Образовательная программа
"Фундаментальная математика"

3 июня 2024 г.

Генерация 3D-изображений дискретных динамических систем, заданных на поверхностях, из трёхцветных графов

Курсовая работа

Турсунов Данил Вячеславович

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 градиентно-подобных каскадов, 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.

Теоретическая часть
Часть 1

Начнём введение в теоретическую часть с определения диффеоморфизма Морса-Смейа и алгоритма построения трёхцветного графа, заданного на поверхности и, в частности, на сфере.

Определение

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_p^s и неустойчивое многообразие W_q^u 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 размерность равна 0, и сёдлами при остальных значениях размерности.
Далее скажем, что для любой периодической точки p диффеоморфизма f компоненты связности $W^s_p(p)$ и $(W^u_p(p))$ называются её устойчивыми или неустойчивыми сепаратрисами соответственно.

Теоретическая часть
Часть 3

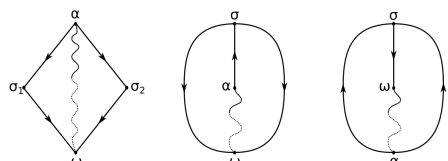
Введём более узкое определение: рассмотрим класс диффеоморфизмов на поверхности M^2, тогда диффеоморфизм Морса-Смейла называется градиентно-подобным, если пересечение W_p^s и W_p^u равно пустому множеству для любых различных седловых точек p, q.
В дальнейшем в работе будут рассматриваться исключительно граентно-подобные диффеоморфизмы, заданные на поверхности M^2. Класс градиетно-подобных диффеоморфизмов, заданных на поверхности M^2, обозначим G.

Введём более узкое определение: рассмотрим класс диффеоморфизмов на поверхности M^2, тогда диффеоморфизм Морса-Смейла называется градиентно-подобным, если пересечение W_p^s и W_p^u равно пустому множеству для любых различных седловых точек p, q.
В дальнейшем в работе будут рассматриваться исключительно граентно-подобные диффеоморфизмы, заданные на поверхности M^2. Класс градиетно-подобных диффеоморфизмов, заданных на поверхности M^2, обозначим G.

Теоретическая часть

Часть 5

Удалим из поверхности M^2 замыкание объединения устойчивых и неустойчивых многообразий седловых точек f и получим множество M'. M' является объединением ячеек, гомеоморфных открытому двумерному диску, граница которых имеет один из 3-х видов, показанных на рис. 1.



Пусть A - ячейка из M', α и ω - источник и сток, входящие в её границу. Кривую $\tau \in A$, началом и концом которой являются α и ω, будем называть $t-$кривой. Через T обозначим множество $t-$кривых, взятых по одной из каждой ячейки. Разобьём каждую ячейку этой кривой на 2 области, компоненты связности $M_\Delta = M' \setminus T$ назовём треугольными областями. В границу каждой треугольной области входят 3 периодические точки: источник, сток и седло, а также устойчивая сепаратриса, неусточивая сепаратриса и кривая τ. В дальнейшем будем называть их $s-$кривой, $u-$кривой и $t-$кривой соответственно. Замыкание каждой из этих кривых будем называть стороной треугольной области. Скажем, что сторона является общей стороной для двух треугольных областей, если она принадлежит замыканиям этих треугольных областей.

Определение

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

Простой цикл трёхцветного графа T назовём двухцветным su-, tu- или st-циклом, 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 двухцветному циклу определённого типа, является отношением эквивалентности, то есть каждая отдельно взятая вершина лежит в точности в одном *su*—, одном *tu*— и одном *st*—цикле.

Определение

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 .

Отметим, что построенный граф T_f полностью удовлетворяет определению трёхцветного графа.

Определение

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.

Отметим, что трёхцветный граф, построенный по градиентно-подобному каскаду на поверхности, является допустимым. Следующие теоремы сформулированы для трёхцветных графов, оснащённых автоморфизмом, однако, во избежание излишей громоздкости теоретической части, данная часть повествования была опущена в курсовой работе.

Теорема

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.

Теорема

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.

Алгоритмическая часть

Часть 1

Программа состоит из 2 частей: алгоритмической и графической, каждая из частей запускается отдельно и независимо друг от друга. Изначально запускается алгоритмическая часть на языке C++, туда вводится корректный трёхцветный граф, программа его обрабатывает и выдаёт в отдельном файле координаты сепаратрис. Далее запускается графическая часть программы, написанная на языке Python. Она считывает координаты из файла и генерирует по заданным координатам конца и начала сепаратрис 3D-изображение, а затем, после рендеринга в библиотеке Manim, показывает её пользователю.

Алгоритмическая часть
Часть 2

На языке C++ написаны следующие функции (краткий список):

- Проверка введённого трёхцветного графа на корректность
  - Поиск в ширину
  - Поиск двухцветных циклов
  - Проверка графа на трёхцветность, неориентируемость и отсутствие петель
- Проверка поверхности на ориентируемость при помощи базы циклов
- Генератор графов по заданной характеристике Эйлера и числу сёдел
- Поиск координат начала и конца сепаратрис для отображения на сферу
  - Поиск соседних неподвижных точек
  - Проверка сепаратрис на пересечение

База циклов

Часть 1

Для построения алгоритма, проверяющего поверхность, на которой задан градиентно-подобный каскад, по трёхцветному графу, потребуется пункт 2 теоремы 2, который гласит о том, что поверхности ориентируема тогда и только тогда, когда все циклы графа имеют чётную длину.
Сделать вывод о четной длине всех циклов графа можно найдя хотя бы один цикл нечётной длины. Для нахождения такого цикла потребуется небольшой экскурс в теорию, связанную с базой циклов и алгоритмом её нахождения.

База циклов

Часть 2

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 так называемое «переплетающееся дерево» (spanning tree), то есть просто выбрать какую-нибудь вершину за корень дерева, а потом идти по нему уже упомянутым выше алгоритмом поиска в ширину из корневой вершины, при этом вместо поиска расстояний отмечать посещённые вершины, если вершина-сосед не посещена, то ребро, связывающее текущую вершину с ней, добавлять в «переплетающееся дерево», а далее, путём добавления по одному в дерево рёбер изначального графа, которые в дерево не попали, по одному найти все циклы. Эти циклы и будут составлять базу циклов в графе. 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.

База циклов

Часть 3

Алгоритм, находящий «переплетающееся дерево» и сразу проверяющий циклы базы циклов на чётную длину, реализован в функции *is_oriented_surface*, которая возвращает True, если поверхность ориентируема, и False, если поверхность неориентируема.

Тривиальный генератор трёхцветных графов

Часть 1

Для дальнейшей проверки алгоритма на корректность потребуется генерировать корректные трёхцветные графы со всевозможным количеством стоков и источников, такие, что соответствующие им градиентно-подобные каскады расположены на сфере, по заданному числу сёдел σ. Изложенную ниже генерацию можно обобщить для генерации трёхцветных графов, такие, что соответствующие им градиентно-подобные каскады расположены на ориентируемой поверхности, определяемой характеристикой Эйлера e.

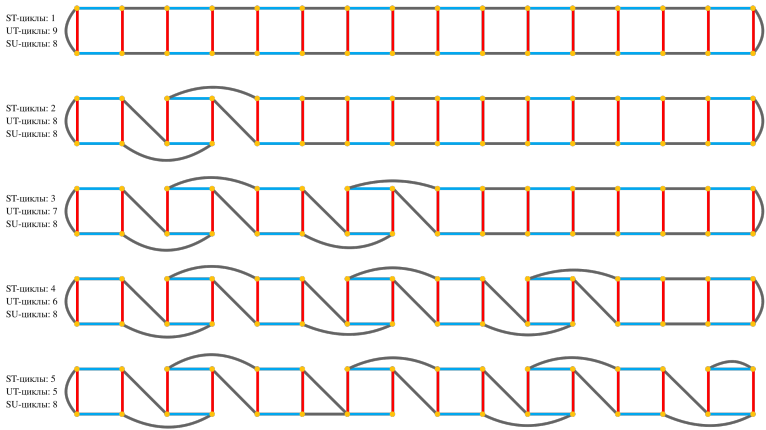
Тривиальный генератор трёхцветных графов

Часть 2

Из определения 10 известно, что длина SU-циклов корректного трёхцветного графа равна 4, причём количество SU-циклов равно числу сёдел σ. Тогда расположим «квадраты» SU-циклов, верхнее и нижнее ребро которых имеют одинаковый цвет для всех «квадратов» (будем считать, что верхние и нижние ребра - s-рёбра), в ряд и будем проводить из каждой вершины рёбра цвета t. Далее будем соединять t-ребром с ближайшей вершиной ближайшего соседнего «квадрата» SU-цикла, кроме первых 2 и последних 2 вершин, которые соединяются между собой соответственно. Получили тривиальный пример графа с числом SU-циклов, равному σ, числом ST-циклов, равному 1, и числом UT-циклов, равному $\sigma + 1$.

Тривиальный генератор трёхцветных графов

Тривиальные графы на сфере



Тривиальный генератор трёхцветных графов

Часть 3

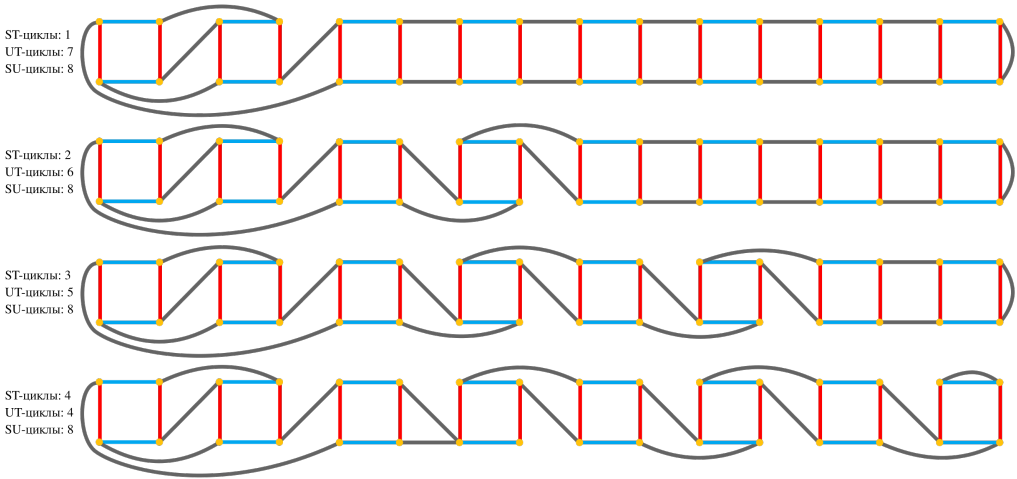
Увеличим количество ST-циклов на 1 и уменьшим количество UT-циклов на 1. Это можно сделать из построенного выше тривиального графа при помощи переподвязок t-циклов. Переподвязывать t-циклы будем следующим способом: возьмём чётный по счёту «квадрат» и перекрасим s-рёбра в u-рёбра и наоборот. Одна такая переподвязка увеличивает количество ST-циклов на 1 и уменьшает количество UT-циклов на 1.

Проводя такие переподвязки на чётных «квадратах» по одной поверх друг друга, сгенерируем циклы, для которых количество ST-циклов принимает значения (с учётом тривиального графа) $\{1, \ \dots, [\sigma/2]\}$, а количество UT-циклов $\{[(\sigma+1)/2], \ \dots, \sigma+1\}$.

Ниже приведены примеры построения трёхцветных графов при $\sigma = 8$ и $e = 0$ и $e = -2$ для рис. 3 и для рис. 4 соответственно.

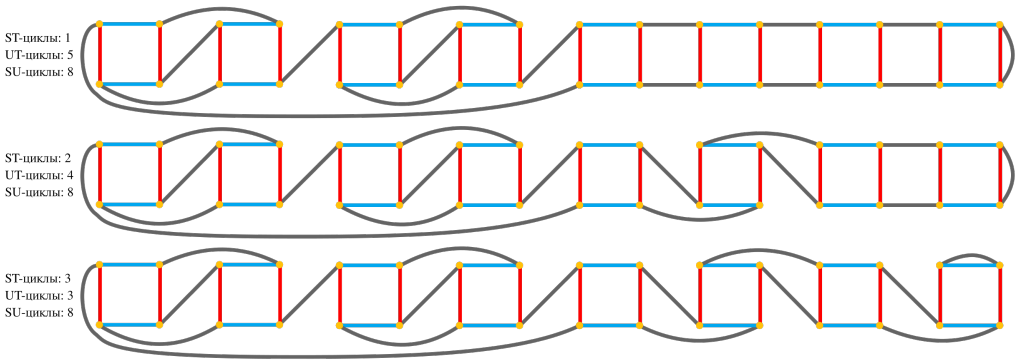
Тривиальный генератор трёхцветных графов

Тривиальные графы на торе



Тривиальный генератор трёхцветных графов

Тривиальные графы на двойном торе



Представим сферу как прямоугольник $[-90, 90]$ x $[0, 360]$, где все точки из отрезка -90 x $[0, 360]$ и из отрезка 90 x $[0, 360]$ отождествлены между собой. Впоследствии при визуализации этот прямоугольник будем отображать на сферу по формуле:

$$\begin{cases} x = r * sin(\psi) * cos(\phi) \\ x = r * sin(\psi) * sin(\phi) \\ x = r * cos(\psi) \end{cases}$$

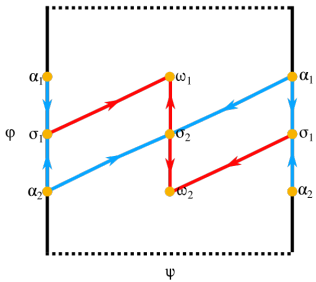
Сепаратрисы будем представлять как пары, состоящие из цвета сепаратрисы, красная или синяя, и вектора координат, содержащего a, $a0$, b, $b0$ и при этом заданной формулой:

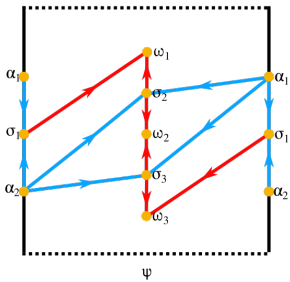
$$\begin{cases} x = a * t + a_0 \\ y = b * t + b_0 \end{cases}$$

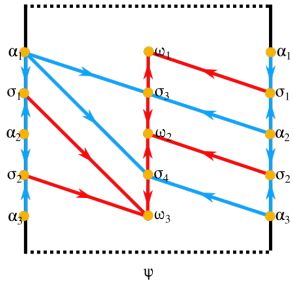
где t $\in [0, 1]$.

Алгоритмическая часть

Представление каскада на сфере







Unit-тестирование

Для стабильной работы программы при изменении в дальнейшем в каталоге под названием *graph_algorithms_unit_test* содержатся файлы, в которых написаны unit-тесты для различных функций. Если при изменении программа будет работать неправильно, то unit-тесты распознают аномалии.

Графическая часть

Часть 1

Графическая часть программы, написанная в файле draw.py, отвечает за генерацию 3D-изображения дискретной динамической системы на сфере.
В файле определён класс DynamicalSystemSphere, который в дальнейшем будет указываться при запуске графической части.
Графическая часть работает по следующему алгоритму:
1) Считывается информация о сепаратрисах, полученная в результате запуска алгоритмической части;
2) Объявляется функция *func_sphere*, задающая параметрически поверхность сферы:

$$\begin{cases} x = r * cos(u) * sin(v) - x_0 \\ y = r * sin(u) * sin(v) - y_0 \\ z = r * cos(v) - z_0 \end{cases}$$

3) Объявляется функция *construct*, которая отвечает непосредственно за генерацию 3D-изображения. При помощи библиотеки Manim создаются поверхность сферы и оси OX, OY и OZ. Далее каждая сепаратриса, разбитая на 3

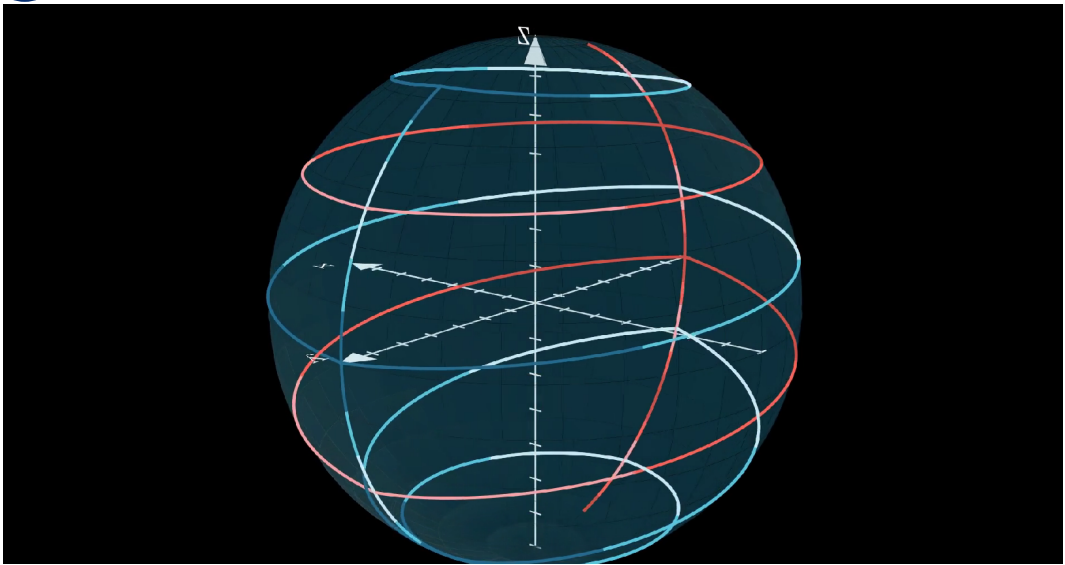
Сепаратриса, заданная параметрически на прямоугольнике значениями a, $a0$, b, $b0$, отображается на поверхность сферы по правилу:

$$\begin{cases} x = cos(pi * (t * b + b_0)/180) * sin(pi * (t * a + a_0)/180) \\ y = cos(pi * (t * b + b_0)/180) * cos(pi * (t * a + a_0)/180) \\ z = sin(pi * (t * b + b_0)/180) \end{cases}$$

где t принадлежит [0, 1] (причём для создания более «говорящей» картинки для различных оттенков сепаратрисы отрезок разбивается на 3 равные части, каждая из которых отображается независимо от остальных).
Для получения объёмного и «говорящего» изображения объявляется полный оборот камеры вокруг сферы.

Запуск и результат работы программы

Для запуска генерации 3D-изображения необходимо:
1) При помощи терминала установить библиотеку Manim на компьютер, если эта библиотека ещё не установлена.
2) Предварительно запустить алгоритмическую часть с введённым в неё корректным трёхцветным графом.
3) При помощи терминала перейти в каталог с файлом draw.py.
4) Запустить графическую часть при помощи команды:
manim -pqh draw.py DynamicalSystemSphere - для генерации изображения высокого качества;
manim -pql draw.py DynamicalSystemSphere - для генерации изображения низкого качества.



Ссылка на репозиторий в Github

Ссылка на репозиторий:
https : //github.com/dan1lka257/graphs_and_algorithms/tree/main