МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ М. В. ЛОМОНОСОВА
ФИЛИАЛ МОСКОВСКОГО ГОСУДАРСТВЕННОГО
УНИВЕРСИТЕТА ИМЕНИ М. В. ЛОМОНОСОВА В ГОРОДЕ
СЕВАСТОПОЛЕ

Факультет «Компьютерной математики»
Направление подготовки «Прикладная математика и информатика»
01.03.02 (бакалавр)

ОТЧЕТ

по лабораторной работе №2

«Локализация цепно-рекуррентного множества для отображения Жюлиа»

Работу выполнил:
студент группы ПМ-401
Хаметов Марк Владимирович

Руководитель: профессор
кафедры прикладной
математики и информатики
Осипенко Георгий Сергеевич

Севастополь, 2023

Оглавление

Постановка задачи

Дана динамическая система в области [-2;-2]x[2;2]. Необходимо найти достаточно малую окрестность цепно-рекуррентное множества отображения этой динамической системы.

Отображение Жулия задается следующим образом:

$$x_n = x_{n-1}^2 - y_{n-1}^2 + a$$
$$y_n = 2 * y_{n-1} * x_{n-1} + b$$

Для построения символического образа цепно-рекуррентного множества берется область. Эту область впоследствии разбивают на ячейки. Ячейки соответствующие цепно-рекуррентному множеству закрашиватся. Размер ячейки не превышает размера пикселя.

Теоретическая часть

Сильно связанные вершины графа – это подмножества таких вершин ориентированного графа, между которыми существует путь в обоих направлениях.

Разбиение области на ячейки в данной работе - это разбиение на прямоугольники одинакового размера. Длина ребер задается пользователем. Нумерация ячеек идет в порядке сначала слева направо, затем сверху вниз. Тогда обозначим ячейку $M(i)$, где i номер вершины графа.

Тогда вершины графа это номера ячеек. Ребро исходящее из вершины соответствует отображению из соответствующей ячейки в другую ячейку. Номер полученной ячейки задает конечную точку ребра. Так как мы не можем отобразить каждую точку в области, мы отображаем k равномерно распределенных точек каждой ячейки. Это число задается пользователем.

По теореме 5.1 из источника [1]: Пусть P(d) - это окрестность равная объединению всех ячеек соответствующих возвратным вершинам графа, где d - это длина стороны ячейки.

$$P(d) = \{\cup\ M(i),\ i - \text{возратная}\}$$

Тогда цепно-рекуррентное множество совпадает с пересечением множеств P(d) по формуле:

$$Q = \bigcap_{d>0} P(d)$$

По теореме 5.2 из источника [1]: При уменьшении размера ячейки новая окрестность оказывается вложена в старую. Из этого следует то, что уменьшение диаметра ячеек приводит к меньшему размеру окрестности. Таким образом, последовательность окрестностей монотонно убывает и сходится к цепно-рекуррентному множеству по формуле:

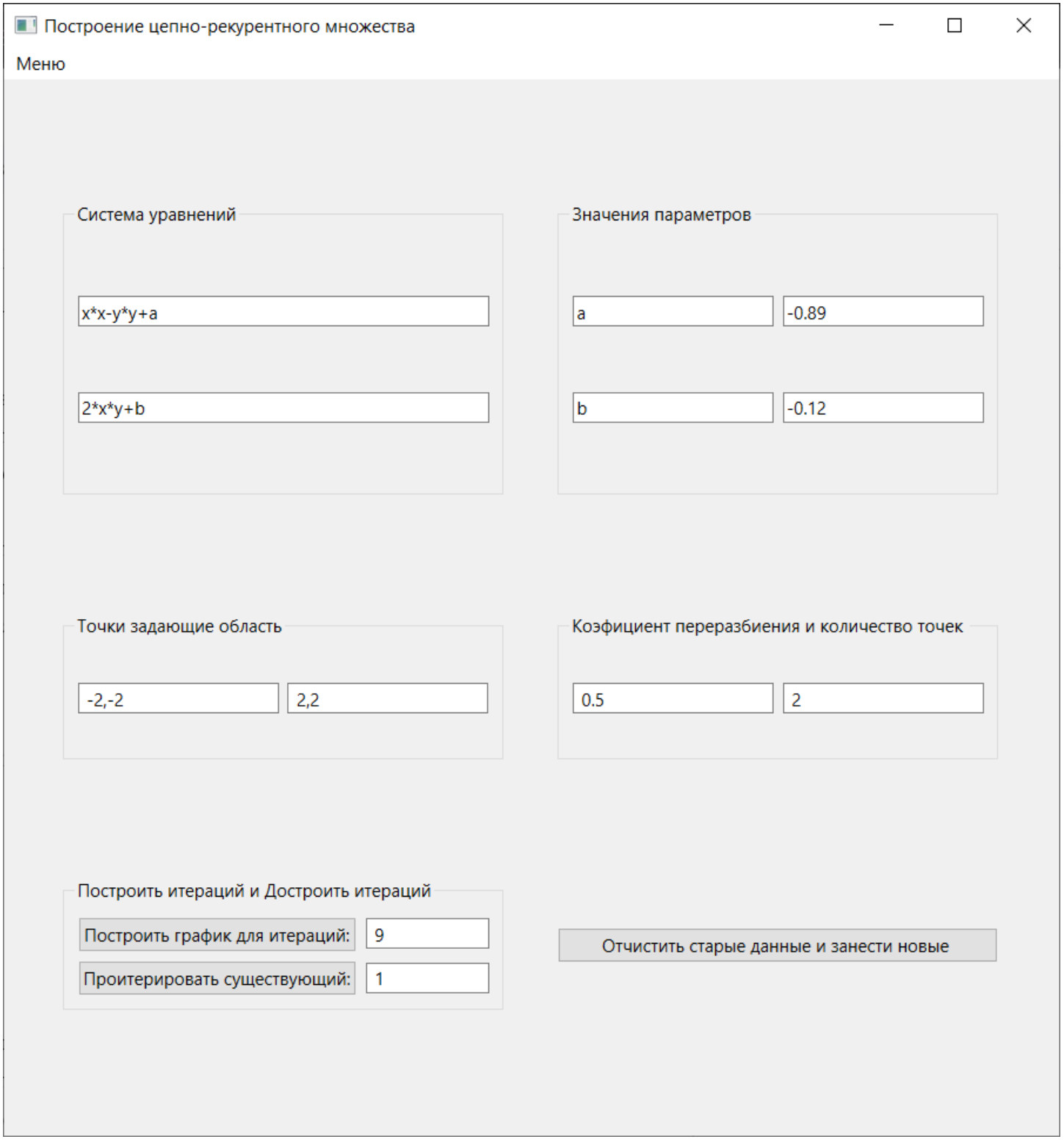
$$\lim_{k\to\infty} P_k = \bigcap_k P_k = Q.$$

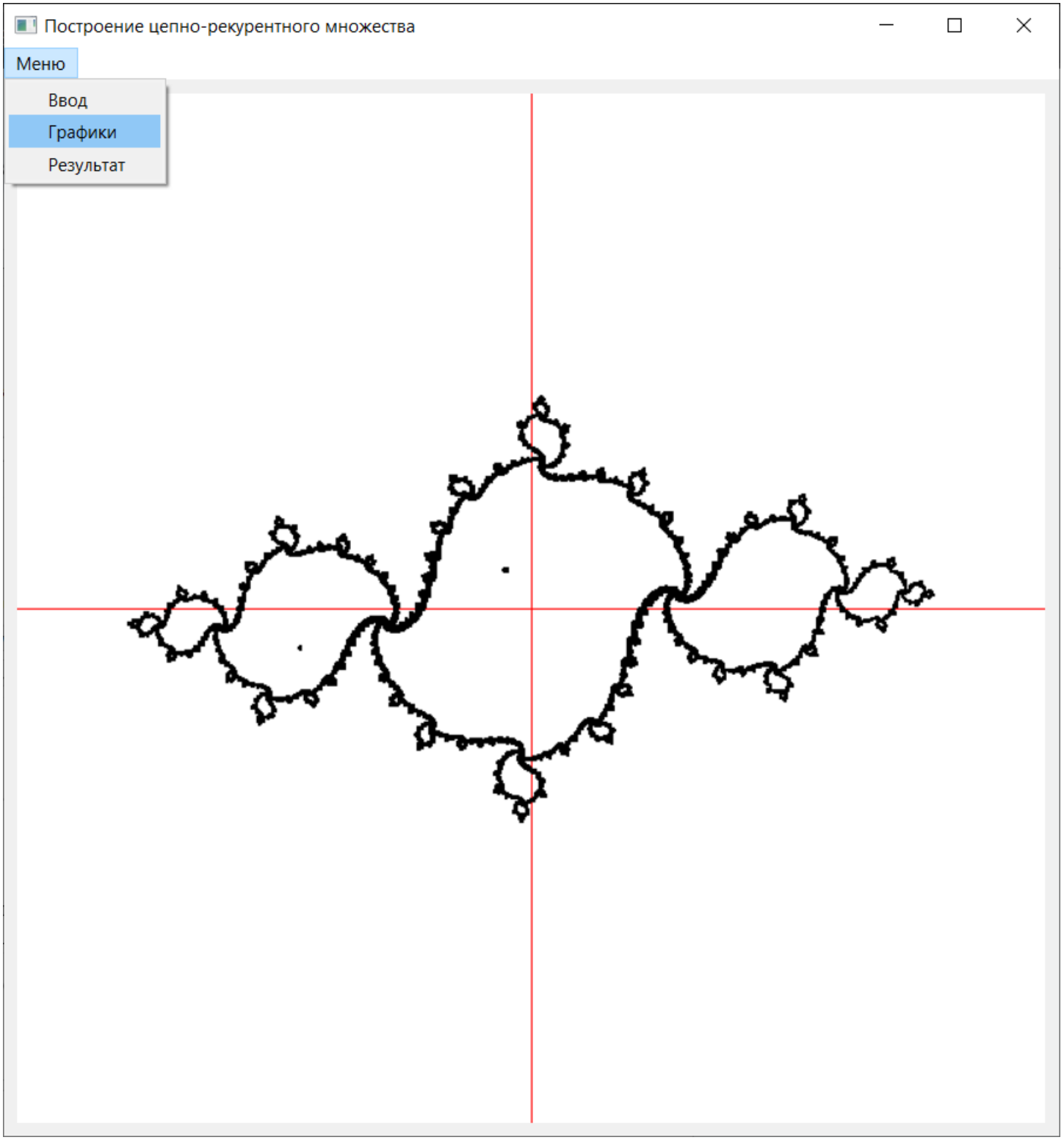
Для подсчета номера ячейки полученного после отображения точки области применяем формулу: $n = [(x - x_{min}) / h_x] + 1 + [(y_{min} - y) / h_y] * [(x_{max} - x_{min}) / h_x]$

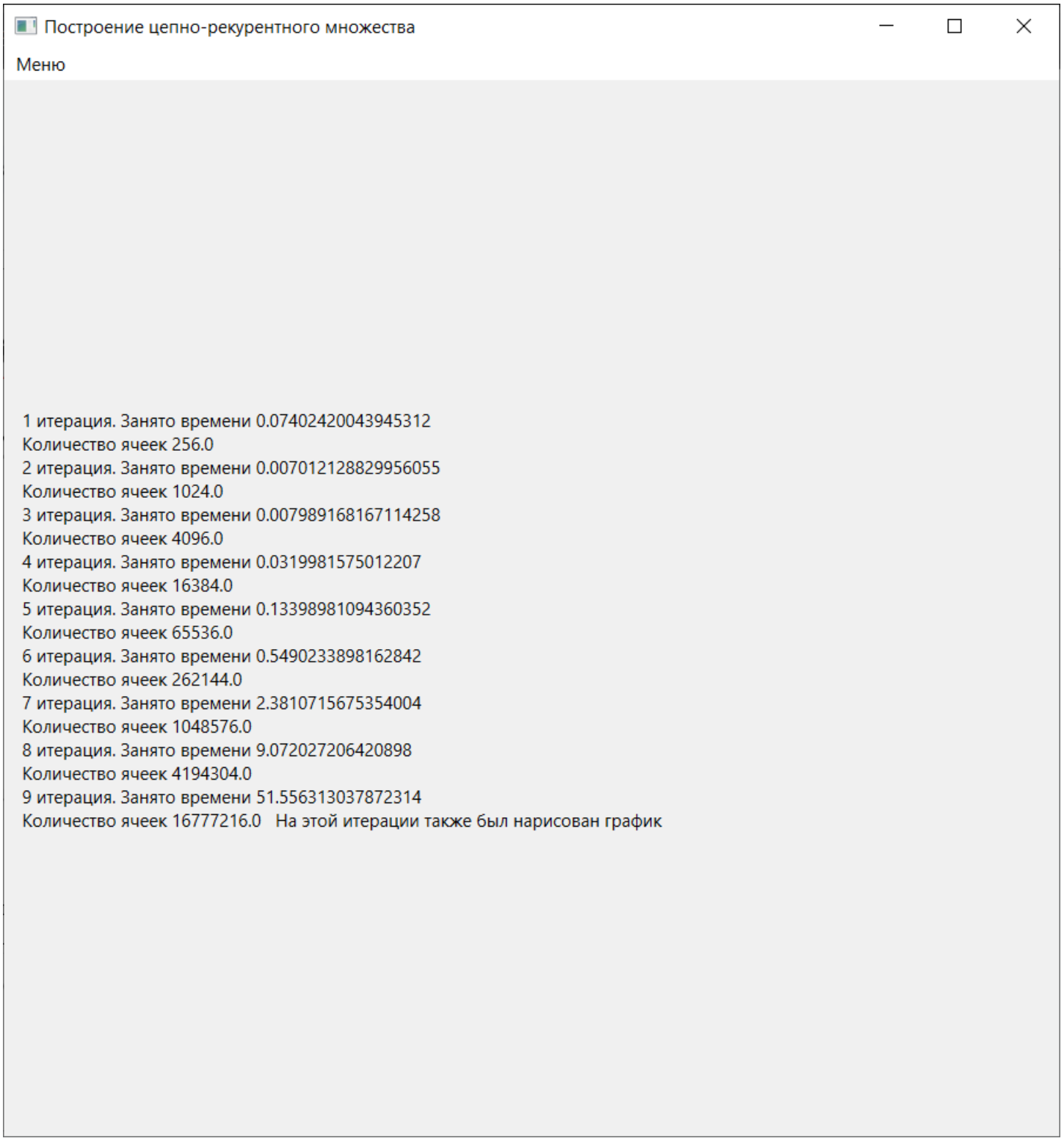
Для нахождения сильно связанных вершин графа была использована функция strongly_connected_components из библиотеки networkx для языка программирования python.

Так как разбиение на необходимое количество ячеек сразу приводит к применению сложного алгоритма к огромному количеству ячеек, мы используем алгоритм локализации цепно-рекуррентного множества. Для этого мы разбиваем область на малое количество ячеек и ищем сильно связанные вершины графа. Затем мы отбрасываем остальные вершины, для этого мы создаем список ячеек с которыми необходимо продолжить работу. Мы разбиваем область на ячейки размером меньше. Далее мы применяем алгоритм только для ячеек из списка. Алгоритм продолжает работу до момента достижения желаемого размера ячеек.

Интерфейс программы




Построение цепно-рекурентного множества
Меню
Ввод
Графики
Результат

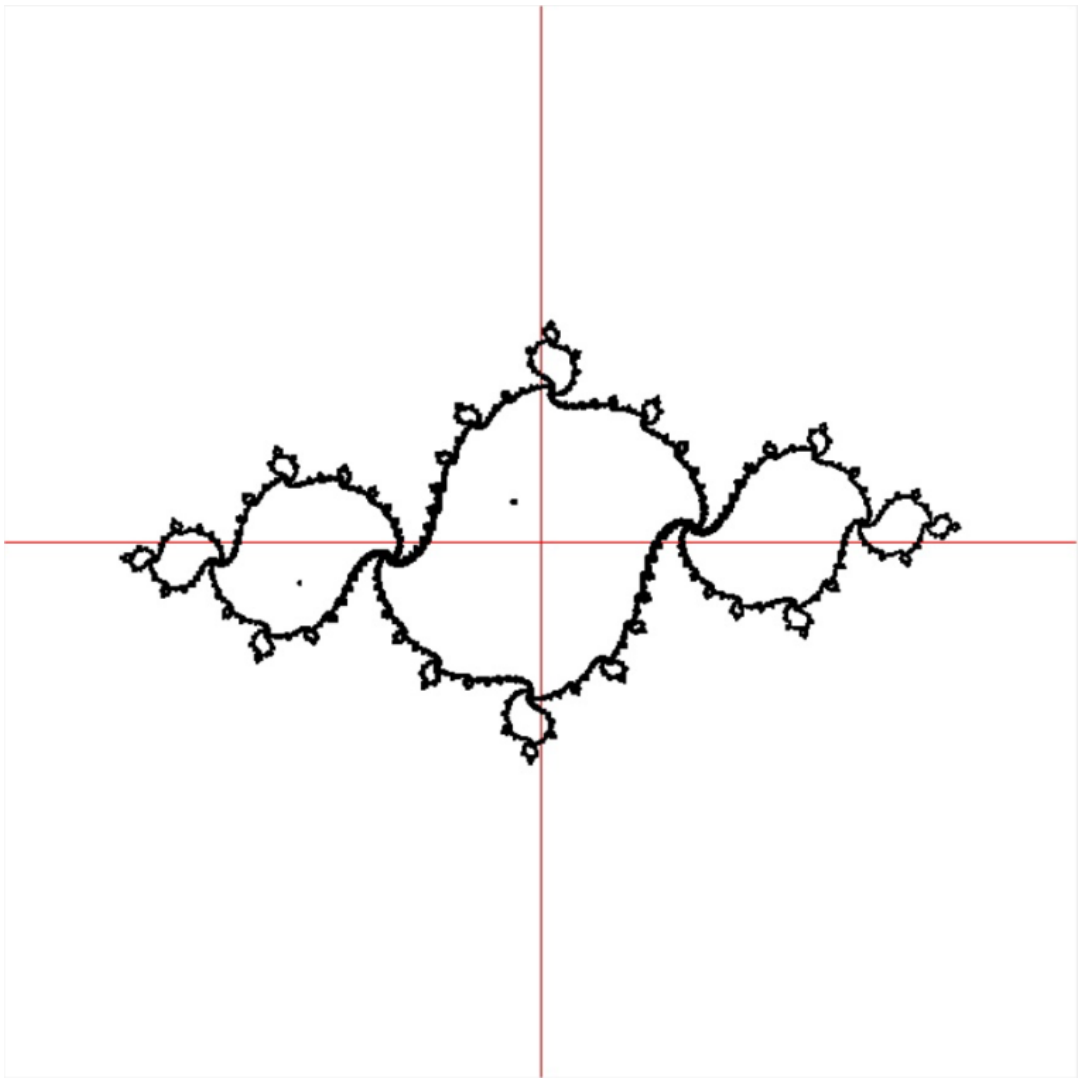

Построение цепно-рекурентного множества
Меню
1 итерация. Занято времени 0.07402420043945312
Количество ячеек 256.0
2 итерация. Занято времени 0.007012128829956055
Количество ячеек 1024.0
3 итерация. Занято времени 0.007989168167114258
Количество ячеек 4096.0
4 итерация. Занято времени 0.0319981575012207
Количество ячеек 16384.0
5 итерация. Занято времени 0.13398981094360352
Количество ячеек 65536.0
6 итерация. Занято времени 0.5490233898162842
Количество ячеек 262144.0
7 итерация. Занято времени 2.3810715675354004
Количество ячеек 1048576.0
8 итерация. Занято времени 9.072027206420898
Количество ячеек 4194304.0
9 итерация. Занято времени 51.556313037872314
Количество ячеек 16777216.0 На этой итерации также был нарисован график

Результаты

Программа написана на языке Python. Для создания интерфейса использовалась библиотека PySide6. Для расчетов использовались библиотеки networkx, math, time.

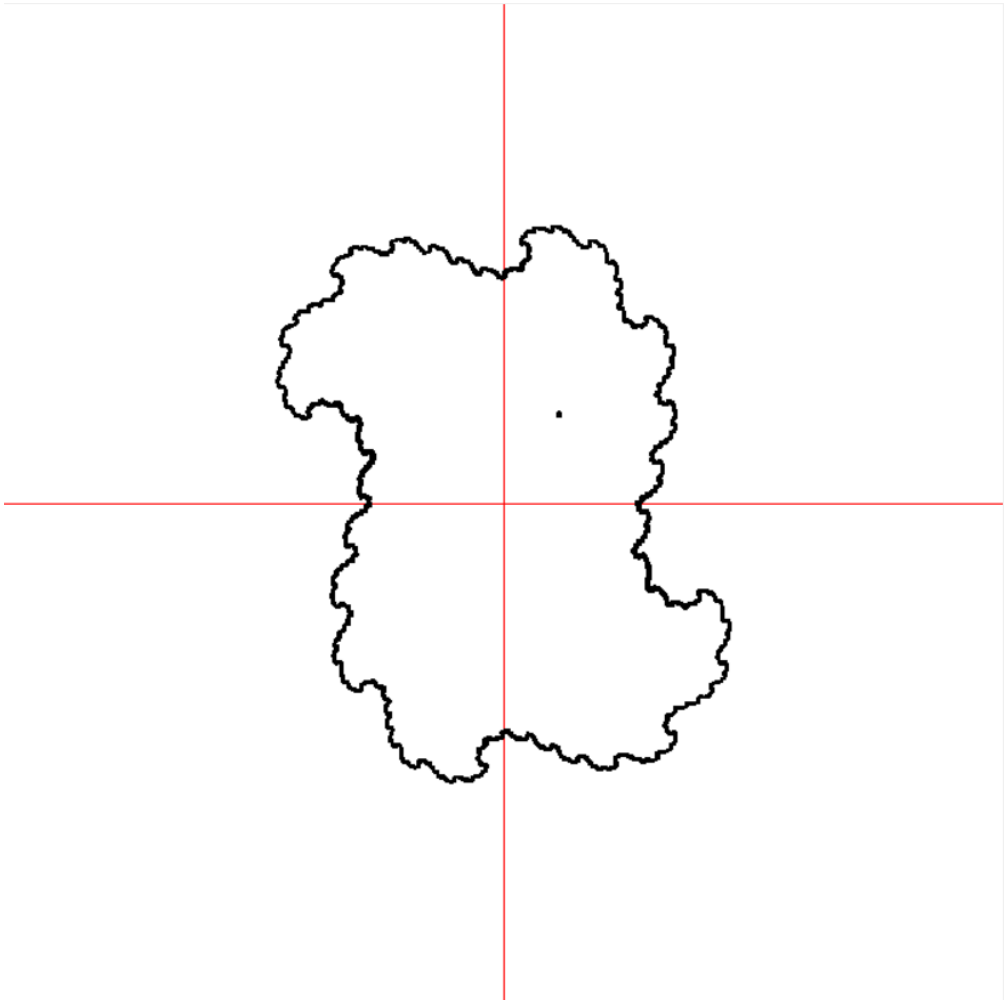
Для всех измерений результатов: область задавалась [-2, -2]x[2, 2], на каждой итерации длина ребра ячейки делилась на два, изначальная длина ребра была равна 0.5 .

Для параметров a = -0.89, b = -0.12 . Для 4 отображаемых точек в виде сетки 2 на 2 из ячейки. Девятая итерация заняла 52 секунды. Всего программа исполнялась немного больше минуты. Количество ячеек на финальном изображении равнялось 16777816. Диаметр ячейки равен 0.00390625.



Для параметров a = 0.3, b = 0.2 . Для 16 отображаемых точек в виде сетки 4 на 4 из ячейки. Девятая итерация заняла 110 секунд. Всего программа исполнялась около двух

минут.. Количество ячеек на финальном изображении равнялось 16777816. Диаметр ячейки равен 0.00390625.



Использованная литература

1. “Введение в символический анализ динамических систем” Г.С.Осипенко, Н.Б.Ампилова.
2. https://ru.wikipedia.org/wiki/Компонента_сильной_связности
3. https://cyberleninka.ru/article/n/otsenka-pokazateley-lyapunova-metodamisimvolicheskogo-analiza
4. https://zetcode.com/gui/pysidetutorial/drawing/
5. https://github.com/Zenoro/ODU-solutions
6. https://www.freecodecamp.org/news/lambda-expressions-in-python/
7. https://www.geeksforgeeks.org/topological-sorting/
8. https://stackoverflow.com/questions/17200117/how-to-get-the-object-name-from-within-the-class
9. https://srinikom.github.io/pyside-docs/PySide/QtCore/QRectF.html
10. https://en.wikipedia.org/wiki/Coordinate_system
11. https://stackoverflow.com/questions/60918473/how-do-i-convert-pixel-screen-coordinates-to-cartesian-coordinates
12. https://www.pythonguis.com/tutorials/pyside6-plotting-pyqtgraph/
13. https://stackoverflow.com/questions/17200117/how-to-get-the-object-name-from-within-the-class