Hello everyone!

План работы

1 Вступление

1.1 Цели и задачи работы

Основной целью работы является создание приложения на языках C++ и Python с применением библиотеки Manim. Суть приложения заключается в генерации 3D-изображений дискретных динамических систем, заданных на сфере, из соответствующих им трёхцветных графов, заранее проверенных программой на корректность. Алгоритмическая часть приложения написана на C++, так как этот язык в разы быстрее чем язык Python, а на языке Python написана визуальная составляющая программы, так как язык содержит множество удобных для этого библиотек. Помимо этого поставлены следующие подзадачи: 1) Разобраться в связи динамических систем (КАКИХ?) и трёхцветных графов. 2) Научиться писать Unit-тесты на языке C++, необходимые для стабильной работы программы при её изменении в дальнейшем 3) Придумать алгоритм генерации трёхцветных графов с заданными параметрами: число Эйлера и число сёдел динамической системы.

1.2 Актуальность работы

Программа, написанная в результате работы, будет полезна начинающим научным сотрудникам или обычным студентам, желающим разобраться в динамических системах на сфере, так как поможет им быстрее визуализировать эти динамические системы. Помимо этого, отдельные функции из алгоритмический части могут быть полезны другим исследователям трёхцветных графов в написании рабочих программ.

2 Теоретическая часть

ОТРЕДАЧИТЬ ВСЁ И ПЕРЕПИСАТЬ СВОИМИ СЛОВАМИ

Трёхцветный граф

В этой главе будет представлено построение трёхцветного графа по градиентноподобному каскаду на поверхности. Стоит отметить, что на языке трёхцветных графов получена полная топологическая классификация градиентно-подобных каскадов на поверхностях.

Трёхцветный граф как полный топологический инвариант диффеоморфизма на поверхности

Определение 1. *Диффеоморфизм $f : M^n \to M^n$, заданный на гладком замкнутом n-многообразии, называется диффеоморфизмом Морса-Смейла, если:*
1) неблуждающее множество Ω_f гиперболично и конечно (т.е. состоит из конечного чила периодических точек, для которых модули собственных значений матрицы Якоби не равны единице);
2) для любых периодических точек p, q устойчивое многообразие W^s_p и неустойчивое многообразие W^u_q либо не пересекаются, либо трансверсальны в каждой точке пересечения.

Пусть $f : M^n \to M^n$ - диффеоморфизм Морса-Смейла, тогда периодические точки называются источниками, если неустойчивое многообразие W^u_q имеет размерность n, стоками, если 0, и сёдлами при остальных.
Далее скажем, что для любой периодической точки p диффеоморфизма f компоненты связности $W^s_p \backslash p$ ($W^u_p \backslash p$) называются её устойчивыми (неустойчивыми) сепаратрисами.
Рассмотрим класс диффеоморфизмов на поверхности M^2, тогда диффеоморфизм Морса-Смейла называется градиентно-подобным, если $W^s_p \cap W^u_p = \emptyset$ для любых различных седловых точек p, q.
Удалим из поверхности M^2 замыкание объединения устойчивых и неустойчивых многообразий седловых точек f и получим множество $M' = M^2 \backslash (W^u_{\Omega^0_f} \cup W^u_{\Omega^1_f} \cup W^s_{\Omega^1_f} \cup W^s_{\Omega^2_f})$.
M' является объединением ячеек, гомеоморфных открытому двумерному диску, граница которых имеет один из 3-х следующих видов: (картинка с ячейками)
Пусть A - ячейка из M'

Определение 2. *Конечным графом называется упорядоченная пара (B,E), для которой выполнены следующие условия:*
1) B - непустое конечное множество вершин
2) E - множество пар вершин, называемых рёбрами

Определение 3. *Если граф содержит ребро e = (a,b), то каждую из вершин a, b называют инцидентной ребру e и говорят, что вершины a* *и b соединены ребром e.*

Определение 4. *Путём в графе называют конечную последовательность его вершин и рёбер вида:* $b_0, (b_0, b_1), b_1, \ldots, b_{i-1}, (b_{i-1}, b_i), b_i, \ldots, b_{k-1}, (b_{k-1}, b_k), b_k, k >= 1$*. Число k называется длиной пути, оно совпадает с числом входящих в него рёбер.*

Определение 5. *Граф называют связным, если любые две его вершины можно соединить путём.*

Определение 6. *Циклом длины* $k \in \mathbb{N}$ *в графе называют конечное подмножество его вершин и рёбер вида* $\{b_0, (b_0, b_1), b_1, \ldots, b_{i-1}, (b_{i-1}, b_i), b_i, \ldots, b_{k-1}, (b_{k-1}, b_0)\}$*. Простым циклом называют цикл, у которого все вершины и рёбра попарно различны.*

Определение 7. *Граф T называется трёхцветным графом, если:*
1) множество рёбер графа T является объединением трёх подмножеств, каждое из которых состоит из трёх рёбер одного и того же определенного цвета (цвета рёбер из разных подмножеств не совпадают, будем обозначать эти цвета буквами s, t, u, а рёбра для краткости будем называть s-, t-, u-рёбрами);
2) каждая вершина графа T инцидентна в точности трём рёбрам различных цветов;
3) граф не содержит циклов длины 1.

Определение 8. *Простой цикл трёхцветного графа T назовём двухцветным циклом типа su, tu или st, если он содержит рёбра в точности двух цветов s и u, t и u, s и t соответственно. Непосредственно из определения трёхцветного графа следует, что длина любого двухцветного цикла является чётным числом (так как цвета рёбер строго чередуются), а отношение на множестве вершин, состоящее в принадлежности двёхцветному циклу определённого типа, является отношением эквивалентности.*

Определение 9. *Построим трёхцветный граф* T_f*, соответствующий диффеоморфизму* $f \in G$*, следующим образом:*
1) вершины графа T_f *взаимно однозначно соответствуют треугольным*

областям множества Δ;
2) две вершины графа инцидентны ребру цвета s, t, u, если соответствующие этим вершинам треугольные области имеют общую s-, t- или u-кривую

Граф T_f полностью удовлетворяет определению трёхцветного графа

Теорема 1. *Теорема 1. Для того чтобы диффеоморфизмы f, f' из класса G были топологически сопряжены, необходимо и достаточно, чтобы их графы (T_f, P_f) и $(T_{f'}, P_{f'})$ были изоморфны*

Определение 10. *Определение 2. Трёхцветный граф (T,P) назовём допустимым, если он обладает следующими свойствами:*
1) граф T связен;
2) длина любого su-цикла графа T равна 4;
3) автоморфизм P является периодическим.

Лемма 1. *Пусть $f \in G$. Тогда трёхцветный граф (T_f, P_f) является допустимым.*

Теорема 2. *Пусть (T,P) - допустимый трёхцветный граф. Тогда существует диффеоморфизм $f : M^2 \to M^2$ из класса G, граф (T_f, P_f) которого изоморфен графу (T,P). При этом:*
1) эйлерова характеристика поверхности M^2 вычисляется по формуле $X(M^2) = v_0 - v_1 + v_2$, где v_0, v_1, v_2 - число всех tu-, su-, st-циклов графа T соответственно;
2) поверхность M^2 ориентируема тогда и только тогда, когда все циклы графа T имеют чётную длину

3 Алгоритмическая часть

3.1 Структура программы

Программа запускается как скрипт с графическим интерфейсом на языке Python. В графическом интерфейсе в окно для ввода трёхцветного графа вводится трёхцветный граф, при этом его нельзя ввести некорректно, однако проверка графа на то, действительно ли он является корректным трёхцветным графом, присутствует в алгоритмической части.

Далее, при нажатии кнопки "Показать динамическую систему появляется зацикленное видео с динамической системой на сфере, показанное в отдельном окне. Программа на языке Python передаёт введённую информацию в алгоритмическую часть на языке C++. Алгоритмическая часть, в свою очередь, возвращает в часть с графическим интерфейсом координаты сепаратрис. Далее программа на Python генерирует по заданным координатам конца и начала сепаратрис 3D-картинку, а затем, после рендеринга в библиотеке Manim, показывает её пользователю.

3.2 Проверка введённого трёхцветного графа на корректность

Проверяет граф на корректность функция $is_acceptable$, которая принимает на вход заданный граф и возвращает булевое значение: True, если граф является корректным трёхцветным графом, и False, если граф таковым не является. Согласно (ЧЕМУ?) граф является коррестым трёхцветным графом, если: 1) граф трёхцветный и неориентированный 2) граф связный 3) все SU-циклы в графе имеют длину равную четырём Функция $is\_acceptable$ для проверки пункта 1 вызывает функцию $is\_3\_colored\_and\_non\_o$ которая...(ДОПИСАТЬ) Для проверки пункта 2 функция вызывает функцию $is_connected$, которая...(ДОПИСАТЬ) Для проверки пункта 3 функция вызывает функцию $find_cycles$ с переданными в неё графом, литералом 's' и литералом 'u', которые отвечают за то, какого цвета циклы надо найти. Функция $find_cycles$ возвращает вектор, состоящий из циклов, каждый цикл представляет собой последовательность номеров вершин цикла, так же последовательно соединённых между собой в самом цикле. Далее функция $find_cycles$ проверяет, имеют ли все SU-циклы в графе длину 4. Если все условия выполнены, функция возвращает True, в противном случае возвращает False.

3.3 Проверка поверхности на ориентируемость

Для проверки графа на ориентируемость используется теория о базе циклов (ССЫЛКА НА ТЕОРИЮ). Базой циклов неориентированного графа является такой набор циклов, путём соединения или вычитания которых могут получиться все остальные циклы. Для нахождения базы циклов необходимо построить из графа дерево, а далее путём соединения дерева и рёбер графа, которые в дерево не попали, по одному найти все циклы из базы. По (ОТКУДА?) поверхность ориентируема тогда и только тогда, когда все циклы графа имеют четную длину. Очевидно, что при вычитании или сложении циклов четной длины получится цикл чётной длины, то есть на чётность достаточно проверить всего лишь циклы из базы циклов. Этот алгоритм реализован в функции $is_oriented_surface$, которая возвращает True, если поверхность ориентируема, и False, если поверхность неориентируема.

3.4 Генератор графов по заданному числу Эйлера и числу сёдел

Тут надо рисовать

3.5 Поиск соседних неподвижных точек

Тут тяжко

3.6 Нахождение сепаратрис

Тут очень тяжко

3.7 Unit-тестирование

4 Ссылка на репозиторий в Github

$https://github.com/dan1lka257/graphs_and_algorithms/tree/main$

5 Список литературы

капкаева, починка какая-нибудь ссылка на базу циклов