



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 ДОМАШНЕГО ЗАДАНИЯ

по дисциплине «Вычислительная математика»

Студент:	Антоненко Григорий Андреевич
Группа:	РК6-53Б
Тип задания:	Лабораторная работа №1
Тема:	Интерполяция параметрическими кубическими сплайнами

Студент

подпись, дата

Антоненко Г.А.
Фамилия, И.О.

Преподаватель

подпись, дата

Фамилия, И.О.

Москва, 2023

Содержание

Интерполяция параметрическими кубическими сплайнами	3
1 Задание	3
2 Цель выполнения лабораторной работы	4
3 Выполнение	5
4 Заключение	10

Интерполяция параметрическими кубическими сплайнами

1 Задание

В данной работе потребуется интерполировать некоторый фрагмент границы фрактала \mathbf{c} , описываемой в параметрическом виде неизвестными функциями $x(t), y(t)$, на основе его отдельных точек, используя параметрически задаваемый кубический сплайн.

Базовая часть

1. Используя заранее подготовленный скрипт, выбрать произвольную область множества Мандельброта и построить фрагмент его границы (контура), сформировав файл contours.txt. Файл contours.txt содержит упорядоченную последовательность точек на плоскости $P = (x_i, y_i) N_i = 1$, принадлежащих выбранному фрагменту границы фрактала \mathbf{c} . Сопоставляя каждой паре координат естественную координату t , предполагая, 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 $\hat{P} = (x_j, y_j)_{j=1}^{\hat{N}}$, $\hat{N} = \lfloor N/M \rfloor$, $j = M \times i$, $\hat{P} \subset P$.

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

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

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

5. Вычислить расстояние $\rho[(\tilde{x}(t_i), \tilde{y}(t_i); (x(t_i), y(t_i)))]$ и представить вывод (среднее и стандартное отклонение) в отчете

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 так же 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}, b_{jk} в виде матрицы размером $\hat{N}-1$ строк на 8 столбцов. Функция lab1_base должна реализовывать базовую часть задания.

2 Цель выполнения лабораторной работы

Цель выполнения лабораторной работы: Научиться использовать библиотеки Matplotlib и NumPy; научиться интерполировать некоторый фрагмент границы фрактала с, описываемой в параметрическом виде неизвестными функциями $x(t)$, $y(t)$, на основе его отдельных точек, используя параметрически задаваемый кубический сплайн.

3 Выполнение

Задача 1

Используя заранее подготовленный скрипт, выбрать произвольную область множества Мандельброта и построить фрагмент его границы (контура), сформировав файл contours.txt. Файл contours.txt содержит упорядоченную последовательность точек на плоскости $P = (x_i, y_i) N_i = 1$, 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).

Используйте инструмент ПРИБЛИЖЕНИЯ(лупа) для выбора области, приближать можно много раз.
После выбора, окно можно закрыть, контур сохранится.

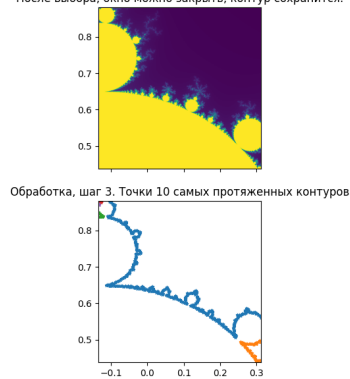


Рис. 1. Сверху – отображение точек фрактала (ярко-желтый) на комплексной плоскости, граничные и пр. точки в виде оттенков. Снизу – отдельные точки, формирующие приблизительные наиболее длинные непрерывные контуры множества

Для выполнения задания была реализована функция `visualise`, которая отображает полученное множество точек P . Координаты читаются из файла методом `<файл>.readlines()`, отображение осуществляется методом `scatter` библиотеки `matplotlib`.



Задача 3

С помощью срезов списков были заданы координаты узлов интерполяции — множество $\hat{P} = (x_j, y_j)_{j=1}^{\hat{N}}$

Задача 4

По каждому измерению были найдены коэффициенты естественного параметрического кубического сплайна a_{jk} и b_{jk} путем решения соответствующих разрешающих СЛАУ.

Для этого определена функция `cubic_spline_interpolation`. Она принимает в качестве аргументов массив значений t (`np.arange(len(x_selected))`), и значения функций $\tilde{x}(t_i)$ и $\tilde{y}(t_i)$. В этой функции составлялось и решалось матричное уравнение вида $A\mathbf{c} = \mathbf{B}$. В выглядят так:

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

где $h_i = t_{i+1} - t_i$

$$B = \begin{bmatrix} 0 \\ \frac{3}{h_2}(a_{30} - a_{20}) - \frac{3}{h_1}(a_{20} - a_{10}) \\ \frac{3}{h_3}(a_{40} - a_{30}) - \frac{3}{h_2}(a_{30} - a_{20}) \\ \vdots \\ \frac{3}{h_{n-1}}(a_{n0} - a_{(n-1)0}) - \frac{3}{h_{n-2}}(a_{(n-1)0} - a_{(n-2)0}) \\ 0 \end{bmatrix}$$

Вектор \mathbf{c} представляет собой столбец $\{a_{i2}\}_{i=1}^n$. Уравнение решается относительно этого вектора методом библиотеки Numpy — `np.linalg.solve(A, B)`. Следующий шаг, выполняемый функцией — нахождение оставшихся коэффициентов a_{i1} , a_{i3} и b_{i1} , b_{i3} .

$$a_{i1} = \frac{1}{h_i}(a_{(i+1)0} - a_{i0}) - \frac{h_i}{3}(a_{(i+1)2} - 2a_{i2})$$

$$a_{i3} = \frac{a_{(i+1)2} - a_{i2}}{3h_i}$$

Аналогично находятся коэффициенты b_{ij}

Ниже приведен листинг этой функции

```

1 def cubic_spline_interpolation(x_, y_):
2     n = len(x_) - 1
3     h_ = np.diff(x_)
4
5     A = np.zeros((n + 1, n + 1))
6     A[0, 0] = 1
```

```

7   A[-1, -1] = 1
8   for i in range(1, n):
9       A[i, i - 1] = h_[i - 1]
10      A[i, i] = 2 * (h_[i - 1] + h_[i])
11      A[i, i + 1] = h_[i]
12
13  B = np.zeros(n + 1)
14  for i in range(1, n):
15      B[i] = 3 * ((y_[i + 1] - y_[i]) / h_[i] - (y_[i] - y_[i - 1]) / h_[i - 1])
16
17  c = np.linalg.solve(A, B)
18
19  a = y_
20  b = np.zeros(n)
21  d = np.zeros(n)
22  for i in range(n):
23      b[i] = (a[i + 1] - a[i]) / h_[i] - h_[i] * (c[i + 1] + 2 * c[i]) / 3
24      d[i] = (c[i + 1] - c[i]) / (3 * h_[i])
25
26  return a, b, c, d

```

Задача 5

В цикле были найдены расстояния между точками исходного множества точек и точками, рассчитанными сплайном, с помощью теоремы Пифагора. Затем были найдены среднее и стандартное отклонения.

Среднее отклонение = 0.000481055743833842

Стандартное отклонение = 0.0003764030568006566

Задача 6

С помощью функции `compute_spline`, которая принимает в качестве аргументов коэффициенты сплайна и значения **размноженного** параметра t . По полученным значениям, вместе с отображением узлов и исходного множества точек строится сплайн. Возникающая ошибка связана с шагом выбора узлов интерполяции, для её уменьшения можно уменьшить этот шаг.



Рис. 3. Вывод на координатной плоскости точек исходного контура, узлов интерполяции и кубического сплайна

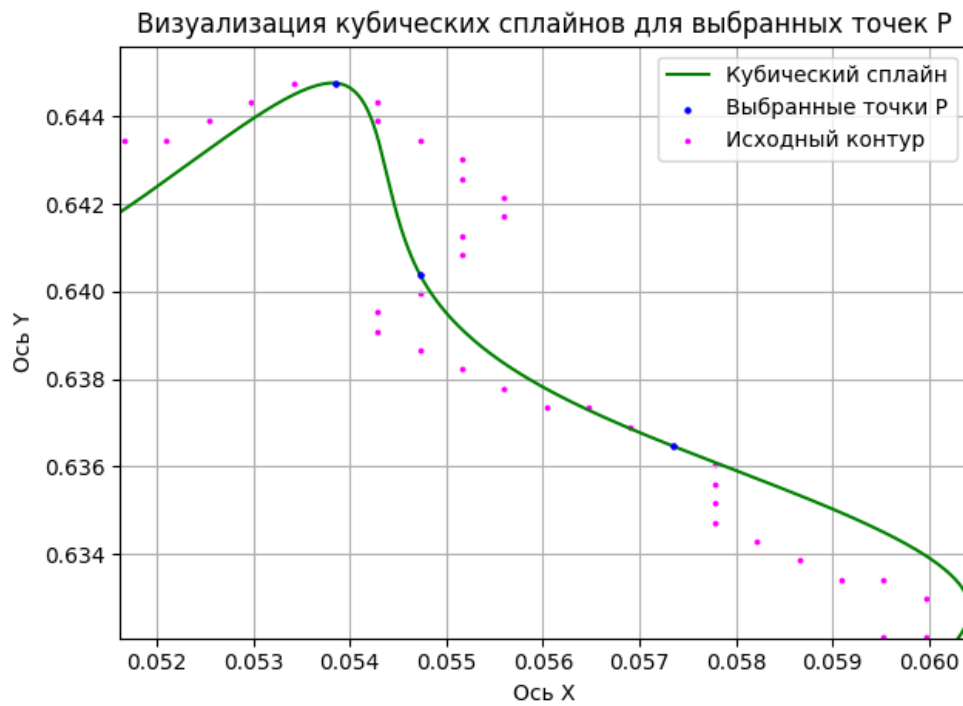


Рис. 4. Увеличенное изображение

```

1  def compute_spline(a, b, c, d, t_):
2      n = len(a) - 1
3      f_ = np.zeros_like(t_)
4
5      for i in range(n):
6          mask = (t_ >= i) & (t_ <= i + 1)
7          dx = t_[mask] - i
8          f_[mask] = a[i] + b[i] * dx + c[i] * dx ** 2 + d[i] * dx ** 3
9
10     return f_

```

Задача 7

В результате 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.

4 Заключение

1. Изучены приложения библиотек Numpy и Matplotlib
2. Изучен метод интерполяции кубическими сплайнами

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).