию, вводимую пользователем, и сообщает ему свой ответ, помогая пользователю в работе или выполнении задачи[16].

Пользовательский интерфейс – компоненты интерактивной системы (программное обеспечение и аппаратное обеспечение), которые предоставляют пользователю информацию и возможность управления для выполнения производственных заданий[16].

При конструировании пользовательского интерфейса необходимо рассматривать составляющие информации, получаемой человеком на основе:

- сенсорных ощущений;
- чувств и желаний;
- познаний;
- взаимообмен информацией между людьми.

Диалог: Взаимодействие между пользователем и интерактивной системой, рассматриваемое как последовательность действий пользователя (вводы) и ответных реакций системы (выводы), направленное на достижение цели[16].

Для большинства пользователей знакомство с программным приложением часто формируется непосредственно от работы с пользовательским интерфейсом. Он показывает значительное влияние на психофизиологические состояния человека при взаимодействии программного обеспечения. В связи с этим все большее количество разработчиков программных приложений учитывают вопросы эргономики и применимости пользовательских интерфейсов и удобства работы. В результате пользователи могут быстро и легко выполнять поставленные задачи, не обременяя себя длительным изучением интерфейса ПО. Рассмотрение применимости и эргономики в жизненном цикле программных приложений имеет следующие последствия:

- Повышение скорости работы и удовлетворенности пользователей;
- Уменьшение затрат на эксплуатацию и разработку ПО;
- Сокращение времени и затрат на обучение пользователей;
- Расширения диапазона пользователей, которые смогут использовать ПО;
- Увеличение продаж ПО.

Эргономичный пользовательский интерфейс должен удовлетворять следующим требованиям:

- Обеспечивать ввод, вывод и взаимодействие пользователя наиболее удобным для него способом, не беспокоясь о ходе вычислений;
- Обеспечивать согласование ПО и пользователя, т.е. информация должна быть наглядной и понятной пользователю, объем представленной информации не должен быть избыточным;
- Обеспечивать интуитивно понятное управление и навигацию, организованную в соответствии с задачами пользователей;
- В течение всего времени работы не должно приводить к прерыванию работы ПО;
- Обеспечивать исправление ошибок оператора при вводе исходных данных, либо указывать на ошибки при отсутствии возможности их исправления;
- Обеспечивать обратную связь пользователя с ПО.

Основным документом при анализе пользовательского интерфейса является стандарт ИСО 9241-100:2010.

5.2. Оценка удобства и пригодности при использовании Python

Python – один из самых простых в освоении и популярных мультипарадигмальных языков программирования с динамической строгой типизацией и автоматическим управлением памятью^[18]. Благодаря Python можно собирать, обрабатывать и визуализировать данные. Он известен как интерпретатор, а также используется для написания скриптов^[19] с минимальными кодом.

Его преимущества описаны следующие:

- С стандартными библиотеками Python позволяет проводить математические расчеты, построение графиков, создания пользовательских интерфейсов приложений и т.д.
- За счёт читабельности, простого синтаксиса и грамматики^[20], отсутствия необходимости в компиляции, позволяя концентрироваться на изучении и реализации алгоритмов, концептов и парадигм^[21].

- За счёт простоты язык посилен новичку, так же, за счёт того, что язык реализует почти все известные средства программирования, предоставляет разработчикам выбор в их методологии кодирования.

- Пакеты расширения (более 178 тысяч пакетов^[22]), которые входят в Python, позволяют ориентировать систему под определенную задачу в той или иной предметной области: такие как математическое моделирование, работа с оборудованием, написание веб-приложений или разработка игр и т.д.

5.3. Оценка удобства и пригодности пользовательского интерфейса приложения программы.

Согласно руководству по использованию ИСО 9241 оценка программного обеспечения и его интерфейса начинаем определить цели, которых нужно добиться при использовании программы. На рис. 13 представлен пользовательский интерфейс (UI) приложения.

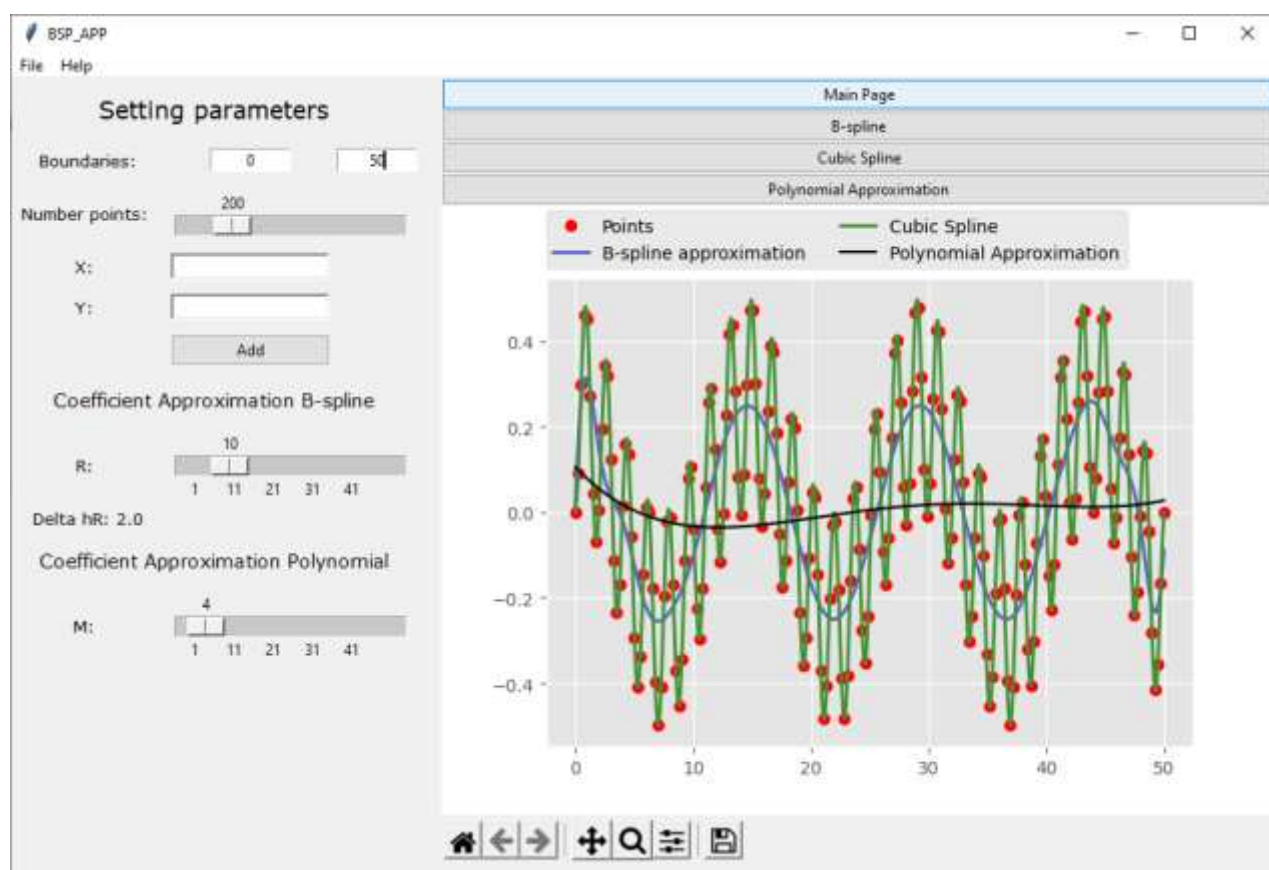


Рисунок 13 – UI приложения

5.3.1. Принципы организации диалога для выполнения задания

При проектировании и оценке интерфейсов следует применять общие эргономические принципы, установленные в ИСО 9241-10. Семь основных принципов диалога, применяемых для проектирования и оценки интерактивных приложений:

а) Пригодность интерфейса для выполнения производственного задания: Используются графические анимации для отображения узлов в сети а также соответствующие линии интерполяции, приближительные в этих узловых точках;

б) Информативность: Информация и графические анимации отображаются в виде графика направо по окну и параметров в настройках “*Setting parameters*” налево по окну;

с) Управляемость: параметры расчета может быть установлены пользователем , а также выбор метода решаемой задачи с помощью кнопок;

д) Соответствие ожиданиям пользователя: Элементы системы управления размещены на экране единообразно и они работают единообразно для всех форм представления информации;

е) Устойчивость к ошибкам: Валидация и верификация ввода данных должны быть выполняться, как можно скорее после настройки параметров. Если параметры настройки недействительны, пользователю отображается сообщение о возможной причине ошибки.

На рис. 14 представлен экран диалогового окна с сообщением об ошибке. Диалог должен обеспечивать достаточной информационной обратной связью, чтобы пользователь добился успешно выполненных действий;

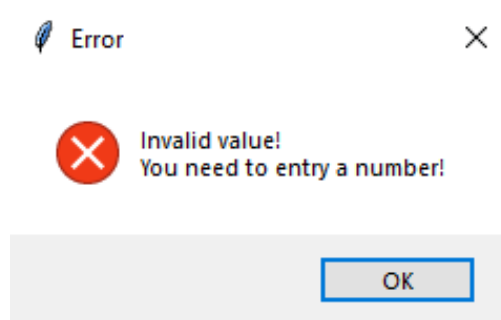


Рисунок 14 – Экран диалоговое окно с сообщением об ошибке.

f) Пригодность к индивидуализации: Пользователи могут внести изменения в настройках информации для того, чтобы удовлетворить свои индивидуальные возможности и потребности. Пользователь должен иметь возможность выбрать способ обработки данных.

g) Пригодность для изучения: Использование комбинации форм представления информации для демонстрации предмета обсуждения с различных точек.

5.3.2. Дополнительные принципы проектирования

В дополнение к общим принципам ИСО 9241-10 в настоящем стандарте установлены дополнительные принципы проектирования пользовательских интерфейсов:

- Пригодность для целей коммуникации: Программное средство должно быть разработано с учетом целей получателя, его производственных заданий и информационных потребностей. Пользователи возможность:
 - удобного ввода и контроля правильности вводимых исходных данных;
 - выбора типа решаемой задачи и используемого метода;
 - задания параметров расчета для выбранного метода;
 - графического отображения полученных результатов;
 - оценки погрешности аппроксимации в узловых точках.
 - запоминания информации о решенной задаче
- Пригодность для восприятия и понимания:

- Обнаруживаемость: Контраст между фоном экрана и набором навигационных кнопок должен быть достаточным, чтобы пользователь легко мог обнаружить эти кнопки;
- Понятность и различимость: В графической анимации некоторого механизма различные части этого механизма отображены различным цветом для облегчения восприятия пользователями;
- Удобочитаемость: текстовый заголовок четко отображается;
- Согласованность: при изменении параметров соответственно меняется и анимация;
- Краткость и понятность: если ошибка произошла при введении данных, то ее представления в диалоговом окне должно быть объяснено пользователю для облегчения исправления..
- Пригодность для изучения:
 - Использование приложение не требует дополнительного обучения, интерпретация недостатков однозначна, поскольку пользователю предоставлено их объяснение;
 - Изображения, отображаемые на экране, предназначены для индивидуального изучения.
 - Помочь пользователю изучать инструкции по использованию системы
 - Кроме этого, обратная связь или пояснения должны помочь пользователю сформировать представление об интерактивной системе: пользователю предоставляются подсказки для заполнения полей настроек.
- Привлекательность: Навигация в приложении должна быть единообразным и интуитивно понятной, чтобы пользователь мог точно определить свою текущую позицию. Эстетическое качество презентации также влияет на привлекательность приложений.

5.4. Выводы по разделу

В ходе проверки интерфейса информационной системы, с точки зрения внешнего вида, программа полностью спроектирована, согласно требованиям и рекомендациям ГОСТа Р9241, были обнаружены недочеты, снижающие эргономичность интерфейса пользователя, но не являющиеся критичными для работы с ней. Важность проектирования грамотного, доступного и легкого в понимании пользовательского интерфейса заключается в том, что он воздействует на чувства, эмоции и настроение пользователя. Интерактивный интерфейс позволяет быстрее воспринимать полученную информацию.

ЗАКЛЮЧЕНИЕ

По итогам данной выпускной квалификационной работы были получены следующие результаты:

- На основании проведенного обзора было проведено исследование возможности аппроксимация данных с помощью В-сплайнов.
- Было разработано приложение, реализующее методы интерполяции или аппроксимации в виде пользовательского интерфейса на языке программирования Python.
- Вычислительные эксперименты проводились на заданных исходной данных, введенных файлом или параметрами расчета в окне приложения. На основании полученных результатов был проведен анализ и сравнение методов, использованных в работе.

Поставленные задачи были решены, цель работы была достигнута.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

-
- [1] Мария «Mifrill» Нефедова, Создатели языков программирования: Они такие разные, но коддинг их объединяет, Хакер № 09/08 (117).
- [2] Самарский А. А., Гулин А. В. Численные методы: Учеб, пособие для вузов,— М.: Наука. Гл. ред. физ-мат. лит., 1989.— 432 с.— ISBN 5-02-013996-3.
- [3] Калиткин Н.Н., Алыпина Е.А. Численный анализ : Ч-671 учебник для студ. учреждений высш. проф. образования М. : Издательский центр «Академия», 2013. — 304 с. — ISBN 978-5-7695-5089-8.
- [4] Сплайн-интерполяция таблично заданной функции // studopedya.ru URL: https://studopedia.ru/7_25916_splayn-interpolyatsiya-tablichno-zadannoy-funktsii.html (дата обращения 01.04.2015).
- [5] Moawwad E.A., El-Mikkawy "On the inverse of a general tridiagonal matrix" // Applied Mathematics and Computation. 150 (3): 669–679. doi:10.1016/S0096-3003(03)00298-4 (дата обращения 17.03.2004).
- [6] Runge C.: 'Über empirische funktionen und die interpolation zwischen äquidistanten ordinaten', *Z. Math. Phys.*, 1901, 46, (224–243), p. 20 [Google Scholar].
- [8] Вержбицкий В. М. Основы численных методов. — М.: Высшая школа, 2009. — 840 с. —(Источники книг: ISBN 9785060061239
- [8] Numpy // [Электронный ресурс]. URL: <https://numpy.org/>, свободный. Яз. англ. (дата обращения 31.01.2021).
- [9] Mathematical functions // [Электронный ресурс]. URL: <https://docs.python.org/3/library/math.html>, свободный. Яз. англ. (дата обращения 08.05.2021).
- [10] Random - Generate pseudo-random numbers // [Электронный ресурс]. URL: <https://docs.python.org/3/library/random.html#module-random>, свободный. Яз. англ. (дата обращения 08.05.2021).

^[11] Time — Time access and conversions // [Электронный ресурс]. URL: <https://docs.python.org/3/library/time.html>, свободный. Яз. англ. (дата обращения 08.05.2021).

^[12] Data Classes // [Электронный ресурс]. URL: <https://docs.python.org/3/library/dataclasses.html>, свободный. Яз. англ. (дата обращения 08.05.2021).

^[13] Matplotlib: Visualization with Python // [Электронный ресурс]. URL: <https://matplotlib.org/>, свободный. Яз. англ. (дата обращения 08.05.2021).

^[14] Python interface to Tcl/Tk // [Электронный ресурс]. URL: <https://docs.python.org/3/library/tkinter.html>, свободный. Яз. англ. (дата обращения 08.05.2021).

^[15] Исходный код решения. // [Электронный ресурс]. URL: <https://github.com/Duyen38/BSPAproximation>, свободный. Яз. англ. (дата обращения 10.05.2021).

^[16] ГОСТ Р ИСО 6385-2007 Применение эргономических принципов при проектировании производственных систем Введ. 01.06.2018. – М.: Изд-во стандартов, 2007 г.

^[17] ГОСТ Р 55241.1-2012/ISO/TR 9241-100:2010 Эргономика взаимодействия человек-система. Часть 100. Введение в стандарты, относящиеся к эргономике программных средств. Введ. 01.12.2013. – М.: Изд-во стандартов, 2010 г.

^[18] Why is Python a dynamic language and also a strongly typed language - Python Wiki (англ.). wiki.python.org (24 February 2012). Дата обращения: 14.03.2021. Архивировано 14.03.2021 г.

^[19] Yogesh Rana. Python: Simple though an Important Programming language (англ.) // International Research Journal of Engineering and Technology (IRJET). — 2019. — 2 February (vol. 06, iss. 2). — P. 1856—1858. — ISSN 2395-0056. Архивировано 11.02.2021 г.

^[20] Mark Lutz. A Python Q&A Session (англ.). Learning Python, 3rd Edition [Book]. O'Reilly Media, Inc. (2007). Дата обращения: 11.02.2021. Архивировано 08.12.2021 г.

^[21] Python Introduction | (англ.). Python Education. Google Developers (20 August 2018). Дата обращения: 21.02.2021. Архивировано 4.12.2020 года.

^[22] Ethan Bommarito, Michael James Bommarito. An Empirical Analysis of the Python Package Index (PyPI) (англ.) // Social Science Research Network. — Rochester, NY: Social Science Research Network, 25.06.2019.