ГУАП

КАФЕДРА № 42

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

канд. техн. наук, доцент		А. В. Аграновский
должность, уч. степень, звание	подпись, дата	инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ № 1

ИССЛЕДОВАНИЕ ФРАКТАЛЬНОЙ ГРАФИКИ

по курсу:

КОМПЬЮТЕРНАЯ ГРАФИКА

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ гр. №	4326		Г. С. Томчук
		подпись, дата	инициалы, фамилия

Санкт-Петербург 2024

1 Цель работы

Цель лабораторной работы представляет из себя изучение теоретических основ фрактальной графики и получение практических навыков визуализации фракталов с помощью языка программирования и графической библиотеки. Вопросы, изучаемые в данной работе:

- Алгоритм метода Ньютона и его применение для решения нелинейных уравнений.
- Использование численных методов (метода Ньютона) для визуализации математических объектов (фракталов).
- Разработка программы, которая строит фрактал Ньютона для заданной функции, задавая диапазон координат и параметры отображения, с возможностью настройки цветовой гаммы.
- Работа с комплексными числами и использование графических библиотек Python (matplotlib) для отображения сложных объектов.
- Изучение свойств фрактала Ньютона: симметрия, точки сходимости и границы, влияние параметров итераций на результат.

2 Формулировка задания

Необходимо реализовать программу построения фрактала Ньютона на любом языке высокого уровня в соответствии с заданием. Предусмотреть возможность установки диапазона изменения координат. Использование для генерации фракталов готовых программных продуктов не разрешается.

Вариант 5:

$$f(z) = z^5 - 1$$

3 Теоретические сведения

Фрактал — это геометрическая фигура, обладающая самоподобием, то есть повторяющая свою структуру на разных масштабах. Фракталы часто имеют сложную, бесконечную детализацию и могут быть описаны через рекуррентные или итерационные процессы.

Фрактал Ньютона — это визуализация метода Ньютона, применённого к нахождению корней нелинейного уравнения. Он создаётся для функций комплексного переменного, где для каждой точки на плоскости (представленной комплексным числом) проверяется, к какому корню сойдётся метод Ньютона.

Метод Ньютона — это численный метод нахождения корней функции f(z)=0, где z — переменная комплексного типа. Для каждого начального значения z_0 итеративно рассчитывается последовательность значений по формуле:

$$z_{n+1} = z_n - \frac{f(z_n)}{f'(z_n)} \quad (1)$$

где $f'(z)$ — производная функции $f(z)$. Процесс повторяется до тех пор, пока не достигнем заданной точности.

Для построения фрактала Ньютона используется конкретная функция, $f(z) = z^5 - 1$, где производная $f'(z) = 5z^4$. У этой функции пять корней, равномерно распределённых на комплексной плоскости по окружности радиуса 1.

Для визуализации фрактала каждая точка комплексной плоскости окрашивается в цвет, который соответствует корню, к которому эта точка сходится. Дополнительно количество итераций, потребовавшихся для сходимости, может использоваться для регулирования интенсивности или оттенков цвета.

На комплексной плоскости каждая точка имеет реальные и мнимые компоненты, которые можно отобразить на двумерную область. Задаются границы по осям X и Y, которые определяют область исследования и масштаб изображения.

4 Алгоритм построения заданного фрактала

1. Определить диапазоны изменения координат (реальной и мнимой

части) и размеры сетки (ширина, высота).

2. Создать сетку комплексных чисел $z = x + iy$ для каждой точки на плоскости.
3. Для каждой точки z_0, применить итерационную формулу метода Ньютона:

$$z_{n+1} = z_n - \frac{f(z_n)}{f'(z_n)}$$

до тех пор, пока не будет достигнута сходимость к одному из корней или пока не истечёт максимальное количество итераций.

4. Определить, к какому корню (из 5 возможных) сходится каждая точка.
5. Каждой точке присвоить цвет, соответствующий корню, к которому она сходится, и количество итераций для дополнительного оттенка.
6. Построить изображение фрактала на основе вычисленных цветов для каждой точки сетки.

5 Выбор языка программирования и используемых библиотек

Для написания программы построения фрактала Ньютона я выбрал язык программирования Python. Данный язык имеет богатую экосистему библиотек, в том числе и необходимых для работы с отрисовкой и графикой. Помимо этого, я уже имел опыт работы с данным языком, что ускорило и облегчило рабочий процесс.

Используемые библиотеки:

- NumPy — библиотека для работы с многомерными массивами и матрицами. Она обеспечивает высокую производительность при математических операциях и используется для создания сетки комплексных чисел и вычислений в методе Ньютона.
- Matplotlib — библиотека для создания графиков и визуализации данных. В данной задаче используется для отображения фрактала в виде изображения, где каждой точке на плоскости присваивается определённый цвет на основе её сходимости.

6 Описание программы построения фрактала

Написанная программа реализует метод Ньютона для функции, выполняя итерационные вычисления для каждой точки на комплексной плоскости и визуализируя результат.

Основные компоненты:

1. Импорт библиотек
   - numpy используется для работы с массивами комплексных чисел.
   - matplotlib — для отображения и визуализации фрактала.
2. Функции f(z) и f_derivative(z) определяют основную функцию и её производную, необходимые для метода Ньютона.
3. Функция newton_fractal
   - Основная функция программы, которая принимает параметры для диапазона координат, размер изображения и количество итераций.
   - Создаёт сетку комплексных чисел для каждой точки изображения.
   - Для каждой точки применяет метод Ньютона для нахождения корня, повторяя процесс до достижения сходимости или максимального числа итераций.
4. Цветовая палитра. Заранее определяются пять цветов, соответствующих каждому из пяти корней функции. Каждый пиксель изображения окрашивается в цвет, связанный с тем корнем, к которому сходится точка.
5. Генерация и отображение:
   - Для каждой точки вычисляется, к какому корню она сходится, и ей присваивается соответствующий цвет.
   - Сгенерированное изображение выводится на экран с

помощью matplotlib.

Основные параметры:

x_min, x_max, y_min, y_max — диапазон значений для отображаемой области комплексной плоскости.

- width, height — размер изображения.
- max_iter — максимальное количество итераций метода Ньютона для каждой точки.

7 Скриншоты с изображениями фрактала

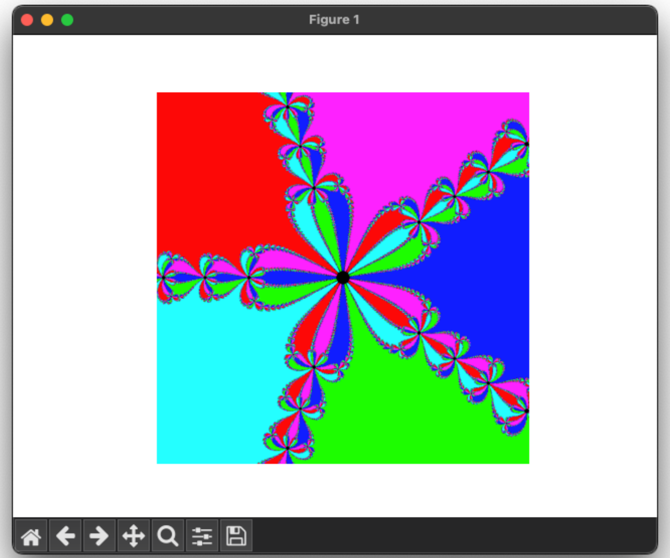

Рисунок 1

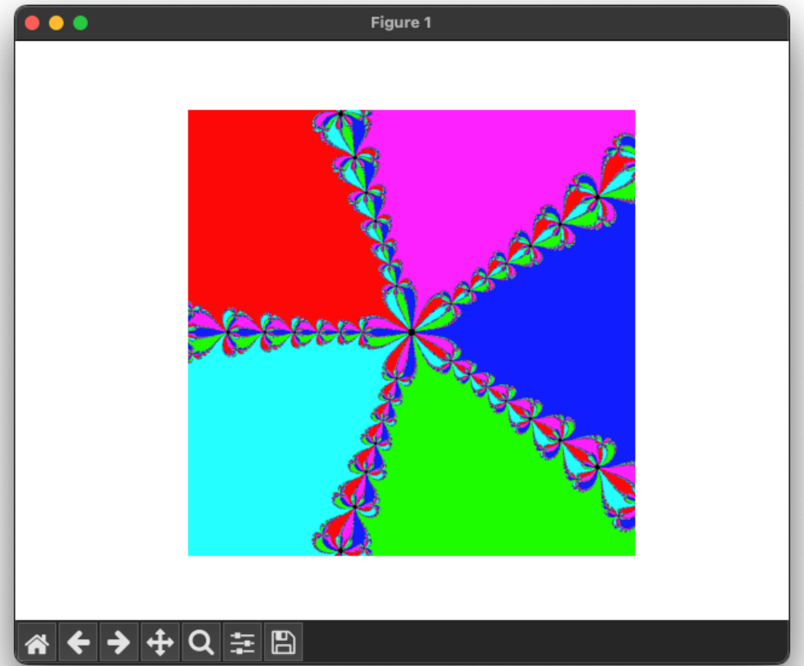

Рисунок 2

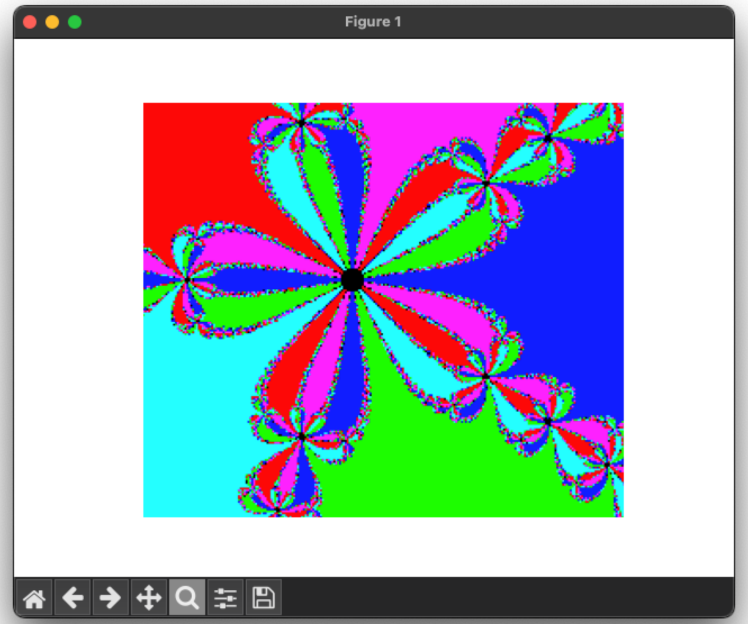

Рисунок 3

8 Вывод

В результате выполнения данной лабораторной работы была реализована программа для построения фрактала Ньютона для функции $f(z) = z^5 - 1$ с использованием метода Ньютона. Программа позволяет генерировать фрактал в заданном диапазоне координат на комплексной плоскости и визуализировать его с использованием собственной цветовой гаммы.

Освоенные теоретические знания:

1. Метод Ньютона — изучен и применён для поиска корней нелинейных уравнений. Этот численный метод показал свою эффективность для решения сложных функций и их визуализации на комплексной плоскости.
2. Фракталы — было подробно изучено понятие фрактала и особенности его структуры. Фракталы являются примерами самоподобных объектов с бесконечной детализацией, и в работе исследован фрактал Ньютона, который наглядно демонстрирует, как различные начальные значения комплексных чисел сходятся к

разным корням функции.

3. Работа с комплексными числами — навыки работы с комплексными числами были закреплены через программную реализацию операций с ними, а также через их отображение на плоскости.

Приобретённые навыки:

1. Программирование на Python — работа позволила улучшить навыки работы с такими библиотеками, как NumPy и Matplotlib. Библиотека NumPy оказалась особенно полезной для выполнения операций над массивами комплексных чисел и эффективного вычисления итераций метода Ньютона.
2. Визуализация данных — изучены и применены методы визуализации сложных математических объектов (фракталов) с помощью библиотеки Matplotlib. Навыки построения графиков и работы с цветами были также развиты в ходе создания цветовой гаммы для фрактала.
3. Численные методы — был углублён опыт использования численных методов, таких как метод Ньютона, для решения задач, связанных с нахождением корней нелинейных уравнений и построением визуальных результатов.

Проблемы и пути их решения:

1. Проблема с точностью сходимости. В ходе работы возникала проблема с выбором порога точности для сходимости точек к корням. Если порог слишком мал, процесс становился менее эффективным, так как увеличивалось количество итераций. Решение состояло в установке разумного порога точности (например, 1e-6)), что обеспечило баланс между точностью и скоростью работы программы.

2. Производительность программы. Для больших размеров изображения или большого числа итераций время работы программы значительно увеличивалось. Это связано с вычислительной сложностью метода Ньютона. Проблема была частично решена за счёт использования оптимизированных

операций с массивами в библиотеке NumPy, что позволило параллельно обрабатывать множество точек на комплексной плоскости.

3. Выбор цветовой палитры. Определение подходящей цветовой гаммы также представляло сложность. Разные корни функции требуют разных цветов, и важно было подобрать такие оттенки, которые чётко разделяли области сходимости. В результате было принято решение использовать контрастные цвета для каждого корня, а также дополнительно кодировать количество итераций через вариации оттенков.

Выполнение лабораторной работы позволило не только закрепить теоретические знания по численным методам и фракталам, но и приобрести практические навыки программирования и визуализации данных. Возникшие в ходе работы проблемы были решены с использованием эффективных подходов, что позволило добиться требуемого результата — построения и визуализации фрактала Ньютона с настраиваемыми параметрами.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Ахо А. В., Хопкрофт Д. Э., Ульман Д. Дж. Структуры данных и алгоритмы. — М.: Мир, 1983. — 400 с.
2. Зорич В. А. Математический анализ. Часть 1. — 5-е изд., испр. — М.: Фазис, 2004. — 560 с.
3. Иванов И. И., Петров П. П. Численные методы решения нелинейных уравнений. — СПб.: БХВ-Петербург, 2010. — 256 с.
4. Дьяконов В. П. MATLAB 6/6.1/6.5: Специальный справочник. — М.: СОЛОН-Пресс, 2003. — 512 с.
5. Кнут Д. Э. Искусство программирования, том 1: Основные алгоритмы. — 3-е изд. — М.: Вильямс, 2005. — 752 с.

ПРИЛОЖЕНИЕ А

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


# Заданная функция: f(z) = z^5 - 1
def f(z): return z ** 5 - 1


# Производная заданной функции: f'(z) = 5z^4
def f_derivative(z): return 5 * z ** 4


# Метод Ньютона
def newton_fractal(x_min, x_max, y_min, y_max, width, height,
iteration_max=100, tolerance=1e-6):
    # Задаём сетку значений на комплексной плоскости
    real = np.linspace(x_min, x_max, width)
    imaginary = np.linspace(y_min, y_max, height)
    X, Y = np.meshgrid(real, imaginary)
    Z = X + 1j * Y

    # Корни и цвета
    roots = np.array([1, np.exp(2j * np.pi / 5), np.exp(4j * np.pi / 5),
np.exp(6j * np.pi / 5), np.exp(8j * np.pi / 5)])
    colors = np.array([[0, 0, 255], [0, 255, 0], [0, 255, 255], [255, 0, 0],
[255, 0, 255]])

    # Инициализируем массив для хранения цветов
    image = np.zeros((height, width, 3), dtype=np.uint8)

    for i in range(iteration_max):
        # Вычисляем новое приближение
        Z -= f(Z) / f_derivative(Z)

        # Для каждого корня проверяем, близко ли текущее значение к нему
        for k, root in enumerate(roots):
            mask = np.abs(Z - root) < tolerance
            image[mask] = colors[k]

    return image


# Параметры фрактала
x_min, x_max, y_min, y_max = -1.5, 1.5, -1.5, 1.5
width, height = 800, 800
iteration_max = 50

# Генерация фрактала
fractal = newton_fractal(x_min, x_max, y_min, y_max, width, height,
iteration_max)

# Отображаем фрактал
plt.imshow(fractal)
plt.axis('off')
plt.show()
```