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:	Рогалева Юлия Александровна
Группа:	РК6-52Б
Тип задания:	лабораторная работа
Тема:	Интерполяция кубическими сплайнами и автоматическое дифференцирование

Студент ________ подпись, дата — Рогалева Ю.А. (Фамилия, И.О.)

Преподаватель ________ подпись, дата — ________ Фамилия, И.О.

Москва, 2023

Содержание

Интерполяция кубическими сплайнами и автоматическое дифференцирование

Задание

Множество Мандельброта (фрактал) — удивительное явление широко известное за пределами соответствующей области математики благодаря бесконечному многообразию форм и ярким визуализациям.

Определение 1

Точки фрактала $\mathbf{c} \subset \mathbb{C}$ в пространстве комплексных чисел определяются рекуррентной формулой, задающей последовательность, ограниченную только для точек, принадлежащих фракталу:

$$\begin{aligned} \forall c \in \mathbf{c}, \quad \exists Z \in \mathbb{R} : |z_i| < Z, i \in \mathbb{N}, \\ z_i = z_{i-1}^2 + c, \\ z_0 = 0 \end{aligned} \tag{1}$$

1 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, ровно 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 $\mathbf{c}$. 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: $\hat{P} = \{(x_j, y_j)\}_{j=1}^{\hat{N}}$, где $\hat{N} = \lfloor N/M \rfloor$, $j = M \times i$, $\hat{P} \subset P$. 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 получиться сплайн вида:

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

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

5. Вычислить расстояния $\rho[(\tilde{x}(t_i), \tilde{y}(t_i)), (x(t_i), y(t_i))]$ 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 коэффициенты 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 $v = a + \epsilon b, \epsilon^2 = 0$, 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:

$$t^* = \underset{t \in [0, t_N]}{\operatorname{argmin}} \sqrt{\left(\tilde{x}(t) - x_i\right)^2 + \left(\tilde{y}(t) - y_i\right)^2}. \tag{3}$$

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

Цель выполнения лабораторной работы

Цель выполнения лабораторной работы: исследование интерполяции данных с использованием кубических сплайнов, реализация продвинутых методов автоматического дифференцирования и оптимизации.

1 Формирование файла contours.txt

Используя заранее подготовленный скрипт, произведено выделение специфической области множества Мандельброта. На основе выбранной области сгенерирован и зафиксирован в файле contours.txt фрагмент границы данного множества.

2 Визуализация множества точек P

Файл contours.txt содержит упорядоченную последовательность точек на плоскости $P = \{(x_i, y_i)\}_{i=1}^{N}$, представляющую собой контур множества Мандельброта. Разработан код, который осуществляет загрузку и визуализацию данных из файла contours.txt. Полученное множество точек продемонстрировано на рисунке 1.

3 Задание разреженного множества интерполяционных узлов

Множество интерполяционных узлов задаётся следующим образом: $\hat{P} = \{(x_j, y_j)\}_{j=1}^{\hat{N}}$, где $\hat{N} = \lfloor N/M \rfloor$, $j = M \times i$, $\hat{P} \subset P$, $M = 10$ по условию, N — общее количество узлов в исходном множестве P [1]. Для формирования разреженного множества интерполяционных узлов разработана функция select_points (листинг 1). Полученное разреженное множество представлено на рисунке 2.

Листинг 1. Функця для задания разреженного мнжества точек

```python
def select_points(x_points, y_points, m): # m = 10
    selected_x_points = x_points[::m] # выбор каждой m-й точки
    selected_y_points = y_points[::m]
    if selected_x_points[-1] != x_points[-1] or selected_y_points[-1] != y_points[-1]:
    # добавление последнего узла в случае необходимости
        selected_x_points = np.append(selected_x_points, x_points[-1])
        selected_y_points = np.append(selected_y_points, y_points[-1])
    return selected_x_points, selected_y_points
```

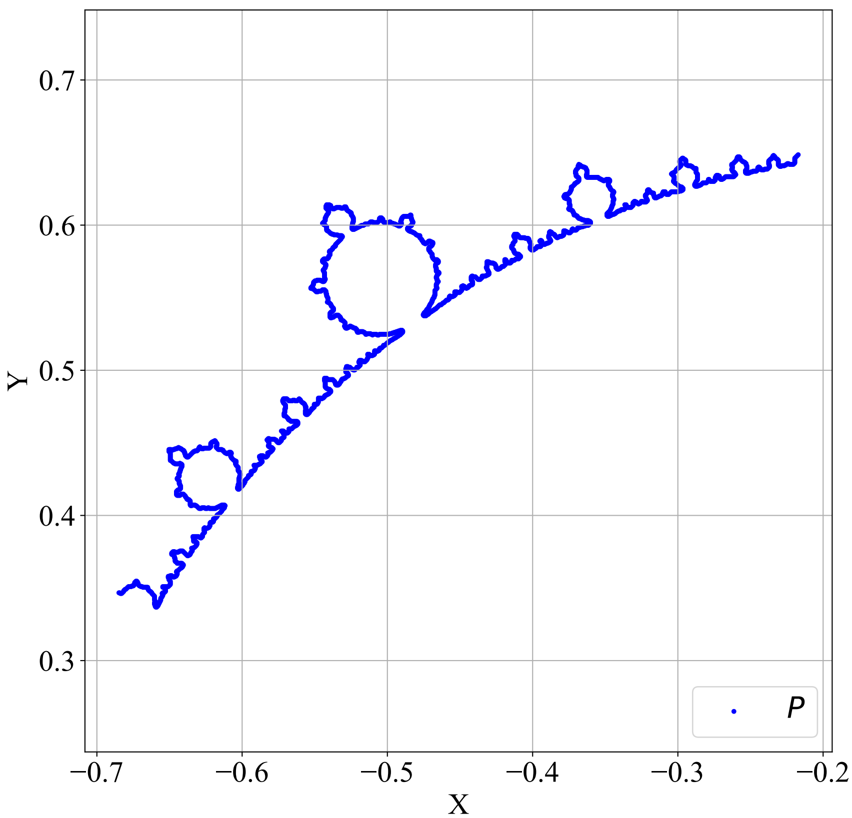

Рис. 1. Визуализация множества точек P

4 Получение кубического сплайна

Определение 2

Пусть функция $f(x)$ задана в n интерполяционных узлах $a = x_1, x_2, \ldots, x_n = b$ на отрезке $[a; b]$. Тогда кубическим сплайном для функции $f(x)$ называется функция $S(x)$, для которой верно:

1. $S(x)$ кусочно задана кубическими многочленами $S_i(x)$ на каждом отрезке $[x_i; x_{i+1}]$, $i = 1, ..., n-1$;

2. $S_i(x_i) = f(x_i)$ и $S_i(x_{i+1}) = f(x_{i+1}), i = 1, \ldots, n-1$;

3. значения смежных многочленов совпадают в общих узлах: $S_i(x_{i+1}) = S_{i+1}(x_{i+1}), i = 1, \ldots, n-2$;

4. значения первых производных смежных многочленов совпадают в общих узлах: $S'_i(x_{i+1}) = S'_{i+1}(x_{i+1}), i = 1, \ldots, n-2$;

5. значения вторых производных смежных многочленов совпадают в общих узлах: $S''_i(x_{i+1}) = S''_{i+1}(x_{i+1}), i = 1, \ldots, n-2$;

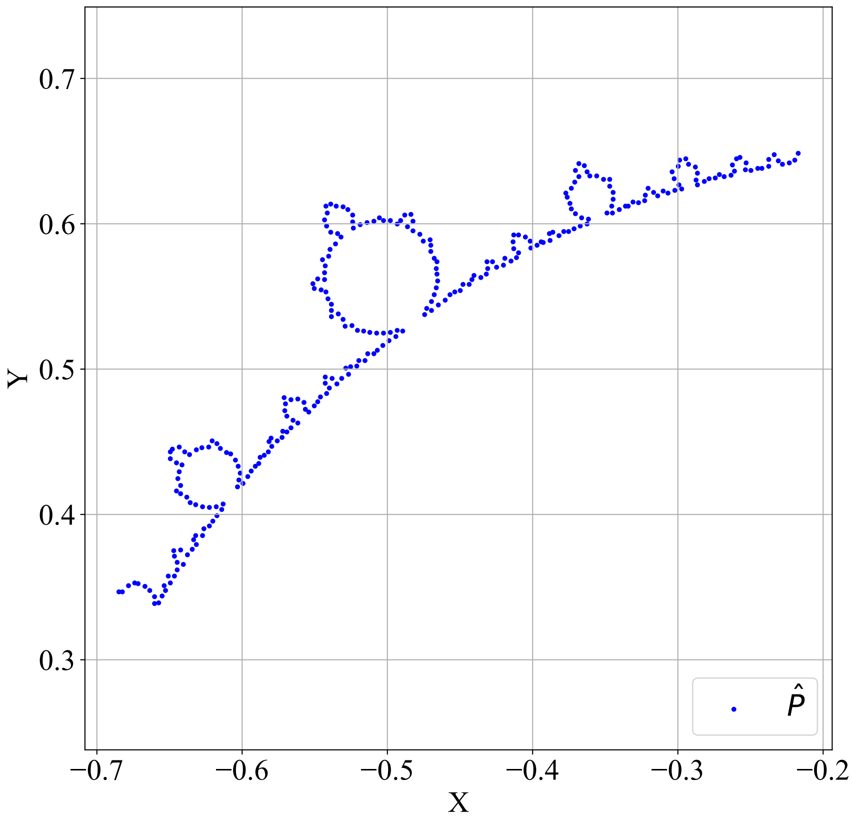

Рис. 2. Визуализация разреженного множества интерполяционных узлов $\hat{P}$

6. заданы граничные условия:

   - естественные граничные условия $S''(x_1) = S''(x_n) = 0$;
   - граничные условия на касательную $S'(x_1) = f'(x_1)$ и $S'(x_n) = f'(x_n)$ [2].

Для получения естественного параметрического кубического сплайна (2), необходимо определить натуральные координаты, которые будут равномерно распределены вдоль параметра t. Эти координаты вычисляются от 0 до значения, равного общему количеству узлов в множестве точек, деленному на параметр M. Шаг определяется как произведение значений factor и h. Параметры M, factor, и h имеют заранее заданные значения в соответствии с условиями задачи [1]. Расстояния между натуральными координатами определяются вычисления использования разностей между последовательными значениями. Код, реализующий данные действия, представлен в листинге 2.

Листинг 2. Получение естественных координат и расстояний между ними

```
t = np.arange(0, points_x.shape[0] / M, factor * h)
if t[-1] != points_x.shape[0] / M:
    t = np.append(t, points_x.shape[0] / M)
t_dist = np.diff(t)
```

Коэффициенты естественного параметрического кубического сплайна получаются путём решения соответствующих разрешающих систем линейных алгебраических уравнений. Матричное уравнение $\boldsymbol{Ac} = \boldsymbol{b}$ для кубического сплайна имеет вид:

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

$$
= \begin{bmatrix}
0 \\
\frac{3}{h_2}(a_3-a_2) - \frac{3}{h_1}(a_2-a_1) \\
\frac{3}{h_3}(a_4-a_3) - \frac{3}{h_2}(a_3-a_2) \\
\vdots \\
\frac{3}{h_{n-1}}(a_n-a_{n-1}) - \frac{3}{h_{n-2}}(a_{n-1}-a_{n-2}) \\
0
\end{bmatrix}, \tag{4}
$$

где a_i, c_i - коэффициенты сплайна, $h_i = x_{i+1} - x_i$.

Коэффициенты a_i, b_i, d_i вычисляются по формулам (5)-(7), а c_i — это элементы вектор-столбца $\boldsymbol{c}$, который равен $\boldsymbol{Ab}^{-1}$.

$$a_i = f(x_i), \tag{5}$$

$$b_i = \frac{1}{h_i}(a_{i+1} - a_i) - \frac{h_i}{3}(c_{i+1} + 2c_i), \tag{6}$$

$$d_i = \frac{c_{i+1} - c_i}{3h_i}[2]. \tag{7}$$

5 Вычисление расстояний

Для вычисления расстояний между точками реального контура и соответствующими точками на сплайне, применяется формула (8).

$$\rho[(\tilde{x}(t_i), \tilde{y}(t_i)), (x(t_i), y(t_i))] = \sqrt{(\tilde{x}(t_i) - x(t_i))^2 + (\tilde{y}(t_i) - y(t_i))^2} \tag{8}$$

В листинге 3 представлен код, выполняющий вычисление расстояний между каждой парой точек на множестве P и соответствующей точке на сплайне $S(t)$. Затем для полученных расстояний вычисляются среднее и стандартное отклонение.

Листинг 3. Вычисление расстояния ρ, среднего и стандартного отклонения

```
distance = []
for real_x, real_y, approx_x, approx_y in zip(points_x, points_y, spline_x, spline_y):
    d = ((real_x - approx_x) ** 2 +(real_y - approx_y) ** 2)**0.5
    distance.append(d)
print("Среднее отклонение:", np.array(distance).mean())
print("Стандартное отклонение:", np.array(distance).std())
```

Полученные результаты для рассматриваемого контура следующие:
Среднее отклонение: 0.0005268200716338058
Стандартное отклонение: 0.00040686144658110197

6 Отображение полученного сплайна

Пусть естественный параметрический кубический сплайн будет обозначен следующим образом: $S(t) = (\tilde{x}(t), \tilde{y}(t))$. Используя параметр $t \in [0, t_N]$ с частым шагом $h = 0.1$ и учитывая исходное множество точек P вместе с узловыми точками $\hat{P}$, формируется график, который представлен на рисунке 3.

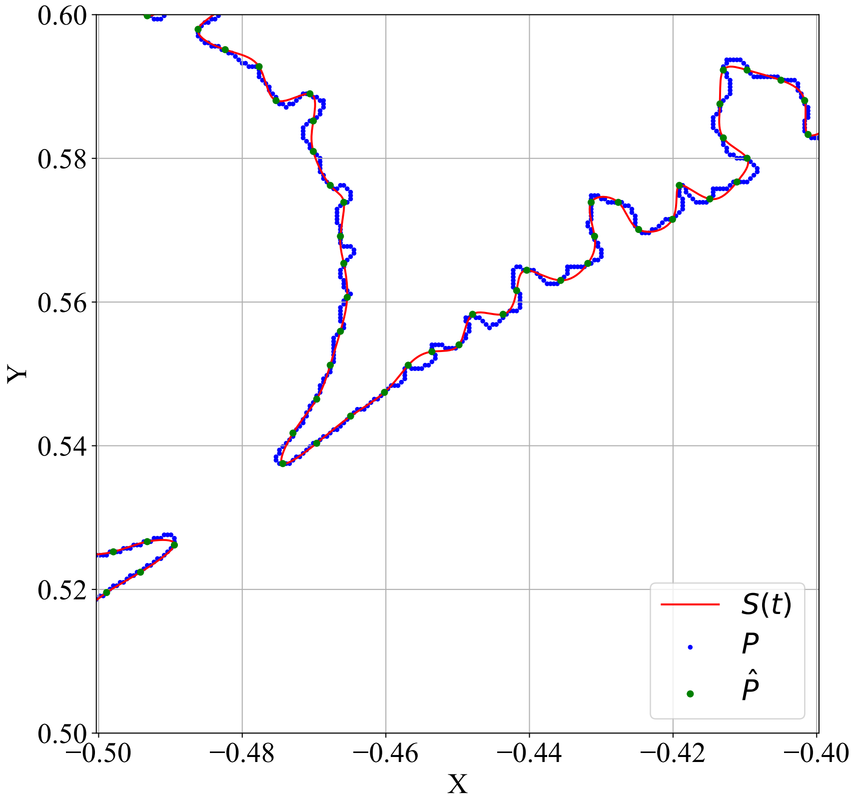

Рис. 3. Визуализация полученных сплайнов $S(t)$, исходного множества точек P и множества интерполяционных узлов $\hat{P}$

Наблюдаемая ошибка интерполяции связана со сложной формой фрактала. Она является естественным результатом ограниченности выбора узлов и степени интерполяционного полинома. Уменьшение ошибки требует баланса между увеличением числа узлов и использованием более сложных методов интерполяции.

7 Разработка основной функции

Реализована 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 (по умлочанию coeffs.txt), содержащего коэффициенты a_{jk} и b_{jk} в виде матрицы размером $\hat{N} - 1$ строк на 8 столбцов.

8 Реализация класса *AutoDiffNum*

Определение 3

Дуальным числом $\langle a, b\rangle$ называется число, представимое парой чисел $a, b \in \mathbb{R}$, так что

$$\langle a, b\rangle = a + b\epsilon, \tag{9}$$

где ϵ – такое бесконечно малое число, что $\epsilon^2 = 0$ и $\epsilon \neq 0$.

Таким образом, для реализации класса *AutoDiffNum* требуется задать 2 поля, обозначающие действительную и мнимую часть дуального числа, а также перегрузку операторов сложения и умножения (листинг 4).

Листинг 4. Реализация класса *AutoDiffNum*

```
class AutoDiffNum:
    def __init__(self, a, b=0):
        self._a = a
        self._b = b
    def __add__(self, other):
        if isinstance(other, AutoDiffNum):
            return AutoDiffNum(self._a + other._a, self._b + other._b)
        return AutoDiffNum(self._a + other, self._b)
    def __mul__(self, other):
        if isinstance(other, AutoDiffNum):
            return AutoDiffNum(self._a * other._a, self._a * other._b + self._b *
                other._a)
        return AutoDiffNum(self._a * other, self._b * other)
```

9 Реализация функции автоматического расчёта первой производной

Пусть $f(x)$ является аналитической функцией на замкнутом интервале $x \in [\alpha; \beta]$ и $\langle a, b\rangle$ является дуальным числом. Тогда $f(\langle a, b\rangle) = f(a) + b\epsilon f'(a)$. Благодаря этому свойству можно вычислить значения функции и ее производной, положив $\alpha = t[i]$, где $t[i]$ — это значение параметра t в некоторой точке, а $\beta = 1$.

Реализация функции автоматического расчёта первой производной кубического сплайна $G(t) = \frac{d}{dt}(\tilde{x}(t), \tilde{y}(t))$ представлена в листинге 5.

Листинг 5. Реализация функции автоматического дифференцирования

```
def spline_deriv(ax, bx, cx, dx, ay, by, cy, dy, t_j, t):
    dxdt = (t − t_j) ∗ (t − t_j) ∗ (t − t_j) ∗ dx + (t − t_j) ∗ (t − t_j) ∗ cx + (t − t_j) ∗ bx + ax
    dydt = (t − t_j) ∗ (t − t_j) ∗ (t − t_j) ∗ dy + (t − t_j) ∗ (t − t_j) ∗ cy + (t − t_j) ∗ by + ay
    return dxdt._b, dydt._b
```

10 Построение нормали к заданному вектору

Формула (10) представляет собой функцию нормали $R(t)$ к заданному вектору $G(t)$.

$$R(t) = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \frac{d\tilde{x}(t)}{dt} \\ \frac{d\tilde{y}(t)}{dt} \end{bmatrix} = \begin{bmatrix} -\frac{d\tilde{y}(t)}{dt} \\ \frac{d\tilde{x}(t)}{dt} \end{bmatrix}. \tag{10}$$

11 Визуализация касательной и нормали

Результат построения векторов $G(t_j)$ и $R(t_j)$ в соответствующих точках сплайна на некотором участке изображен на рисунке 4.

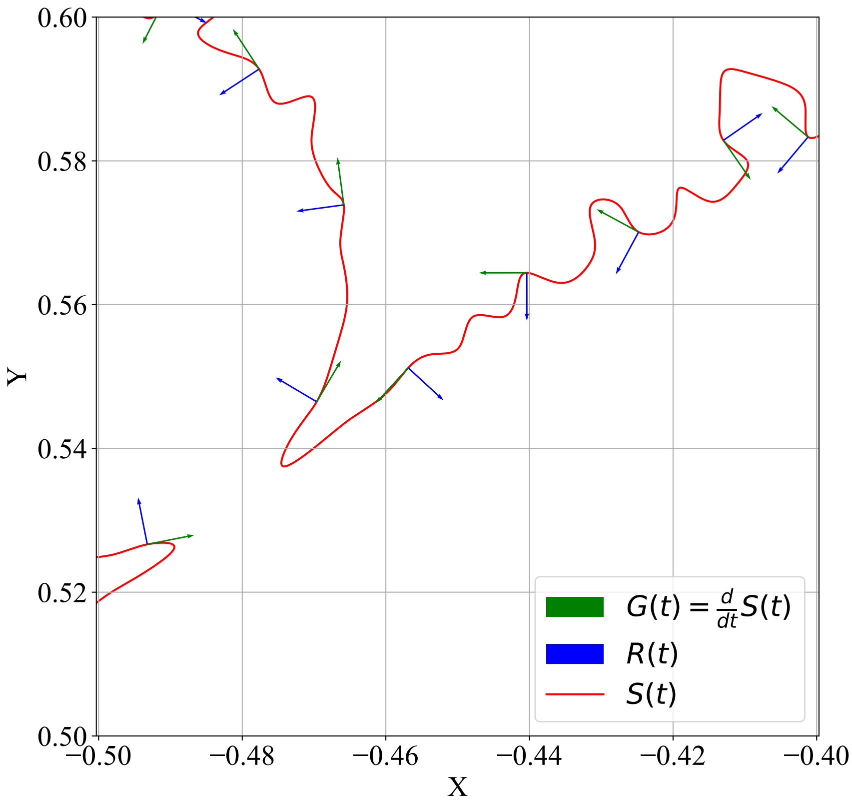

Рис. 4. Векторы G(t) и нормаль R(t) в некоторых точках сплайна

12 Решение оптимизационной задачи

Решение оптимизационной задачи заключается в том, что для каждой пропущенной точки (x_i, y_i) исходного контура необходимо найти (численно) ближайшую на соответствующем участке сплайна (3).

Для каждой пропущенной точки (x_i, y_i) контура вычисляется ближайшая точка на сплайне путем решения оптимизационной задачи, представленной в формуле (3), с использованием тернарного метода до десяти итераций. Полученное множество ближайших точек обозначается как P^*.

График, иллюстрирующий ближайшие пропущенные точки и точки на сплайне, представлен на рисунке 5.

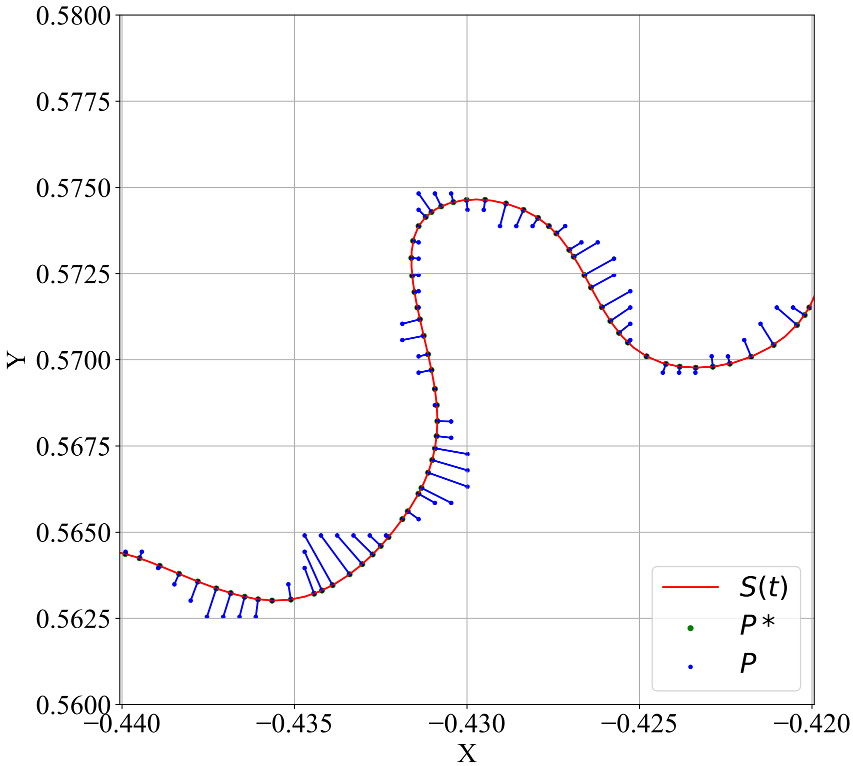

Рис. 5. Визуализация исходного множества P, соединённого с множеством ближайших на сплайне $S(t)$ точек P^*

Результаты оптимизационной задачи:
Среднее отклонение: 0.00045600041387353746
Стандартное отклонение: 0.0004014090087739599

Применение оптимизации приводит к уменьшению среднего отклонения на 0.0000708159328984312 и стандартного отклонения на 0.000005452437807142. С учетом вычисленной средней оценки погрешности в 0.0004914121051845902, относительная погрешность для среднего отклонения составляет 13.42% и для стандартного отклонения — 1.34%. Эти результаты указывают на улучшенное приближение сплайна к исходным данным после оптимизации.

Заключение

В ходе лабораторной работы были выполнены следующие задачи:

1. В файле contours.txt сформирован фрагмент границы множества Мандельброта.
2. Загружено и визуализировано множество точек P из файла contours.txt.
3. Сформировано разреженное множество интерполяционных узлов $\hat{P}$ с использованием параметра $M = 10$.
4. Найдены коэффициенты a_{jk} и b_{jk} для кубического сплайна.
5. Вычислены расстояния между исходными и интерполярованными точками.
6. Визуализирован резултат интерполяции кубическими сплайнами. Определены причины наблюдаемой ошибки интерполяции, а так же способы её уменьшения.
7. Разработана функция lab1_base для автоматизации этапов базовой части.
8. Разработан класс *AutoDiffNum* для автоматического вычисления производных с использованием дуальных чисел.
9. Реализован автоматический расчет первой производной кубического сплайна.
10. Реализована функция построения нормали к вектору сплайна.
11. В соответствующих точках сплайна визуализированы векторы $G(t_j)$ и $R(t_j)$.
12. Решенна оптимизационная задача, результаты проанализированы.

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. Першин А.Ю., Соколов А.П., Гудым А.В. Сборник постановок задач на лабораторные работы по курсу «Вычислительная математика»: Учебное пособие. [Электронный ресурс]. Москва, 2023. С. 47. URL: https://archrk6.bmstu.ru. (облачный сервис кафедры РК6).

Выходные данные

Рогалева Ю.А.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)

2023, осенний семестр