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)

ОТЧЁТ

по вычислительной задаче №2

«Построение цепно-рекуррентного множества для отображения Дуффинга»

Работу выполнил:
Студент группы ПМ-401
Воронец Владимир Олегович

Руководитель: профессор кафедры прикладной математики и информатики
Осипенко Георгий Сергеевич

Севастополь, 2023

ОГЛАВЛЕНИЕ

ПОСТАНОВКА ЗАДАЧИ

Изобразить в графическом виде локализацию цепно-рекуррентного множества для отображения Пуанкаре, порождаемого уравнением Дуффинга перемещением за период t.

$$\begin{cases} \dot{x} = y \\ \dot{y} = x - x^3 - \delta y + \gamma \cos \omega t \end{cases}$$

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

Рассмотрим задачу Коши для системы обыкновенных дифференциальных уравнений первого порядка, применяя метод Рунге-Кутта.

Численное решение уравнения с помощью метода Рунге-Кутта задаётся формулами [1,2]:

$$f(t, x, y) = x - x^3 - \delta y + \gamma \cos \omega t$$

$$g(t, x, y) = y$$

$$k_1 = hf(t_0, x_0, y_0)$$

$$q_1 = hg(t_0, x_0, y_0)$$

$$k_2 = hf(t_0 + \frac{h}{2}, x_0 + \frac{q_1}{2}, y_0 + \frac{k_1}{2})$$

$$q_2 = hg(t_0 + \frac{h}{2}, x_0 + \frac{q_1}{2}, y_0 + \frac{k_1}{2})$$

$$k_3 = hf(t_0 + \frac{h}{2}, x_0 + \frac{q_2}{2}, y_0 + \frac{k_2}{2})$$

$$q_3 = hg(t_0 + \frac{h}{2}, x_0 + \frac{q_2}{2}, y_0 + \frac{k_2}{2})$$

$$k_4 = hf(t_0 + h, x_0 + q_3, y_0 + k_3)$$

$$q_4 = hg(t_0 + h, x_0 + q_3, y_0 + k_3)$$

$$x_1 = y_0 + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

$$y_1 = y_0 + \frac{h}{6}(q_1 + 2q_2 + 2q_3 + q_4)$$

$$t_{1=}t_0 + h$$

где t_0, x_0, y_0 – заданные начальные значения для задачи Коши,
d – шаг изменения времени, определяет точность метода

Чтобы построить цепно-рекуррентное множество для данной динамической системы, необходимо построить символический образ системы, а затем выделить в получившемся графе компоненты сильной связности. Те компоненты, в которые входит по одной вершине, далее не учитываются. Все остальные вершины (входящие в остальные компоненты) являются возвратными и учитываются. Совокупность ячеек, соответствующих данным вершинам, составит окрестность искомого цепно-рекуррентного множества. Чем меньше диаметр ячеек, тем меньше окрестность и тем точнее приближается цепно-рекуррентное множество. Предлагается вначале построить цепно-рекуррентное множество для начального разбиения на 4 ячейки, а затем дробить диаметр пополам n раз и повторять все процедуры, за исключением того, что ячейки, на каком-то этапе не попавшие в окрестность цепно-рекуррентного множества, на всех последующих этапах не рассматриваются. [1]

РЕШЕНИЕ ПОСТАВЛЕННОЙ ЗАДАЧИ

Решение данной задачи заключается в применении алгоритма:

1. Строим исходное покрытие C компакта M. Находим символический образ G данного отображения. Проверяем вхождение полученных номеров ячеек в файл, состоящий из номеров ячеек, принадлежащий рассматриваемой области (если таковой имеется).

2. Выделяем на графе G возвратные вершины $\{i_k\}$ с помощью алгоритма Тарьяна.

3. С помощью метода построения прямоугольников, строим возвратные ячейки на координатной плоскости.

4. Уменьшаем шаг h в два раза.

5. Производим дробление имеющихся вершин графа в соответствии с изменением шага. Заносим новые номера ячеек в файл.

6. Увеличиваем номер итерации.

Выполняем алгоритм до тех пор, пока номер итерации не будет равен данному.

КОМПЬЮТЕРНАЯ РЕАЛИЗАЦИЯ

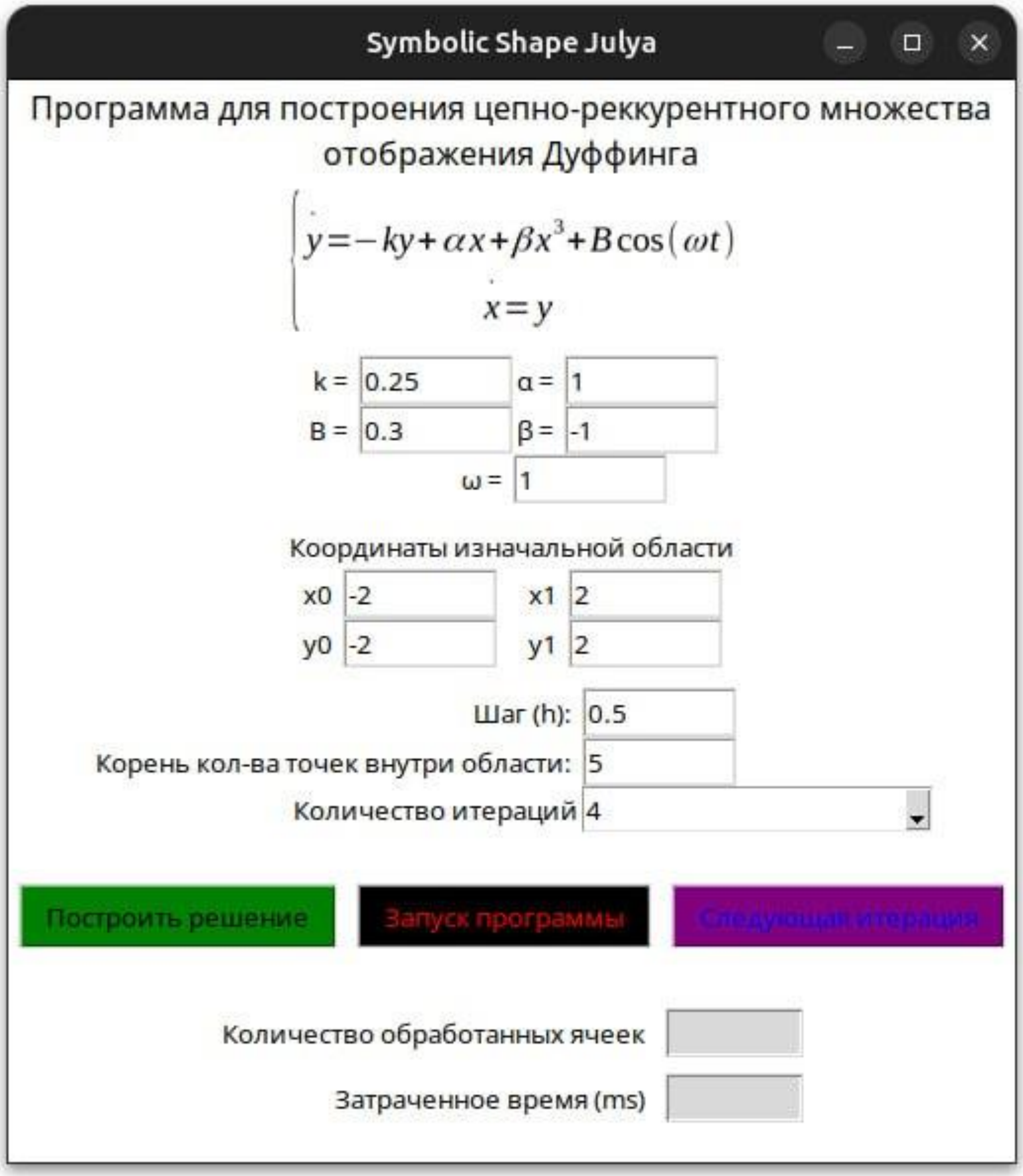

Рисунок 1: Пользовательский интерфейс программы

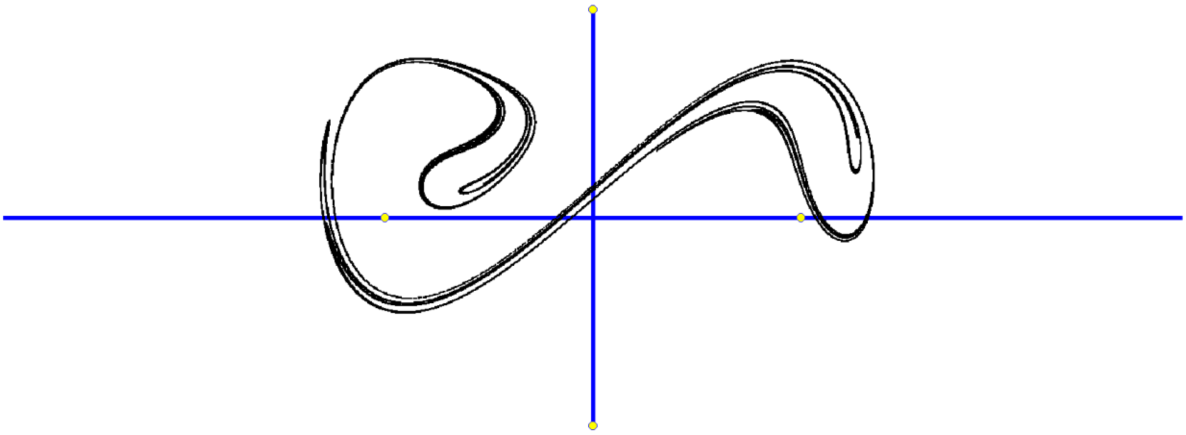

Рисунок 2: Пример отработки программы при семи итерациях, области [-2; 2] x [-2; 2]*, шаге* $h=0.5$ *и параметрах* $\alpha=1$, $\beta=-1$, $\omega=1$, $k=0.25$, $B=0.3$

ХАРАКТЕРИСТИКА ПРОГРАММЫ

Время выполнения программы в приведенном примере: 54.63 секунды для подсчета ячеек и 0.43 секунд для графического отображения. Было обработано 1036242 ячеек.

Было использовано 1 гигабайт памяти компьютера при подсчете ячеек и 31 мегабайт при графическом построении множества.

Нагрузка на процессор (AMD Ryzen 3 3200U) доходила до 90% при подсчете ячеек и около 15% при графическом отображении.

Программа была написана самостоятельно на двух языках программирования: Python3 [2] с использованием встроенной графической библиотеки PIL для создания изображения графического отображения ячеек и библиотеки для создания оконных приложений Tkinter [3]; C++ использовался для выполнения операций алгоритма решения, что помогает нагрузить процессор на максимум и выполнять подсчет ячеек быстрее. Программа ориентирована на UNIX-подобных системах, имеются статические пути, которые необходимо изменить перед запуском программы. Для работы необходимо предварительно установить все вышеперечисленные библиотеки.

СПИСОК ЛИТЕРАТУРЫ

1. Осипенко Г.С., Ампилова Н.Б. Введение в символический анализ динамических систем: – СПб.: Издательство Санкт-Петербургского университета, 2004. 240 с.
2. http://www.mathprofi.ru/metody_eilera_i_runge_kutty.html
3. https://docs.python.org/3/library/tkinter.html#module-tkinter