Министерство науки и высшего образования Российской Федерации
федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Московский государственный технический университет имени
Н.Э. Баумана (национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Робототехники и комплексной автоматизации»
КАФЕДРА «Системы автоматизированного проектирования (РК-6)»

ОТЧЕТ О ВЫПОЛНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ

по дисциплине «Вычислительная математика»

Студент:	Турунов Дмитрий Николаевич
Группа:	РК6-53Б
Тип задания:	лабораторная работа
Тема:	Интерполяция параметрическими кубическими сплайнами и автоматическое дифференцирование

Студент ____________ Турунов Д.Н.
подпись, дата Фамилия, И.О.

Преподаватель ____________ ____________
подпись, дата Фамилия, И.О.

Москва, 2023

Содержание

Интерполяция параметрическими кубическими сплайнами и автоматическое дифференцирование

Задание

Требуется (базовая часть):

1. Используя заранее подготовленный скрипт 1, выбрать произвольную область множества Мандельброта и построить фрагмент его границы (контура), сформировав файл `contours.txt`. Использование неуникального файла `contours.txt` считается списыванием, равно как и использование чужого кода.

Файл `contours.txt` содержит упорядоченную последовательность точек на плоскости $P = \{(x_i, y_i)\}_{i=1}^{N}$, принадлежащих выбранному фрагменту границы фрактала с. Сопоставляя каждой паре координат естественную координату t, предполагать, что $x_i = x(t_i), y_i = y(t_i)$. Выбранный контур должен содержать по меньшей мере 100 точек (100 строк в файле `contours.txt`).

2. Разработать код для загрузки и визуализации множества точек P из файла `contours.txt`.

3. Задать разреженное множество интерполяционных узлов $\hat{P} = \{(x_j, y_j)\}_{j=1}^{\hat{N}}$, $\hat{N} = \lfloor N/M \rfloor, j = M \times i, \hat{P} \subset P$. Положить $M = 10$.

4. По каждому измерению найти коэффициенты естественного параметрического кубического сплайна a_{jk} и b_{jk}, путём решения соответствующих разрешающих СЛАУ, в результате должен получится сплайн вида:

$$
\begin{aligned}
\tilde{x}(t) &= \sum_{j=1}^{\hat{N}-1} I_j(t)\left(a_{j0} + a_{j1}(t - t_j) + a_{j2}(t - t_j)^2 + a_{j3}(t - t_j)^3\right),\\
\tilde{y}(t) &= \sum_{j=1}^{\hat{N}-1} I_j(t)\left(b_{j0} + b_{j1}(t - t_j) + b_{j2}(t - t_j)^2 + b_{j3}(t - t_j)^3\right),\\
I_j(t) &= \begin{cases} 1, & t \in [t_j, t_{j+1}) \\ 0, & \text{в противном случае} \end{cases}
\end{aligned}
$$

где $I_j(t)$ - индикаторная функция принадлежности интервалу.

5. Вычислить расстояния $\rho\left[(\tilde{x}(t_i), \tilde{y}(t_i)), (x(t_i), y(t_i))\right]$ и представить вывод (среднее и стандартное отклонение) в отчёте.

6. Отобразить в отчёте полученный сплайн используя $t \in [0, t_N]$ с частым шагом $h = 0.1$ совместно с исходным множеством точек P, а также узловыми точками $\hat{P}$. С чем связана наблюдаемая ошибка интерполяции? Как её можно уменьшить? Вывод следует привести в отчёте.

7. В результате выполнения базовой части задания, помимо прочих, должна быть разработана функция `lab1_base(filename_in:str, factor:int, filename_out:str)`, где `filename_in` - входной файл `contours.txt`, `factor` - значение параметра M, `filename_out` - имя файла результата (как правило `coeffs.txt`), содержащего ко-

эффициенты a_{jk} и b_{jk} в виде матрицы размером $\hat{N}-1$ строк на 8 столбцов. Функция `lab1_base` должна реализовывать базовую часть задания.

Требуется (продвинутая часть):

8. Используя концепцию дуальных чисел $v = a + \varepsilon b, \varepsilon^2 = 0$, и перегрузку операторов сложения и умножения в Python, необходимо реализовать класс AutoDiffNum, для автоматического вычисления производной некоторой функции.

9. Реализовать функцию автоматического расчёта первой производной кубического сплайна $G(t) = \frac{d}{dt}(\tilde{x}(t), \tilde{y}(t))$.

10. Реализовать функцию построения нормали $R(t_j)$ к заданному вектору $G(t_j)$.

11. Построить векторы $G(t_j)$ и $R(t_j)$ в соответствующих точках сплайна, выбрав наглядную частоту прореживания (не менее 5 точек на контур) и масштаб.

12. Опциональное задание. Для каждой пропущенной точки (x_i, y_i) исходного контура найти (численно) ближайшую на соответствующем участке сплайна. Фактически нужно решить оптимизационную задачу:

$$t^* = \underset{t \in [0, t_N]}{\operatorname{argmin}} \sqrt{(\tilde{x}(t) - x_i)^2 + (\tilde{y}(t) - y_i)^2}.$$

К примеру, используя простейший метод деления отрезка пополам (дихотомии) до 10 итераций. В отчёте необходимо отобразить на графике соответствующие точки, выбрав наглядную частоту прореживания и привести среднюю оценку погрешности. Сравнить результаты с вычисленными ранее значениями $\rho\left[\left(\tilde{x}(t_i), \tilde{y}(t_i)\right), \left(x(t_i), y(t_i)\right)\right]$. Все полученные в работе изображения, включаемые в отчёт, должны быть сохранены в векторном формате (PDF или EPS и размещены рядом с исходным кодом разработанной программы).

Аббревиатуры

СЛАУ система линейных алгебраических уравнений. 6

Цель выполнения лабораторной работы

Цель выполнения лабораторной работы: освоить основные концепции и математические принципы, лежащие в основе интерполяции кубическими сплайнами, а также понять механизмы автоматического численного дифференцирования.

1 Базовая часть

Загрузка и Визуализация Исходных Данных

Загрузка исходных точек фрактала Мандельброта произведена из файла `contours.txt`. В файле `contours.txt` включена упорядоченная последовательность точек на плоскости, каждая из которых представлена парой координат (x_i, y_i). Для визуализации этих данных на Рисунке 1 применяется метод `scatter` из библиотеки `matplotlib`.

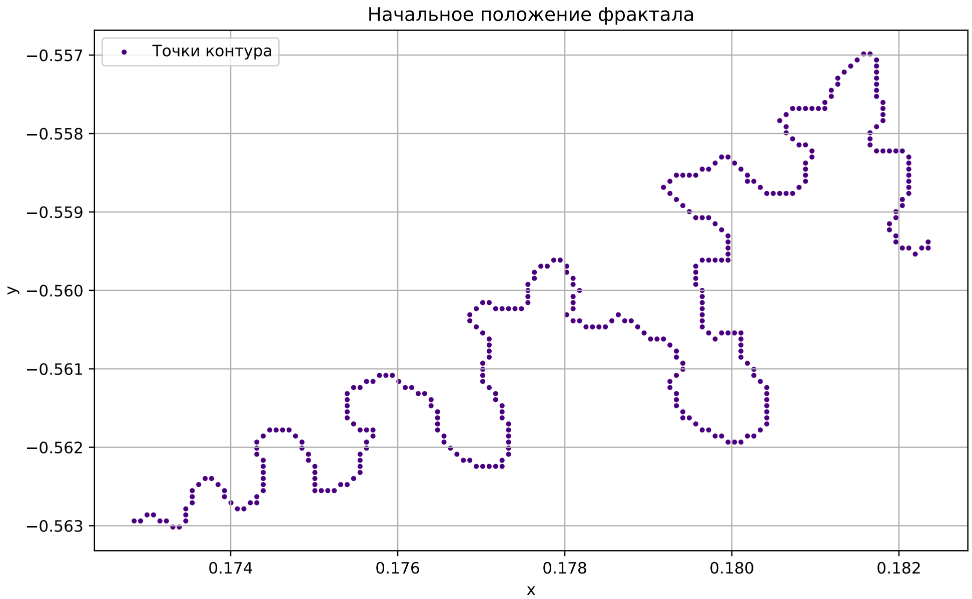

Рис. 1. Визуализация исходных точек фрактала Мандельброта.

Разреженное Множество Интерполяционных Узлов

На Рисунке 2 иллюстрированы исходные точки P и их прореженный вариант $\hat{P}$, полученный с коэффициентом $M = 10$. В $\hat{P}$ каждая десятая точка из P служит узлом для интерполяции.

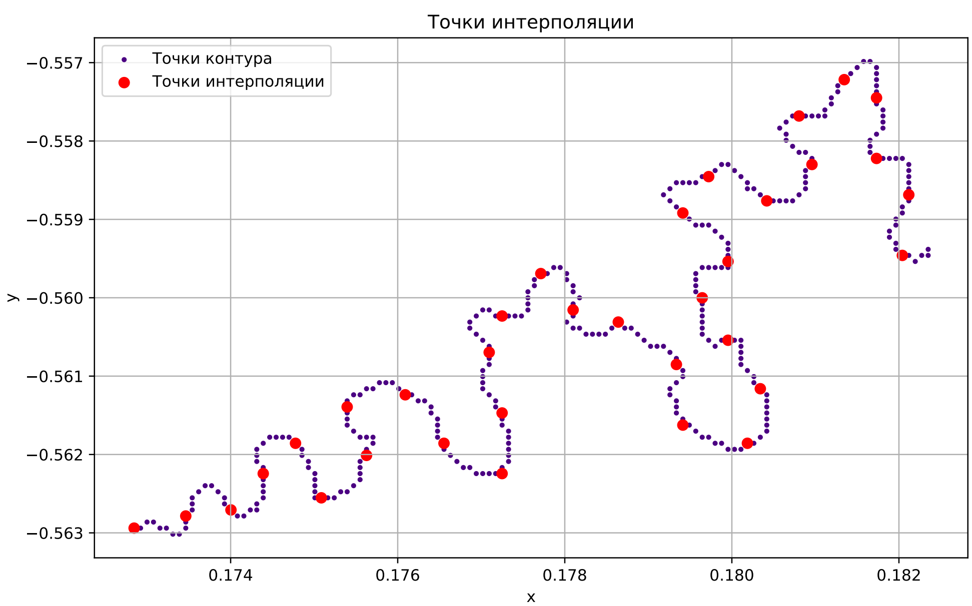

Рис. 2. Исходные точки и интерполяционные узлы.

Вычисление Коэффициентов Кубического Сплайна

Для интерполяции использовались кубические сплайны, параметризованные вдоль каждой из координат. Коэффициенты a_{jk} и b_{jk} сплайнов были найдены путём решения СЛАУ:

$$\begin{bmatrix} 1 & 0 & \dots & \dots & \dots & 0 \\ h_1 & 2\left(h_2+h_1\right) & h_2 & 0 & \dots & 0 \\ 0 & h_2 & 2\left(h_3+h_2\right) & h_3 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & h_{n-2} & 2\left(h_{n-2}+h_{n-1}\right) & h_{n-1} \\ 0 & \dots & \dots & \dots & \dots & 1 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ \vdots \\ c_{n-1} \\ c_n \end{bmatrix} =$$

$$= \begin{bmatrix} 0 \\ \frac{3}{h_2}\left(a_3-a_2\right)-\frac{3}{h_1}\left(a_2-a_1\right) \\ \frac{3}{h_3}\left(a_4-a_3\right)-\frac{3}{h_2}\left(a_3-a_2\right) \\ \vdots \\ \frac{3}{h_{n-1}}\left(a_n-a_{n-1}\right)-\frac{3}{h_{n-2}}\left(a_{n-1}-a_{n-2}\right) \\ 0 \end{bmatrix}$$

Полученные сплайны имеют вид:

$$\tilde{x}(t)=\sum_{j=1}^{\hat{N}-1} I_j(t)\left(a_{j0}+a_{j1}\left(t-t_j\right)+a_{j2}\left(t-t_j\right)^2+a_{j3}\left(t-t_j\right)^3\right),$$

$$\tilde{y}(t)=\sum_{j=1}^{\hat{N}-1} I_j(t)\left(b_{j0}+b_{j1}\left(t-t_j\right)+b_{j2}\left(t-t_j\right)^2+b_{j3}\left(t-t_j\right)^3\right),$$

где $I_j(t)$ - индикаторная функция.

В коде для данного задания представлены три основных функции:

1. ‘gauss(A, B)‘: С использованием метода Гаусса с частичным выбором главного элемента решается система линейных уравнений $Ax = B$. Возвращаемый результат — вектор x.

2. ‘coefficients(t, y)‘: На основе заданных массивов времени t и значений y вычисляются коэффициенты a, b, c, и d кубического сплайна через решение системы линейных уравнений.

3. ‘interpolate(t, t_j, coeffs)‘: Интерполяция значения функции в точке t осуществляется с использованием предварительно вычисленных коэффициентов кубического сплайна.

На графике (Рис. 3) представлены исходные точки, точки интерполяции и кубический сплайн. Процесс описывается следующим образом:

$$\begin{aligned} \text{1. Исходные данные получены:}\quad & x, y \\ \text{2. Вычислены коэффициенты сплайна:}\quad & a_j, b_j, c_j, d_j \\ \text{3. Выполнена интерполяция:}\quad & x(t), y(t) \\ \text{4. Произведена визуализация:}\quad & \text{График} \end{aligned}$$

С помощью описанного подхода данные эффективно интерполируются, а результаты визуализируются.

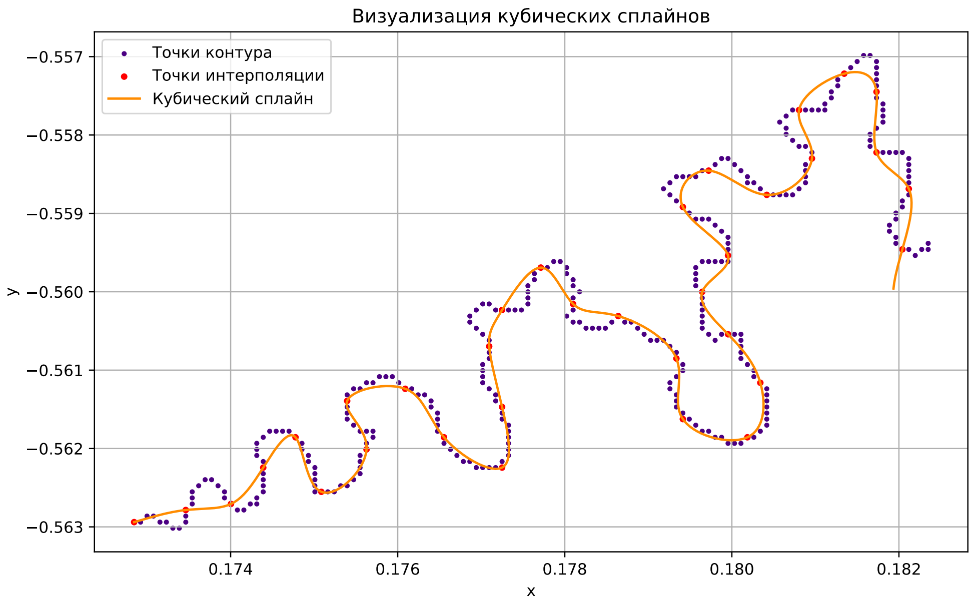

Рис. 3. Кубические сплайны и исходные точки.

Оценка Погрешности Интерполяции

Для оценки качества интерполяции были вычислены расстояния между исходными точками и их интерполяционными значениями. Среднее значение расстояния составило $\rho_{\text{mean}} \approx 1 * 10^{-4}$ и стандартное отклонение $\rho_{\text{std}} \approx 8.6 * 10^{-5}$.

$$\begin{aligned} \text{Евклидово расстояние}: & \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \\ \text{Среднее расстояние}: & \rho_{\text{mean}} \\ \text{Стандартное отклонение}: & \rho_{\text{std}} \end{aligned}$$

Ошибки и их причины

1. Выбор узлов: Ошибка может быть связана с распределением узлов интерполяции. Если узлы распределены неравномерно, это может влиять на точность интерполяции.

2. Численные ошибки: Ошибки округления в вычислениях также могут накапливаться, особенно при больших N.

3. Качество исходных данных: Шум в исходных данных может привести к менее точной интерполяции.

Способы уменьшения ошибки

1. Увеличение количества узлов: Больше узлов может улучшить точность, но это может увеличить вычислительную сложность.

2. Использование адаптивных методов: Адаптивный выбор узлов или параметров сплайна может улучшить результат.

3. Сглаживание данных: Применение методов сглаживания может уменьшить шум в исходных данных, улучшая таким образом качество интерполяции.

4. Повышение точности вычислений: Использование численных методов с повышенной точностью может также снизить ошибки округления.

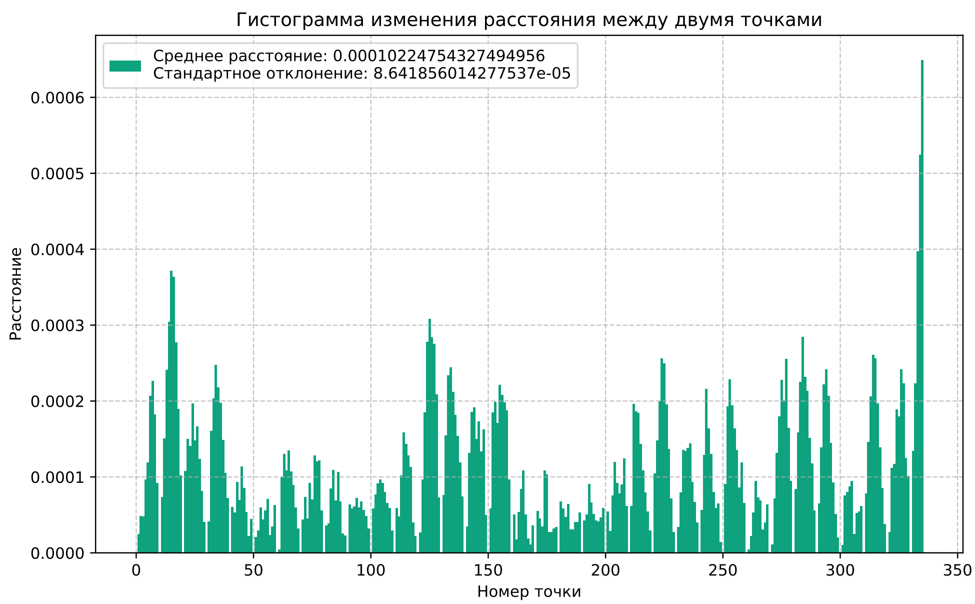

Рис. 4. Расстояния между исходными точками и кубическими сплайнами с учетом точек за пределами интервала интерполяции.

Как иллюстрировано на Рис. 4, более высокая ошибка наблюдается для последнего кубического сплайна. Отсутствие конечной точки интерполяции и 'естественные' краевые условия объясняют более высокую ошибку последнего кубического сплайна. Если рассматривать только контур, жестко ограниченный множеством $\hat{P}$, как показано на Рисунке 5, то общая ошибка ρ характеризуется следующими статистическими параметрами:

$$\rho_{\text{mean}} \approx 9.8 \times 10^{-5}$$

$$\rho_{\text{std}} \approx 7.6 \times 10^{-5}$$

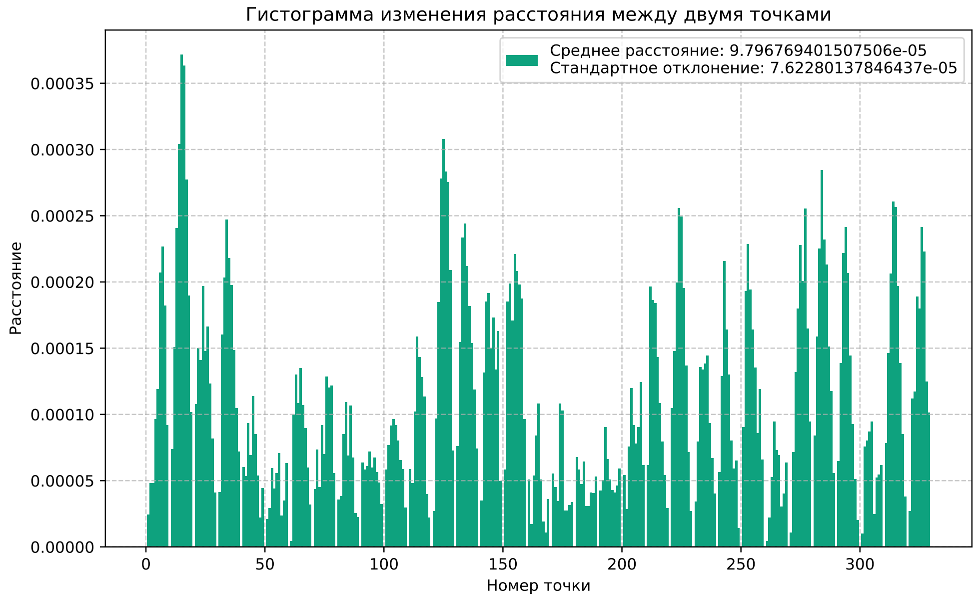

Рис. 5. Расстояния между исходными точками и кубическими сплайнами без учета точек за пределами интервала интерполяции.

Реализация Основной Функции $lab1\_base$

В ходе работы была проведена интерполяция контурных точек множества Мандельброта с использованием кубических сплайнов. Были вычислены коэффициенты для каждого из сплайнов и проведен анализ ошибок. Результаты сохранены в виде коэффициентов в файле `coeffs.txt`.

2 Продвинутая часть

Реализация Класса для Автоматического Дифференцирования

В контексте данного задания представлен класс ‘AutoDiffNum‘, предназначенный для автоматического дифференцирования. Дополнительно реализованы функции для интерполяции и визуализации кубических сплайнов с векторами касательных и нормалей.

Описание класса

1. При инициализации объекта задаются его действительная и ’дуальная’ части.

2. Арифметические операторы (сложение, вычитание, умножение, деление, возведение в степень) перегружены для выполнения соответствующих операций, причем учитывается ’дуальная’ часть объекта.

3. Строковое представление объекта возвращается методом __repr__(self).

Автоматическое Вычисление Производной Кубического Сплайна

При выполнении данного задания были разработаны два статических метода класса ‘AutoDiffNum‘, которые служат для интерполяции и вычисления производной кубического сплайна.

Методы

1. В методе ‘I_j(t, t_j, j)‘ определяется, принадлежит ли значение t интервалу $[t_j, t_{j+1})$. Возвращается булевское значение, указывающее на принадлежность или непринадлежность.

2. В методе ‘interpolate_with_derivative(t, t_j, coeffs)‘ выполняется вычисление значения кубического сплайна и его производной в точке t с использованием дуальных чисел. Объект класса ‘AutoDiffNum‘ с действительной и дуальной частями используется как входной параметр t.

Описание работы

1. Принадлежность значений t к определенному интервалу устанавливается с помощью вложенной функции ‘I_j(t_real, t_j, j)‘.

2. С использованием коэффициентов a, b, c, d кубического сплайна значение функции в точке t вычисляется по формуле:

$$S(t) = a + b\Delta t + c\Delta t^2 + d\Delta t^3$$

где $\Delta t = t - t_j$.

Поскольку t является объектом класса ‘AutoDiffNum‘, информация о производной функции в данной точке также сохраняется.

Построение Нормалей к Вектору $G(t)$

Функция ‘calculate_normals(x, y)‘ представляет собой простой метод для вычисления нормалей к вектору $G(t) = (x(t), y(t))$ в двумерном пространстве. Этот метод основан на свойствах перпендикулярности векторов в двухмерном евклидовом пространстве.

Описание работы

1. В функцию ‘calculate_normals(x, y)‘ передаются координаты x и y вектора $G(t)$.

2. Для определения нормали к вектору $G(t)$, координаты x и y переставляются местами, после чего координата x инвертируется. Это действие соответствует повороту исходного вектора на 90° против часовой стрелки.

$$\text{Нормаль} = (-y, x)$$

В результате выполнения функции возвращается кортеж, содержащий координаты x и y нормали к вектору $G(t)$.

Визуализация Векторов и Нормалей

В данном фрагменте кода выполняется визуализация векторов и нормалей, ассоциированных с кубическим сплайном.

Описание работы

1. На графике отображаются исходные точки (‘x, y‘), точки разреженного множества, и кубический сплайн.

2. Используя автоматическое дифференцирование (‘AutoDiffNum‘), вычисляются касательные и нормали к кубическому сплайну в выбранных точках.

3. Векторы касательных и нормалей нормализуются и масштабируются.

4. Векторы и нормали добавляются на график с помощью функции ‘quiver‘.

1. Автоматическое дифференцирование:	$f(x) = f_{\text{real}}(x) + \varepsilon f_{\text{dual}}(x)$
2. Коэффициенты сплайна:	a_j, b_j, c_j, d_j
3. Интерполяция и производные:	$x(t), y(t), x'(t), y'(t)$
4. Векторы и нормали:	Тангенциальные и нормальные векторы к сплайну
5. Визуализация:	График

Этот подход позволяет не только интерполировать данные с помощью кубических сплайнов, но и вычислять касательные и нормали в определенных точках.

Конечный результат представляет собой график, на котором отображены исходные точки, кубический сплайн, а также векторы касательных и нормалей к сплайну. Данная визуализация (Рис. 6) может быть полезна для анализа геометрических свойств кривой, включая направление и ориентацию в различных точках.

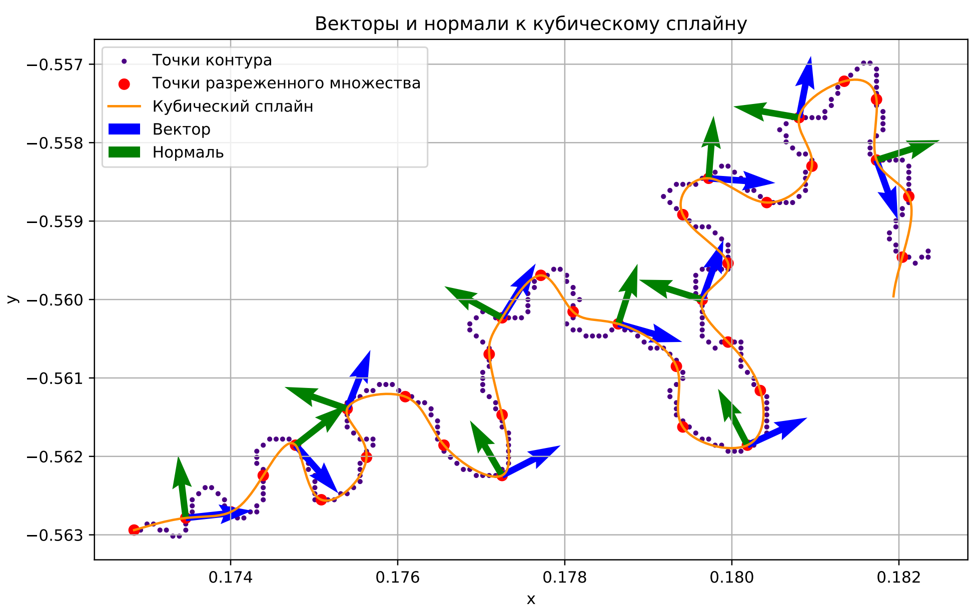

Рис. 6. Векторы и нормали к кубическому сплайну.

Численный Поиск Ближайших Точек на Сплайне

В данном кодовом фрагменте реализован метод для нахождения ближайших точек на кубическом сплайне относительно заданного набора точек. В основе алгоритма лежит метод золотого сечения для минимизации функций.

Описание работы

1. Функция для расчета расстояния (‘distance_to_point‘): Данная функция принимает в себя параметр t кубического сплайна и координаты заданной точки (xi, yi). Она вычисляет расстояние от этой точки до точки на сплайне, параметризованной t.

2. Метод для минимизации функции (‘minimize‘): Этот метод реализует принцип золотого сечения. Он принимает в себя функцию для минимизации, начальный и конечный интервалы $[a, b]$, заданную точность и максимальное число итераций. Цель — найти минимум функции на заданном интервале.

3. Поиск ближайших точек и расстояний: Для каждой точки из заданного массива (xi, yi) находится ближайшая точка на кубическом сплайне. Это реализуется путем минимизации функции расстояния с использованием метода минимизации.

4. Визуализация: В конечном итоге результаты визуализируются на графике, где показаны исходные точки, кубический сплайн и найденные ближайшие точки.

$$\begin{aligned}
\text{Евклидово расстояние} &: \sqrt{(x_t - xi)^2 + (y_t - yi)^2} \\
\text{Минимизируемая функция} &: \text{distance_to_point}(t, xi, yi, a_jk, b_jk, indices) \\
\text{Ближайшая точка} &: (x_{\text{closest}}, y_{\text{closest}}) \\
\text{Средняя ошибка} &: \text{Среднее значение всех расстояний до ближайших точек}
\end{aligned}$$

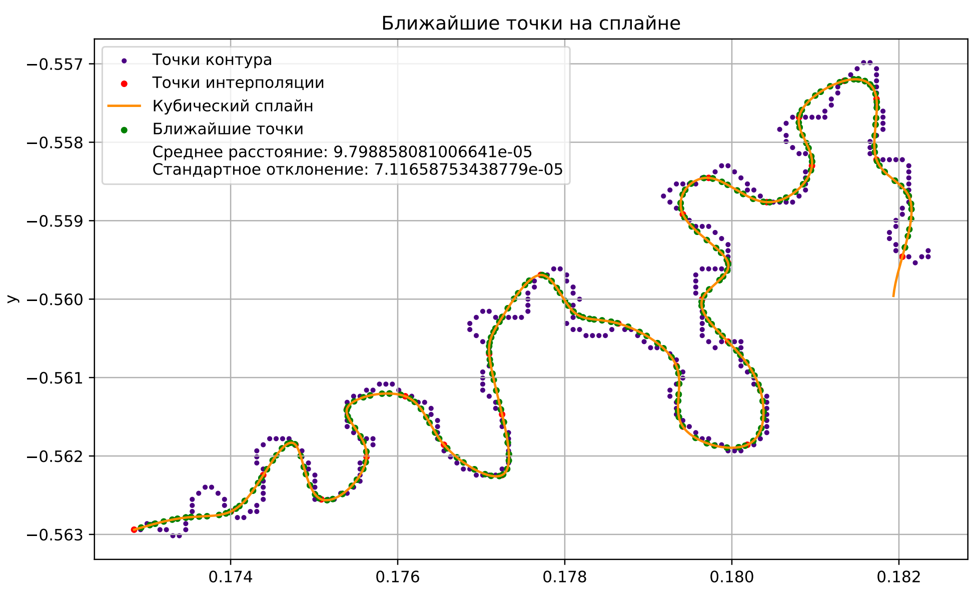

Рис. 7. Ближайшие точки на сплайне.

Изначальные значения:
$\rho_{\text{std}} = 7.6 \times 10^{-5}, \quad \rho_{\text{mean}} = 9.8 \times 10^{-5}$
Новые значения (метод золотого сечения):
$\rho^*_{\text{std}} = 7.1 \times 10^{-5}, \quad \rho^*_{\text{mean}} = 9.8 \times 10^{-5}$

Сравнивая, можно выразить различия:

$$\left|\rho_{\text{std}} - \rho^*_{\text{std}}\right|, \quad \left|\rho_{\text{mean}} - \rho^*_{\text{mean}}\right|$$

Для стандартного отклонения:

$$\left|7.6 \times 10^{-5} - 7.1 \times 10^{-5}\right| = 0.5 \times 10^{-5}$$

Для среднего расстояния:

$$\left|9.8 \times 10^{-5} - 9.8 \times 10^{-5}\right| = 0$$

Это говорит о том, что среднее расстояние осталось неизменным, а стандартное отклонение незначительно уменьшилось. Это может свидетельствовать о том, что метод золотого сечения хорошо справился с задачей, и погрешность интерполяции остается на низком уровне.

3 Заключение

В ходе выполнения данной лабораторной работы были изучены и применены разнообразные методы и технологии, включая:

- Работа с фракталами: Было проведено исследование множества Мандельброта и визуализация его границы.
- Файловый ввод-вывод в Python: Проведена загрузка и сохранение данных в текстовые файлы.
- Использование библиотеки matplotlib для визуализации данных: Графическое представление множества точек и сплайнов.
- Интерполяция кубическими сплайнами: Был реализован алгоритм для нахождения коэффициентов сплайнов.
- Решение систем линейных алгебраических уравнений с использованием NumPy.
- Работа с дуальными числами и перегрузка операторов в Python: Реализована автоматическая дифференциация.
- Методы численной оптимизации: Применен метод золотого сечения для нахождения ближайших точек.

Можно заключить, что лабораторная работа предоставила ценный опыт в области численных методов, компьютерной графики и программирования на Python. Она позволила не только практически применить теоретические знания, но и разобраться в сложных математических концепциях, таких как фракталы и сплайны. Кроме того, были изучены методы оценки точности и ошибок интерполяции, что является важным аспектом в любом численном методе.

Список использованных источников

1. Першин А.Ю. Лекции по курсу «Вычислительная математика». Москва, 2018-2021. С. 140. URL: https://archrk6.bmstu.ru/index.php/f/810046.

2. Соколов, А.П. Инструкция по выполнению лабораторных работ (общая). Москва: Соколов, А.П., 2018-2021. С. 9. URL: https://archrk6.bmstu.ru. (облачный сервис кафедры РК6).

3. Соколов, А.П. Инструкция по выполнению заданий к семинарским занятиям (общая). Москва: Соколов, А.П., 2018-2022. С. 7. URL: https://archrk6.bmstu.ru. (облачный сервис кафедры РК6).

4. Першин А.Ю. Сборник задач семинарских занятий по курсу «Вычислительная математика»: Учебное пособие. / Под редакцией Соколова А.П. [Электронный ресурс]. Москва, 2018-2021. С. 20. URL: https://archrk6.bmstu.ru. (облачный сервис кафедры РК6).

5. Першин А.Ю., Соколов А.П. Сборник постановок задач на лабораторные работы по курсу «Вычислительная математика»: Учебное пособие. [Электронный ресурс]. Москва, 2021. С. 54. URL: https://archrk6.bmstu.ru. (облачный сервис кафедры РК6).

6. Marcus D. R. Klarqvist, *Introduction to Automatic Differentiation Part 2*, MDRK, 2023, https://mdrk.io/introduction-to-automatic-differentiation-part2/.

Выходные данные

Турунов Д.Н.Отчет о выполнении лабораторной работы по дисциплине «Вычислительная математика». [Электронный ресурс] — Москва: 2023. — 14 с. URL: https://sa2systems.ru:88 (система контроля версий кафедры РК6)

Постановка: CC BY *аспирант кафедры РК-6, Гудым А.В.*

Решение и вёрстка: CC BY *студент группы РК6-53Б, Турунов Д.Н.*

2023, осенний семестр