

Министерство науки и высшего образования Российской Федерации
Санкт-Петербургский политехнический университет Петра Великого
Физико-механический институт
Высшая школа теоретической механики и математической физики

Работа допущена к защите

Директор ВШТМиМФ,

Д.ф.-м.н., чл.-корр. РАН

_____ А. М. Кривцов

«___» _____ 2023 г

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА БАКАЛАВРА
ПЛОТНЕЙШИЕ ДВОЯКОПЕРИОДИЧЕСКИЕ ПЕРЕПЛЕТЕНИЯ**

По направлению подготовки

01.03.03 Механика и математическое моделирование

профиль

01.03.03_01 Механика и математическое моделирование сред с
микроструктурой

Выполнил студент группы
5030103/90101

Н. А. Мизина

Руководитель
доцент ВШТМиМФ, к.ф.-м.н.

В. Р. Мешков

Санкт-Петербург

2023

**САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ ПЕТРА ВЕЛИКОГО**

Физико-механический институт

Высшая школа теоретической механики и математической физики

УТВЕРЖДАЮ
Директор ВШТМиМФ
А. М. Кривцов

«__» ____ 20__ г.

ЗАДАНИЕ

на выполнение выпускной квалификационной работы

студенту Мизиной Наталье Алексеевне, гр. 5030103/90101

1. Тема работы: Моделирование плотнейших двоякопериодических переплетений
2. Срок сдачи студентом законченной работы: 05.06.2023
3. Исходные данные по работе: актуальные научные публикации по теме работы.
4. Содержание работы (перечень подлежащих разработке вопросов): написание программы для создания матрицы, описывающей начальное зацепление, написание программы для начальной трехмерной реализации зацепления, разработка алгоритма затягивания начального зацепления до максимальной плотности.
5. Перечень графического материала (с указанием обязательных чертежей): не предусмотрено.
6. Консультанты по работе: отсутствуют.
7. Дата выдачи задания: 27.02.2023

Руководитель ВКР: _____ В. Р. Мешков, доцент ВШТМиМФ, к.ф.-м.н.

Задание принял к исполнению: 27.02.2023

Студент: _____ Н. А. Мизина

РЕФЕРАТ

На 57 с., 52 рисунков

ПЛОТНЕЙШИЕ ДВОЯКОПЕРИОДИЧЕСКИЕ ПЕРЕПЛЕТЕНИЯ, ТЕОРИЯ УЗЛОВ, ИНВАРИАНТЫ УЗЛОВ, ДВИЖЕНИЯ РЕЙДЕМЕЙСТЕРА, ПРЕДСТАВЛЕНИЕ «ТАНГЛ+ЗВЕЗДА», КАСКАДНЫЙ КОД, ХОРДОВЫЕ ДИАГРАММЫ, МЕТОД УПАКОВКИ ОКРУЖНОСТЕЙ, АЛГОРИТМ SONO, МИНИМАЛЬНАЯ ДЛИНА ПЕРЕПЛЕТЕНИЯ

В данной работе реализован алгоритм нахождения идеального состояния двоякопериодических переплетений, то есть его максимальной плотности. Перед тем как реализовывать алгоритм было выполнено матричное представление двоякопериодического переплетения и его визуализация с помощью метода упаковки окружностей. Нахождение идеального состояния двоякопериодического переплетения заключается в его затягивании. Именно поэтому алгоритм основан на алгоритме «SONO», с помощью которого затягиваются узлы до идеального состояния. Так как двоякопериодические переплетения отличаются от узлов своей внутренней структурой, то некоторые функции исходного алгоритма заменены на те, которые лучше работают для переплетений. И так как двоякопериодические переплетения задаются своей элементарной ячейкой, то в алгоритм добавлена функция, которая с помощью деформации элементарной ячейки достигает минимальной длины нити, из которой состоит переплетение, то есть максимальной плотности этого переплетения.

THE ABSTRACT

57 pages, 52 pictures

THE DENSEST DOUBLY PERIODIC INTERTWINING, KNOT THEORY, KNOT INVARIANTS, REIDEMEISTER'S MOVEMENTS, TANGLE+STAR REPRESENTATION, CASCADE CODE, CHORD DIAGRAMS, CIRCLE PACKING METHOD, SONO ALGORITHM, MINIMUM LENGTH OF INTERTWINING

In this work, an algorithm is implemented for finding the ideal state of two-periodic intertwining, that is, its maximum density. Before implementing the algorithm, a matrix representation of the two-periodic intertwining and its visualization using the circle packing method were performed. Finding the ideal state of a two-periodic intertwining consists in tightening it. That is why the algorithm is based on the "SONO" algorithm, with which the knots are tightened to an ideal state. Since two-periodic intertwining differ from knots in their internal structure, some functions of the original algorithm have been replaced with those that work better for intertwining. And since the two-periodic intertwining is defined by its unit cell, a function has been added to the algorithm that, by deforming the unit cell, reaches the minimum length of the thread that the intertwining consists of, that is, the maximum density of this intertwining.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	5
ГЛАВА 1. ПОСТАНОВКА ЗАДАЧИ.....	7
ГЛАВА 2. ТОПОЛОГИЧЕСКОЕ ПРЕДСТАВЛЕНИЕ ПЕРЕПЛЕТЕНИЙ.....	18
2.1. Проекций vs диаграмма на торе. Идея классификации при помощи стягивания.....	19
2.2. Классификация танглов, звезд, генерация проекций и переплетений	21
2.3. Визуализация переплетений с помощью метода упаковки окружностей	29
ГЛАВА 3. ИДЕАЛЬНЫЕ (ПЛОТНЕЙШИЕ) ПЕРЕПЛЕТЕНИЯ.....	34
3.1. Особенности реализации SONO для двоякопериодических переплетений	36
3.2. Основные процедуры алгоритма.....	36
3.2.1. Подготовка переплетений к алгоритму	36
3.2.2. Алгоритм движения по градиенту	37
3.2.3. Деформация ячейки.....	38
3.2.4. Умножение на масштабный коэффициент	40
3.2.5. Удаление пересечений.....	41
ГЛАВА 4. РЕЗУЛЬТАТЫ.....	43
ЗАКЛЮЧЕНИЕ	54
СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ	55

ВВЕДЕНИЕ

Теория узлов является одной из первых областей топологии. Топология – это наука, изучающая свойства пространства, которые остаются постоянными при непрерывных изменениях этого пространства. Одной из основных задач теории узлов являются задачи перечисления и классификаций.

Теория узлов играет важную роль как в математике, так и в физике. Так, например, лорд Кельвин предложил описывать химические элементы с помощью узлов, а некоторые ученые находят связь между моделями статистической механики и узлов. Теория узлов применяется даже в изучении биологии, так как узлы проявляются в ните подобной структуре ДНК.

В теории узлов постоянно появляются новые направления изучения. Первым этапом изучения были узлы – замкнутые кривые без самопересечений. Гаусс был первым, кто решил рассматривать узел как математический объект. Уже к концу XIX века появились первые таблицы перечислений, но пока что только до 10 пересечений. Потом стали рассматривать комбинации узлов, которые называются зацеплениями, и теорию кос.

В 1927 году Курт Рейдемейстер показал, что две плоские диаграммы узлов или зацеплений, которое относятся к одному и тому же узлу и зацеплению, могут быть получены друг из друга с помощью последовательного применения движений Рейдемейстера. После этого у ученых появилась возможность проверить все существующие таблицы перечислений и начать удалять повторяющиеся элементы.

Но данные движения – это не единственный способ вычислить повторы, ещё существуют инварианты. Если у двух проекции равны инварианты, значит эти проекции описывают один и тот же узел или одно и то же зацепление. Так были созданы полиномиальные инварианты, инвариант раскрасок и другие.

Одним из относительно новых направлений в теории узлов является изучение двоякопериодических переплетений. Отличие данного математического объекта заключается в том, что его структура и свойства задаются минимальным повторяющимся элементом – элементарной ячейкой.

Проблема изучения данного направления заключается в том, что отсутствует классификация переплетений и что из созданных таблиц диаграмм сложно удалить повторы. И лучше всего это делать с помощью нахождения их идеальной формы. Так называется еще один инвариант. Его суть заключается в том, что надо найти максимальную плотность переплетения и сделать визуализацию переплетения в таком состоянии.

Изучение двоякопериодических переплетения – актуальная задача на сегодняшний день, потому что из таких переплетений состоят ткани. Именно поэтому другое их название – «текстильные» переплетения.

Главной целью данной работы является разработка алгоритма, с помощью которого можно будет найти идеальное состояние переплетения и его максимальную плотность. Для достижения данной цели необходимо решить следующие задачи:

1. Написать программу для создания матрицы, описывающей начальное двоякопериодическое переплетения;
2. Сделать визуализацию двоякопериодического переплетения;
3. Разработать алгоритм, который находит максимальную плотность переплетения.

После того как алгоритм будет разработан, можно будет очистить все таблицы перечислений от повторов. А следовательно, можно будет переходить к изучению механических свойств двоякопериодических переплетений, например, таких как разрыв при растяжении.

ГЛАВА 1. ПОСТАНОВКА ЗАДАЧИ

Теория узлов изучает топологические свойства одномерных подмногообразий свойства трехмерного пространства. Узлом можно назвать замкнутую гладкую кривую в трехмерном пространстве. Обычно узлы моделируются с помощью проекции на плоскость, например, на плоскость Oxy . Без потери общности можно считать, что проекция узла на заданную плоскость есть вложенный конечный граф со степенью валентности четыре. Вершина получившегося графа называется перекрестком, а часть узла называется ветвью [2]. Примеры диаграмм простейших узлов представлены на Рис. 1.

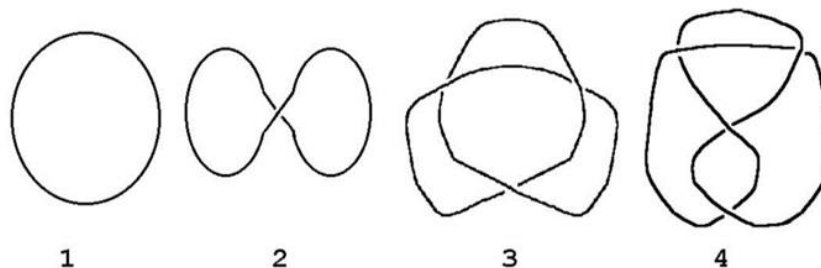


Рис. 1. Диаграммы простейших узлов

Под тривиальным узлом понимается такой узел, диаграмма которого может быть представлена без перекрестков. Так первая и вторая изображенные диаграммы соответствуют тривиальному узлу. Третья диаграмма – диаграмма трилистника, а четвертая соответствует узлу, который называется восьмеркой.

Видно, что в каждом перекрестке пересекается две ветви узла. Причем одна из них проходит сверху, образуя переход, а другая проходит снизу, образуя проход. В каждой точке пересечения указывается ветвь, которая проходит сверху и имеет большую координату по оси Z , и ветвь, которая проходит снизу, то есть имеет меньшую координату по оси Z .

Если сгруппировать между собой несколько узлов с условием, что они не пересекаются между собой и не пересекают сами себя, то получится зацепление. Теория зацепления является одной из задач теории узлов.

Зацепление – это несколько непересекающихся и самонересекающихся замкнутых кривых в пространстве. То есть зацепление состоит из узлов, которые называются компонентами. Моделируются зацепления также, как и узлы. Единственное отличие, что в проекции на плоскости узла всего одна замкнутая кривая, а в проекции зацепления их несколько. Проекции зацеплений изображены на Рис. 2.

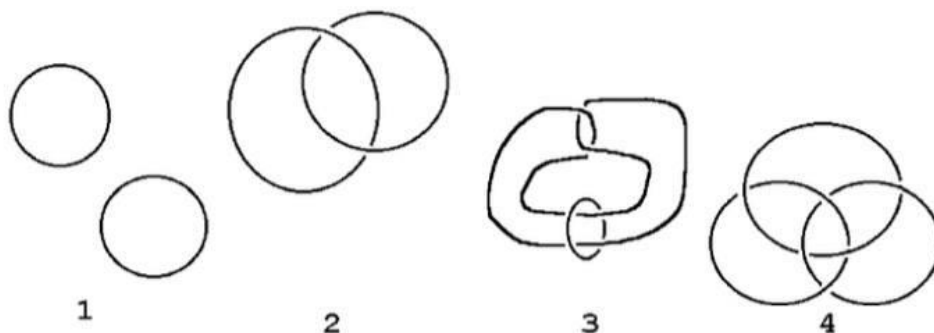


Рис. 2. Диаграммы простейших зацеплений

Первая диаграмма является диаграммой тривиального зацепления, которое состоит из двух компонент. Вторая диаграмма соответствует зацеплению Хопфа, третья – зацеплению Уайтхеда. На четвертой диаграмме изображены кольца Борромео. Три последних пересечения являются нетривиальными. При этом кольца Борромео имеют следующее свойство, которое заключается в том, что если удалить одну из трех компонент, то зацепление распадется на два тривиальных узла [4].

Также узлы можно рассматривать с еще одной стороны. Теория кос дает новое представление на теорию узлов. Косой называется набор непересекающихся гладких кривых, которые соединяют точки первой области с точками второй. Отличие кос от узлов состоит в том, что косы имеют особенную внутреннюю структура, то есть внутренние свойства кос отличаются от свойств узла [3]. Пример косы представлен на Рис. 3.

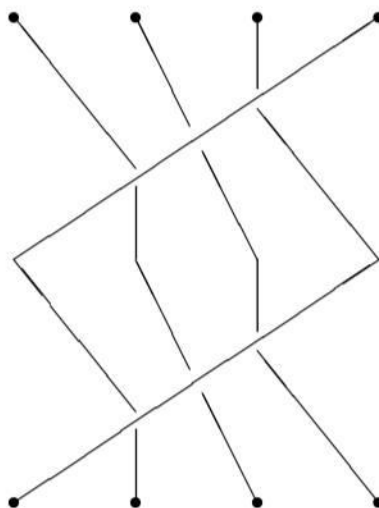


Рис. 3. Коса

Из косы всегда можно получить зацепление с помощью операции замыкания. Результат такой операции изображен на Рис. 4, получилось несколько непересекающихся и самонепересекающихся замкнутых кривых.

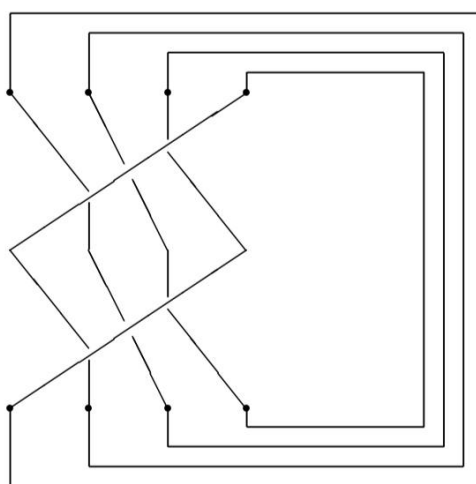


Рис. 4. Замыкание косы

Не всегда удается изобразить трехмерный объект на плоскости с сохранением всех исходных свойств. Поэтому в конце 90-х годов 20-ого века появилось понятие виртуального зацепления.

Виртуальным зацеплением называется такое зацепление, у которого помимо классических точек пересечения, в которых образуются проходы и переходы, еще есть виртуальные пересечения [7]. В виртуальных перекрестках

не образуются новые проходы и переходы, то есть обе ветви имеют одну и ту же координату по оси Z . В трехмерном представлении зацепления таких перекрестков нет. На плоскости они образуются из-за ограничения размеров плоскости, из-за отсутствия возможности растащить эти две ветви на плоскости. На Рис. 5 изображены виртуальные зацепления. Перекрестки, в их классическом понимании, пронумерованы, а виртуальные перекрестки выделены кружками.

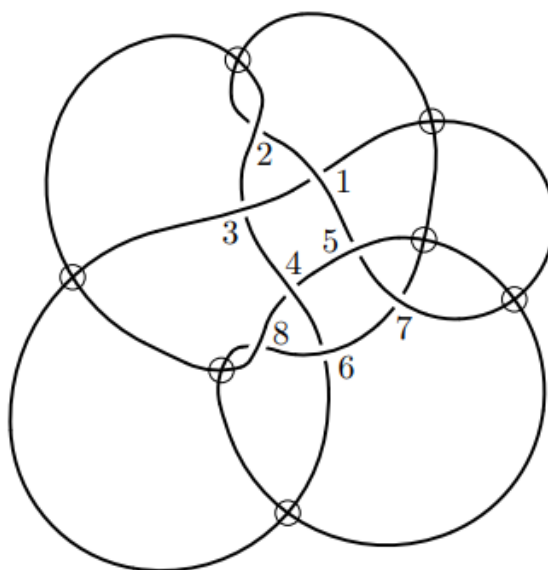


Рис. 5. Виртуальные пересечения

Виртуальные зацепления имеют еще и практическое применение. Такие зацепления соответствуют двоякопериодическим переплетениям или, как их еще называют, «текстильным» переплетениям. Именно поэтому соответствующая теория может быть использована для создания топологической классификация тканых и вязанных переплетений. Примеры двоякопериодических переплетений изображены на Рис. 6.

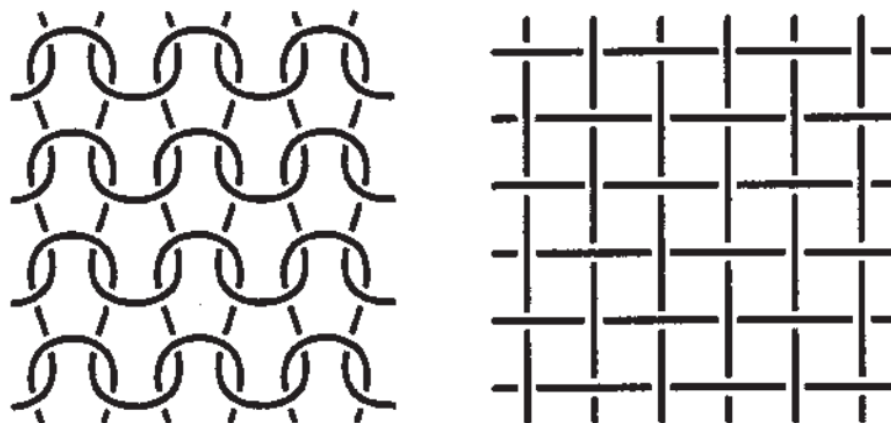


Рис. 6. Диаграммы двоякопериодических переплетений

Двоякопериодическое переплетение полностью определяется заданием минимального повторяющегося элемента, который называется элементарной ячейкой [15]. Данный элемент может быть выбран разными способами, при этом свойства ячейки не зависят от выбора этого способа.

Раздел о двоякопериодических переплетениях является относительно новым в теории узлов, поэтому результатов, полученных по этой теме, гораздо меньше, чем для самих узлов, кос и зацеплений. Проблема топологической классификация «текстильных» переплетений ещё не решена до конца. В данной области топология используется как инструмент описания внутренней структуры, то есть благодаря топологии можно выявить какие две элементарные ячейки, которые на первый взгляд кажутся разными, на самом деле задают одно и то же переплетение. Решением задачи классификации, другими словами, задачи перечисления, являются таблицы, в которых собраны изображения и данные о всевозможных переплетениях, сгруппированные по разным характеристикам. Такими характеристиками, например, может быть количество точек пересечения, количество нитей, из которых состоит диаграмма. В таблицах, которые есть сегодня, встречаются повторы, которые надо выявить и удалить. В решении данной проблемы помогут инварианты.

Инвариант имеет одно и то же понятие как для узлов, зацеплений и кос, так и для двоякопериодических переплетений. Под инвариантом понимается

функция на множестве узлов, которая принимает одинаковые значения для эквивалентных узлов. Два узла называются топологически эквивалентными, если один из них можно непрерывно преобразовать в другой без самопересечений. Для проверки узлов на эквивалентность используются движения Рейдемейстера [8]. Основная идея заключается в том, что если есть две плоские диаграммы одного и того же узла или переплетения, то можно получить одну из другой с помощью таких образований. На Рис. 7 показана визуализация движений Рейдемейстера.

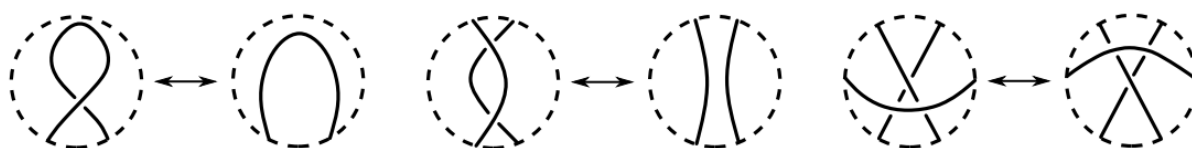


Рис. 7. Движения Рейдемейстера

Движения Рейдемейстера бывают трех типов. Первым изображено скручивание или раскручивание в любом направлении, вторым – перемещение целой петли через другую, третьим – перемещений целой нити над переплетением или под ним. Каждое из этих движений локальное, то есть действует только на отдельно выбранной части диаграммы, в данном случае внутри области, ограниченной пунктиром, в то время как другая часть остается без изменений, то есть вне ограниченной области. Инвариант остается постоянным при изменении вида диаграммы. Теперь под тривиальным узлом понимается такой узел, который можно с помощью движений Рейдемейстера развязать в окружность.

Одним из самых простых инвариантов является инвариант раскрасок. Суть данного метода заключается в том, что раскраска диаграммы считается правильной, если в каждой её точке пересечения три дуги, инцидентные этой точке, покрашены либо в один цвет, либо в три разных цвета. Под дугой понимается такая ветвь диаграммы, которая все время идет сверху при прохождении перекрестков, то есть образует только переходы. Дуга считается инцидентной данной точке, если эта точка является концевой для дуги. У

каждой диаграммы может быть не одна правильная раскраска, количество правильных раскрасок называется функцией раскрасок на данной диаграмме. Функция раскрасок постоянна при движениях Рейдемейстера, то есть является инвариантом. В данной функции учитываются только правильные раскраски, другими словами, только правильные раскраски являются инвариантом узла или переплетения. Чтобы это проверить надо сопоставлять раскраски диаграммы до использования движений Рейдемейстера и после. Для одноцветных раскрасок, то есть для таких, в которых все дуги имеют один цвет, надо сопоставлять одноцветные раскраски такого же цвета. Рассмотрим каждое движение по отдельности. Первое движение, скручивание или раскручивание в любом направлении, сохраняет количество раскрасок, потому что при добавлении петли в точке, в этой точке сходится всего лишь две дуги, а следовательно, правильная раскраска должна раскрасить эти дуги в один и тот же цвет. И вся дуга, на которой образуется эта петля, должна быть изображена одним цветом. Первое движение Рейдемейстера изменяет только одну ветвь проекции, а второе и третье движение сразу несколько, поэтому для них используются неоднородные раскраски. Каждой такой раскраске до применения движений можно сопоставить соответствующую ей однозначным образом раскраску после применения второго или третьего движений. Дуги, которые не участвуют в движениях, сохраняют свой прежний цвет. Изменение раскраски перекрестка при применении второго движения изображено на Рис. 8, а при применении третьего – на Рис. 9.

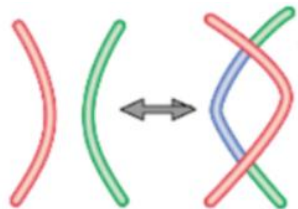


Рис. 8. Изменение раскраски после второго движения Рейдемейстера

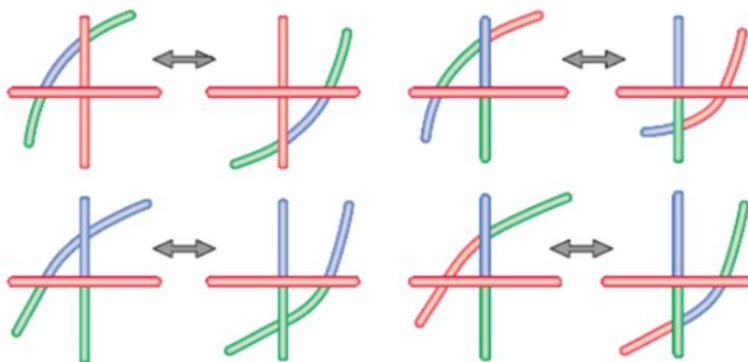


Рис. 9. Изменение раскраски после третьего движения Рейдемейстера

Значениями инварианта могут быть как числа, векторы, матрицы, так и полиномы. Поэтому еще одной группой инвариантов являются полиномиальные инварианты узлов, то есть значениями инвариантов являются многочлены. Существует несколько таких инвариантов. Один из них – это многочлен Джонса. Его суть заключается в том, что для каждого узла или переплетения соответствует многочлен от формальной переменной $t^{\frac{1}{2}}$ с целыми коэффициентами. Но такой многочлен подходит только для ориентированных узлов или переплетений. Ориентированными называются такие узлы, на которых указываются направления обхода для каждой компоненты. Для них также строятся плоские диаграммы и также можно использовать движения Рейдемейстера, главное, сохранять ориентацию до и после их использования. Рис. 10 – пример ориентированного узла.

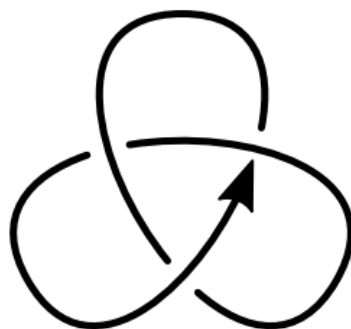


Рис. 10. Пример ориентированного узла

Еще один полиномиальный инвариант – полином Конвея – полином от одной переменной с постоянными значениями [6]. Данный полином

используется также только на ориентированных узлах, но не зависит от конкретного выбора обхода. Еще одно условие для такого инварианта, что полином Конвея тривиального узла равен 1. Полином равен нулю для тривиального зацепления из двух или более компонент. Для каждой тройки диаграмм, полином Конвея C должен удовлетворять следующему условию:

$$C(L_+) - C(L_-) = C(L_0), \quad (1)$$

где L_+ - начальная диаграмма рассматриваемого перекрестка, L_- - диаграмма того же перекрестка после движений Рейдемейстера, а L_0 - диаграмма зацепления вне рассматриваемого перекрестка. То есть основная идея данного инварианта заключается в том, что на каждом этапе его расчета задача сводится к вычислению инварианта зацепления с количеством перекрестков на единицу меньше, чем на предыдущем этапе. Причем полином Конвея единственен для каждого зацепления или узла.

Еще одним инвариантом для узлов и зацеплений является их идеальное состояние. С точки зрения физики, узлы и зацепления рассматривают как материальные объекты, которые имеют энергию внутреннего взаимодействия. Основным состоянием узлов называется такое состояние, при котором достигается минимум энергии. Энергию можно рассматривать как функцию различных переменных, но в теории узлов, чаще всего энергию рассматривают как функцию толщины, то есть она зависит от масштабного инвариантного параметра, равного отношению длины веревки к её диаметру. Конфигурация узла, при которой данный параметр достигает своего минимума, называется основным состоянием узла. Минимум масштабного коэффициента достигается при достижении минимума суммарной длины всех компонент узла или зацепления, при этом диаметр этих компонент остается постоянным. Одновременно с достижением минимума длины достигается максимум плотности узла или переплетения, так как плотность равна обратному масштабному инвариантному коэффициенту. Конфигурация узла, зацепления или переплетения, при которой достигается их максимальная плотность,

называется их идеальным состоянием. Идеальное состояние также является инвариантом, потому что на диаграмме компоненты пересечения можно разложить различными способами, но при стягивании, а именно так достигается и минимум энергии внутренних взаимодействий, и максимальная плотность, все диаграммы будут выглядеть одинаково и будут иметь равные значения инвариантов [9].

Идеальные узлы уже давно изучаются. Для них существуют алгоритмы, которые для заданной диаграммы могут найти максимальную плотность и минимальную длину. Также существуют таблицы, в которых перечислены не только идеальные узлы, но еще и значения их масштабнo-инвариантного узла. Так как уже для каждого узла или зацепления можно найти их идеальную форму, то можно составить таблицу перечисления всевозможных узлов и зацеплений без повторов, так как для каждого будет вычислен инвариант и будет проведено сравнение с другими элементами таблицы. Следовательно, топологическая задача классификации для узлов и зацеплений решена, можно переходить к изучению их механических свойств. Пример стягивания узла до его идеальной формы представлен на Рис. 11.

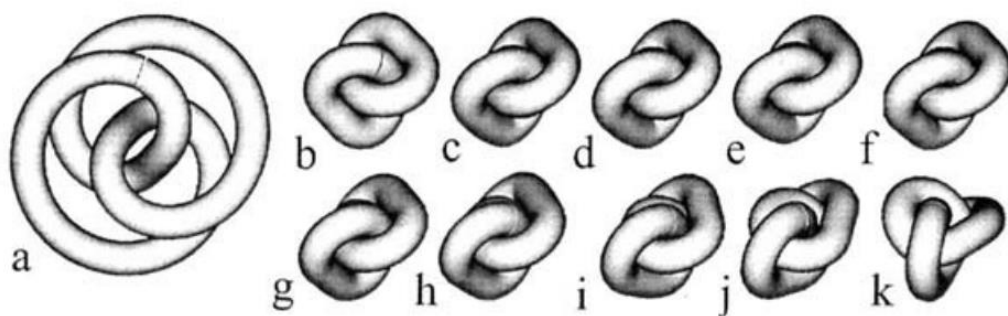


Рис. 11. Стягивание узла до идеальной формы

Для двоякопериодических переплетения нет алгоритмов, которые находили бы их минимальную длину и максимальную плотность, а таблицы перечисления содержат множество повторов, которые без данных алгоритмов невозможно выявить. На сегодняшний день эта актуальная задача, так как двоякопериодические переплетения используются в изготовлении тканей и

было бы полезно учитывать механические свойства каждого из них. Но если начинать изучать такие свойства для каждого переплетения отдельно, то это будет неструктурированная информация, получение которой займет много времени. Именно поэтому необходимо разработать алгоритм, который для каждого переплетения будет находить их идеальное состояние.

ГЛАВА 2. ТОПОЛОГИЧЕСКОЕ ПРЕДСТАВЛЕНИЕ ПЕРЕПЛЕТЕНИЙ

Двоякопериодические переплетения являются виртуальными зацеплениями. Введем понятие рода виртуального зацепления. Род виртуального зацепления равен минимальному роду поверхности, на которой можно изобразить зацепление. А род поверхности характеризует порядок связности этой поверхности, то есть он равен максимальному числу замкнутых самонепересекающихся кривых, которые можно изобразить на поверхности, не разделяющих её на части. Так, например, род поверхности сферы равен 0, а род поверхности тора равен 1.

Двоякопериодические переплетения – виртуальные зацепления первого рода. Для создания их топологической классификации можно использовать метод «Вложенного графа». Его суть заключается в том, что плоскую проекцию переплетения можно изображать на замкнутой поверхности. Вложенным графом называется связный мультиграф с фиксированным циклическим порядком ребер в каждой вершине. Мультиграфом называется такой граф, в котором разрешается присутствие кратных ребер, то есть в одном графе может существовать несколько ребер, у которых одни и те же конечные вершины. Говорят, что граф вложен в поверхность, если его ребра разбивают поверхность на грани, гомеоморфные кругу [1]. Другими словами, ребра проекции разбивают поверхность на такие грани, которые путем обратимых преобразований можно представить в виде круга. Пример зацепления трёхвалентного графа на поверхностях 0, 1 и 2 рода представлен на Рис. 12. Видно, что поверхность сферы, поверхность нулевого рода делится на шесть четырехугольных граней, а поверхность тора, то есть поверхность первого рода, на четыре шестиугольных грани, поверхность второго рода, которая называется поверхность кренделя, делится на две двенадцатиугольные грани.

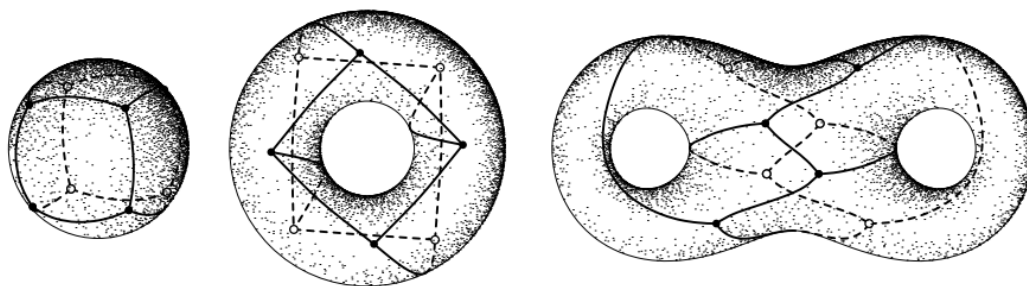


Рис. 12. Зацепление трёхвалентного графа на поверхностях 0, 1 и 2 рода

2.1. Проекций vs диаграмма на торе. Идея классификации при помощи стягивания

Диаграмма на торе – это один из способов изображать проекцию виртуальных переплетений. Его преимущество заключается в том, что благодаря изображению на замкнутой поверхности можно избавиться от виртуальных перекрестков, которые образуются из-за ограниченности плоскости. То есть в диаграмме на торе сохраняются только те перекрестки, в которых образуют переходы и проходы. Когда плоскую проекцию изображают на поверхностях соответствующего рода, то говорят, что рассматривается зацепление «над» этой поверхностью. Например, диаграммы двоякопериодических переплетений рассматривают как зацепление «над» тором.

Диаграммы переплетений делятся ещё на два типа: альтернированные и неальтернированные [5]. Альтернированной называется такая диаграмма, всякая дуга которой, соединяющая соседние перекрестки, проходит в одном из них сверху, а в другом из них – снизу. При определении типа диаграммы виртуальные перекрестки не учитываются. Само переплетение называется альтернированным, если из всех его диаграмм существует хотя бы одна такого типа. Пример неальтернированного узла представлен на Рис. 13, дуга, которая в двух соседних перекрестках проходит сверху, выделена красными кругами.

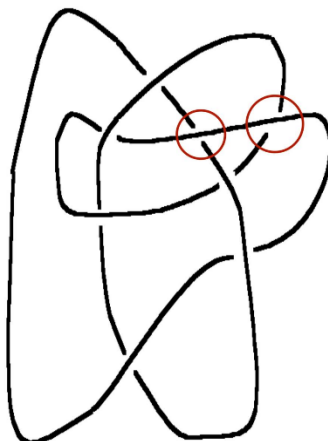


Рис. 13. Неальтернированный узел

Одну и ту же проекцию можно изображать на поверхностях разного рода. Для того чтобы найти максимальный род поверхности, на которой можно изобразить данную диаграмму используют формулу Эйлера. Она выражает зависимость между количеством вершин диаграммы, другими словами, точками пересечений переплетения, количеством ребер и граней этой диаграммы. Сама формула записывается следующим образом:

$$v - e + f = 2 - 2g, \quad (2)$$

где v – количество вершин, e – количество ребер, f – количество граней, g – род поверхности. Грани проекции образуются её ребрами. Грань – это область, внутри которой отсутствуют вершины и ребра проекции. Так как мы рассматриваем проекции со степенью валентности четыре, то есть из каждой вершины исходит по четыре ребра, то количество ребер равно удвоенному значению количества вершин, то есть:

$$e = 2v. \quad (3)$$

Тогда формулу Эйлера можно переписать в таком виде:

$$f - v = 2 - 2g. \quad (4)$$

Существует также «ленточное» представление вложенных проекций. Оно используется для того, чтобы определить число граней. Для этого надо изобразить вершины проекции и отрезки ребер этих вершин, причем отрезки

ребер надо изображать в заданном циклическом порядке. После этого надо соединить ребра с одинаковыми номерами лентами без перекручивания. Получилась ленточная проекция, край которой состоит из нескольких компонент. Число этих компонент равно числу граней исходной проекции. После этого можно применять формулу Эйлера и находить род поверхности. Рис. 14 – пример ленточной проекции. Слева изображена каждая вершина с ребрами, расположенными в циклическом порядке, справа ребра с одинаковыми номерами уже соединены. Для того чтобы выделить число компонент, из которых состоит край диаграммы, надо двигаться по краю ленты, начиная с некоторой точки, до возвращения в эту же точку.

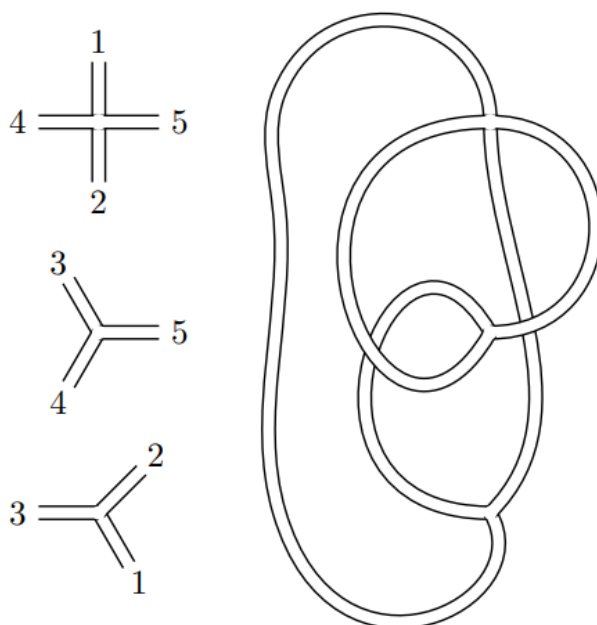


Рис. 14. Ленточная проекция

2.2. Классификация танглов, звезд, генерация проекций и переплетений

Одним из способов генерировать проекцию двоякопериодического переплетения является представление «Тангл + звезда». Его удобство заключается в том, что каждый элемент этого представления отвечает за отдельную часть информации о переплетении. А именно, звезда отвечает за все свойства, связанные с родом переплетения, а тангл представляет «плоскую

часть» диаграммы. Это позволяет рассматривать виртуальные зацепления каждого рода отдельно, заранее перечислив соответствующие им звезды. Примеры звезды и тангла и их комбинация изображены на Рис. 15. Далее рассмотрим каждую составляющую проекции отдельно.

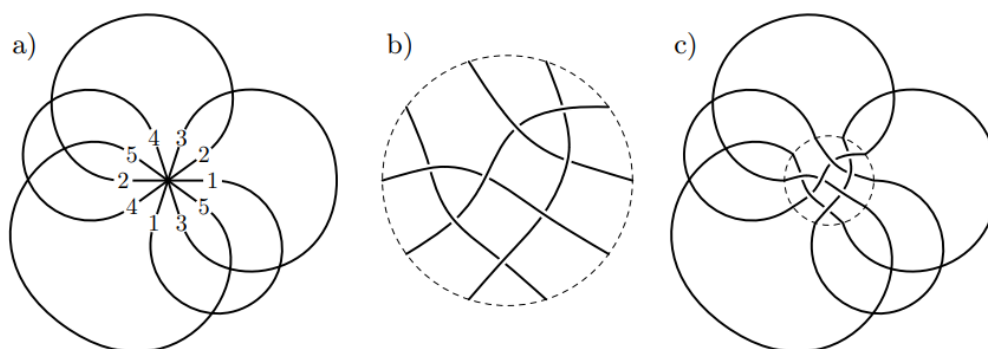


Рис. 15. Пример звезды, пример тангла и их комбинация

k -танглом называется класс эквивалентности планарного графа с $2k$ вершинами валентности один, то есть с одним ребром, и некоторым заданным числом вершин валентности 4, в каждой из которых фиксирован циклический порядок ребер, причем одна из пар противоположащих ребер помечена как «верхняя». Отношение эквивалентности задается движениями Рейдемейстера. Но проекции k -танглов от простых диаграмм отличаются тем, что в перекрестках танглов нет деления дуг на верхние и нижние, а значит нельзя пользоваться движениями Рейдемейстера. Пример 4-тангла изображен на Рис. 16. Видно, что на границе расположено 8 вершин со степенью валентности один.

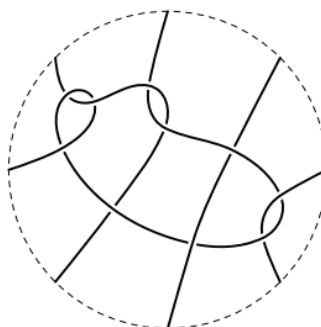


Рис. 16. Пример 4-тангла

Для моделирования двоякопериодических переплетений требуется, чтобы проекции их элементарных ячеек удовлетворяли условиям связности и простоты. В представлении «тангл + звезда» за выполнение этих условий отвечает тангл. Тангл называется связным, если между двумя его вершинами существует как минимум один путь по ребрам. Пример несвязной проекции, другими словами, составной, изображен на Рис. 17. Видно, что проекция состоит из двух частей, несвязных между собой.

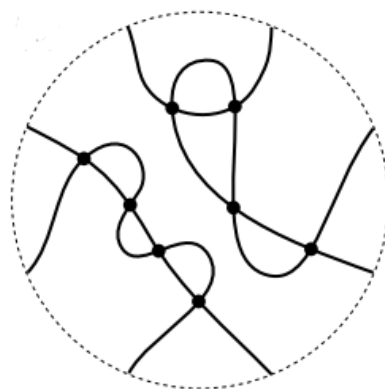


Рис. 17. Пример несвязной проекции

Простым называется такой тангл, в каждой диаграмме которого отсутствует нетривиальный 1-подтангл, то есть двуногий. Пример тангла, не удовлетворяющего данному условию, представлен на Рис. 18. Такие проекции называются составными.

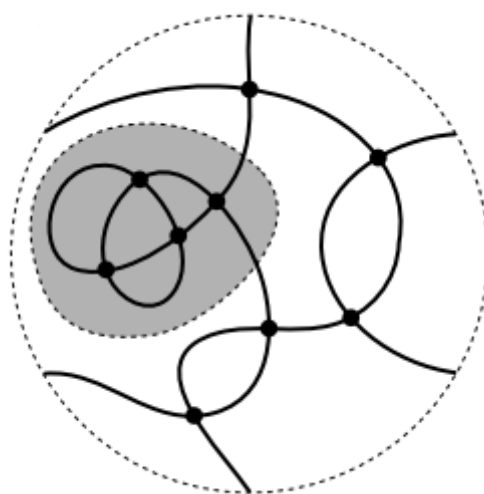


Рис. 18. Пример составного тангла

Аналогией движений Рейдемейстера в теории танглов является удаление приграничных перекрестков. Приграничным перекрестком называется такая вершина, одно из ребер которой соединено с одной из граничных точек, то есть с одной из одновалентных вершин тангла. Удалением приграничных перекрестков называется операция, при которой выбранный перекресток удаляется с проекции вместе с теми ребрами, которые соединяли его с граничными точками, а свободные концы оставшихся двух ребер становятся новыми граничными точками проекции. При данной операции число граничных точек может либо увеличиваться на 2, либо уменьшаться, но также это число может и не изменяться вовсе. Пример последовательно применения операции удаления приграничных перекрестков изображен на Рис. 19.

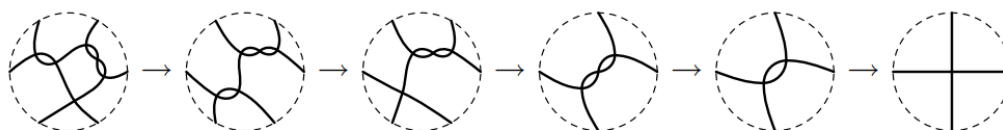


Рис. 19. Последовательное удаление приграничных перекрестков

Видно, что с помощью операции можно любую простую и связную проекцию превратить в простейшую, то есть в проекцию с одним перекрестком. Существует и обратная операция, она называется приклеивание приграничного перекрестка. С её помощью из простейшей проекции можно получить любую также простую и связную проекцию.

Один из способов строить тангл – каскадная диаграмма. Основная идея данного метода заключается в том, что тангл делится на систему вложенных областей таким образом, чтобы внутри каждой области была всего одна вершина. Такое разделение всегда можно сделать несколькими способами. Если разрезать весь тангл по линии, идущей из вершины центральной области к границе и пересекающей границу каждой области только один раз, то, вытягивая разрез за центральную вершину, получится ниспадающий каскад, который и называется каскадной диаграммой тангла [11]. Пример разделения

тангла на области и каскад, который получается в результате, представлены на Рис. 20.

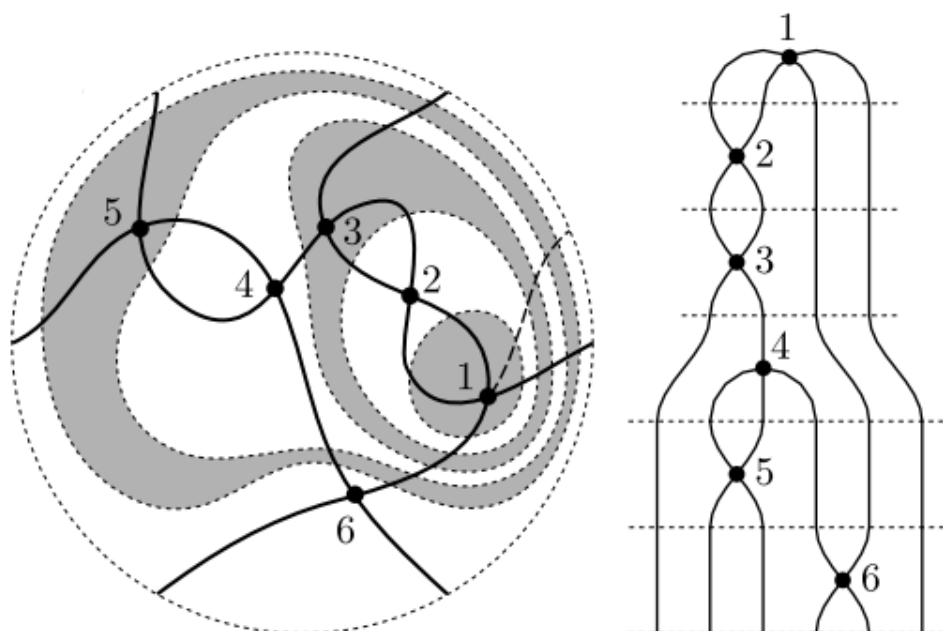


Рис. 20. Пример каскада тангла

Линия разреза не может пересекать ребра проекции. Если получилось так, что линия не может выйти из области, не пересекая ребра, то она должна пройти через вершину данной области. Количество уровней каскада равно количеству перекрестков тангла. На каждом таком уровне есть только одна вершина и ребра, инцидентные ей, пересекают или верхнюю, или нижнюю, или обе границы уровня. Для вершин возможны только пять конфигураций, которые соответствуют следующим шаблонам, изображенным на Рис. 21.

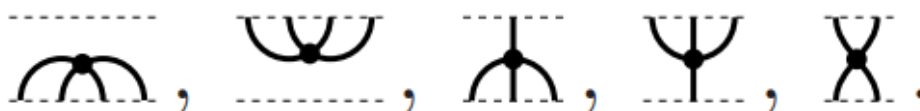


Рис. 21. Шаблоны вершин

При чем первый шаблон соответствует только стартовой вершине, то есть вершине центральной области, и встречается во всей диаграмме только один раз. А второй шаблон вообще не встречается в каскадных диаграммах, если диаграмма строится для простых проекций, так как при такой

комбинации ни одно из ребер не пересекает нижнюю границу каскада, а значит образуется отдельный замкнутый элемент проекции, что не удовлетворяет условию простоты. Третья, четвертая и пятая конфигурации соответствуют оставшимся вершинам каскада.

Кодирование каскадной диаграммы начинается с самой первой, то есть центральной, области. Но так как шаблон для неё всегда одинаковый, то из каскадного кода её исключают. Каждая из трех оставшихся конфигураций в коде заменяется определенным числом: третья конфигурация заменяется числом +1, так как её использование добавляет к числу ребер на предыдущем уровне каскаде еще 2 ребра, четвертая конфигурация заменяется числом -1, так как она уменьшает число ребер на два, а последняя конфигурация заменяется числом 0, так как при её использовании число ребер не изменяется. Также при кодировании нужно определять номер ребра на предыдущем уровне каскада, к которому приклеивается новая конфигурация. При этом отчет ребер начинается с крайнего левого на каждом уровне.

При генерации каждого уровня каскада идет подсчет ребер и их нумерация, а также записывается массив граней, то есть грани под какими номерами окружают каждое ребро. Число ребер, которые пересекают нижнюю границу последнего уровня каскада, должно равняться $2k$. Подсчет граней осуществляется для того, чтобы сразу проверять проекцию на выполнение условия связности, больше всего шансов, что возникнет подтангл в проекции при использовании четвертого шаблона для вершин.

Результатом каскадного кода является матрица, число строк которой равно значению перекрестков танглов, а количество столбцов равно четырем, так как каждая вершина проекции имеет всего четыре ребра. В каждой строке записывается фиксированный циклический порядок номеров ребер, которые исходят из данной вершины. Такая матрица называется «вершины-ребра».

После того как сгенерирована проекция тангла, можно переходить к созданию звезды, которую еще по-другому называют букетом вложенных окружностей. Звезда – это одновершинный граф с фиксированным циклическим порядком ребер [12]. Пример звезды изображен на Рис. 22.

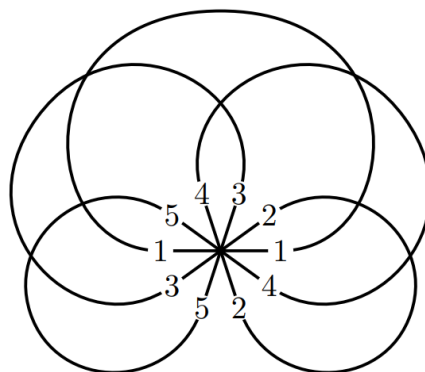


Рис. 22. Пример звезды

Для перечисления вложенных букетов пользуются их взаимно-однозначным соответствием с хордовыми диаграммами. Хордовыми диаграммами называется ориентированная окружность с $2k$ различными отмеченными точками (вершинами), которые произвольным образом разбиты на пары. Звезды и хордовые диаграммы рассматриваются с точностью до изоморфизмов, а все возможные изоморфизмы букетов сводятся к поворотам и отражениям в порядке ребер, другими словами, к поворотам и отражениям соответствующей хордовой диаграммы. Пример хордовой диаграммы, соответствующей примеру звезды, показан на Рис. 23.

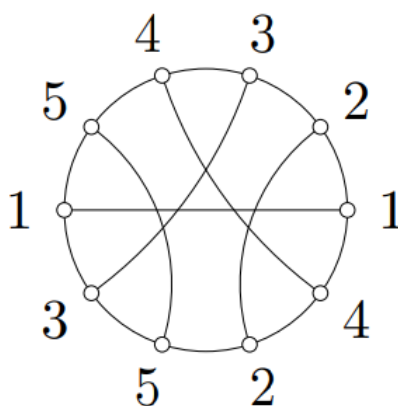


Рис. 23. Хордовая диаграмма звезды

Две хордовые диаграммы считаются эквивалентными, если они обладают одинаковым количеством хорд и если одна может быть получена из другой с помощью поворота окружности или её зеркального отражения.

Для моделирования виртуальных зацеплений, а то есть и для двоякопериодических переплетений, интересны только хордовые диаграммы простых букетов. Простым называется такой букет, который не содержит петель и параллельных ребер. Петлей называется такое ребро хордовой диаграммы, которое может быть стянуто в точку на поверхности вложения. То есть его конечными вершинами являются соседние точки на границе. Два ребра называются параллельными, если они гомотопны на поверхности вложения. Другими словами, если этим ребрам соответствуют хорды, которые начинаются и заканчиваются в соседних точках границы, но при этом внутри диаграммы не пересекаются. Так, например, на Рис. 24 петлей является ребро под номером 5. А на Рис. 25 параллельными ребрами являются ребра под номерами 4 и 5.

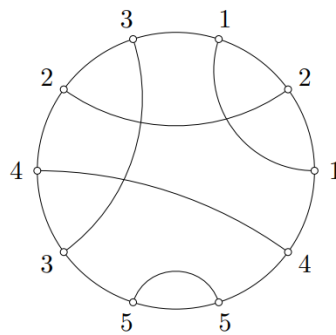


Рис. 24. Пример хордовой диаграммы с петлей

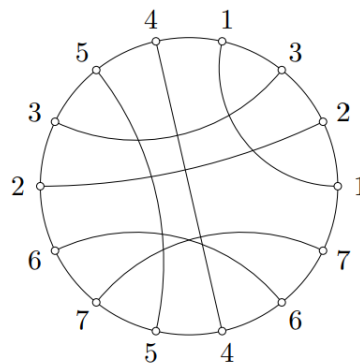


Рис. 25. Пример хордовой диаграммы с параллельными ребрами

После того как получены тангл и звезда, их надо склеить, то есть склеить ребра на последнем уровне каскада по схеме звезды. После данной процедуры получается матрица, которая описывает каждый перекресток двоякопериодического переплетения. И теперь можно переходить к его визуализации.

2.3. Визуализация переплетений с помощью метода упаковки окружностей

Для визуализации двоякопериодического переплетения используется метод упаковки окружностей. Главная идея данного метода состоит в том, что вокруг каждой вершины переплетения строится окружность определенного радиуса, окружности двух вершин могут касаться только в том случае, если эти вершины соединены ребром. Главная задача заключается в том, что надо найти значение каждого радиуса с заданной точностью [13]. Пример простой упаковки с помощью данного метода показан на Рис. 26.

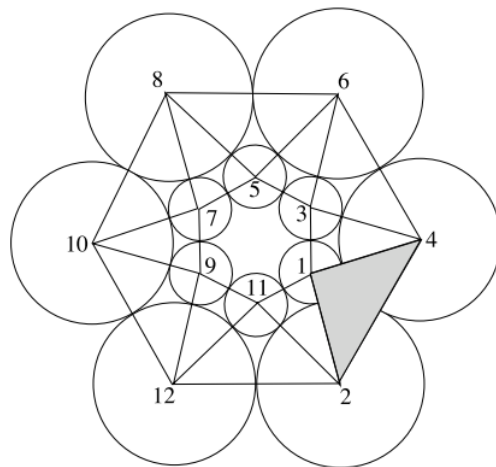


Рис. 26. Пример работы метода упаковки окружностей

Данный метод работает только для триангуляций, то есть для плоскостей, разбитых на треугольники. Поэтому перед использованием данного метода, переплетение надо разбить на перекрестки. Для этого надо каждое ребро разбить на два, то есть на каждое ребро поставить по дополнительной вершине. Также надо в каждую грань поставить вершину и соединить её с вершинами всех ребер, которые её ограничивают, в том числе и

с новыми дополнительными вершинами на ребрах. После данной процедуры можно переходить к созданию упаковки.

Весь алгоритм строится на достижении желаемой суммы углов вокруг каждой вершины. Такая сумма углов равна для каждой вершины и равняется 2π . Сумма углов вокруг каждой вершины является суммой каждого угла, который образуется между ребрами двух соседних вершин, которые являются соседними к данной. На Рис. 27 таким углом является угол α :

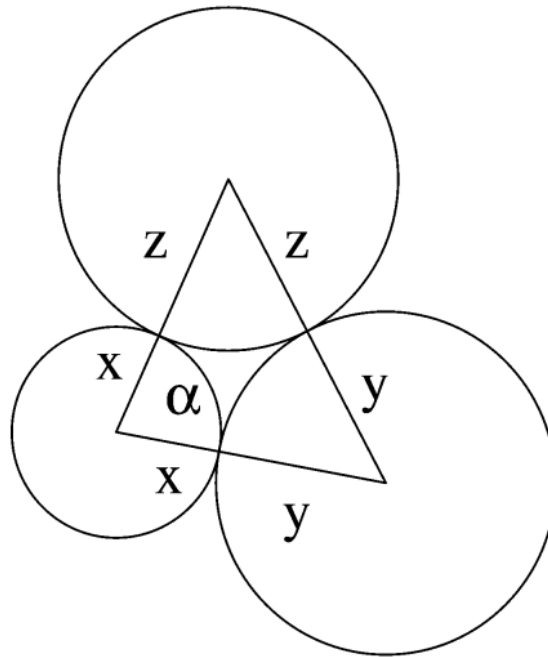


Рис. 27. Визуализация угла α

Данный угол находится по следующей формуле:

$$\alpha = \arccos \left[\frac{(x+y)^2 + (x+z)^2 - (y+z)^2}{2(x+y)(x+z)} \right], \quad (5)$$

где x, y, z – текущие радиусы окружностей. Перед началом работы алгоритма радиусы для всех окружностей равны. Для того чтобы найти суммарный угол вокруг вершины, нужно найти сумму всех углов, которые образуются с соседними вершинами. То есть суммарный угол для каждой вершины равен:

$$\theta = \sum_{i=1}^N \alpha_i. \quad (6)$$

Вокруг каждой вершины исходного переплетения образуется 8 углов. Так как рассматриваем четырёхвалентные проекции, то есть у каждой вершины есть 4 ребра, и между каждым ребром проходит грань, то каждый угол между двумя ребрами делится на два угла. В итоге получается 8 углов, то есть $N = 8$. Вокруг дополнительных вершин на ребрах образуется по 4 угла, $N = 4$. Для всей триангуляции количество углов равно в восемь раз больше, чем количество перекрестков. Радиус следует выбирать таким образом, чтобы окружности, окружающие данную окружность, не пересекались между собой, пример такого случая изображен на Рис. 28. Также чтобы первая и последняя окружности не касались друг друга, то есть чтобы они не были расположены друг относительно друга, как на Рис. 29. Видно, что в первом случае было выбрано маленькое значение радиуса окружности, поэтому все её соседи не поместились. А во втором случае радиус наоборот слишком большой, и соседей не хватило, чтобы покрыть всю границу окружности.

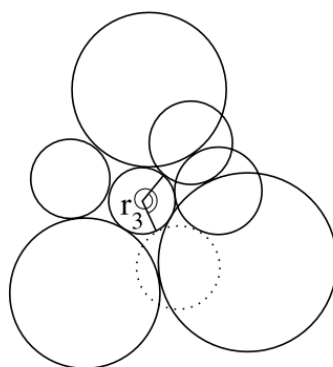


Рис. 28. Первый пример неправильной упаковки

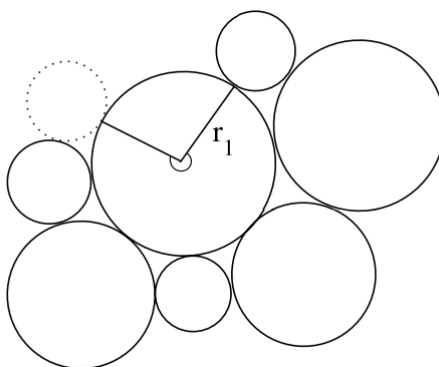


Рис. 29. Второй пример неправильной упаковки

Правильная упаковка окружностей для данного примера изображен на Рис. 30. В данном случае выбрано правильное значение радиуса, и все соседи касаются друг друга, но при этом не пересекают.

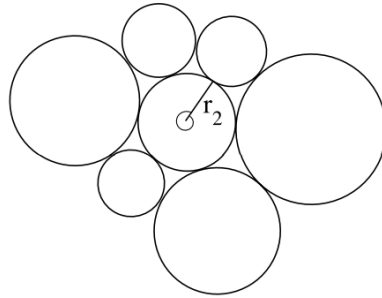


Рис. 30. Пример правильной упаковки

После того как была найдена текущая сумма углов, надо найти коэффициенты, которые будут использоваться в нахождении нового значения для радиуса:

$$\beta = \sin\left(\frac{\theta}{2k}\right), \quad (7)$$

$$\delta = \sin\left(\frac{A}{2k}\right), \quad (8)$$

где k – число соседей вершины, $A = 2\pi$ – желаемая сумма углов. Тогда новые значения для радиусов вычисляются по формуле:

$$R_i = \frac{1 - \delta}{\delta} \frac{\beta}{1 - \beta} R_{i0}, \quad (9)$$

где R_{i0} – значение радиуса для i -ой вершины на предыдущей итерации.

На каждой итерации алгоритма считается оценка накопленной ошибки:

$$c = c + \sum_N (\theta_i - A_i)^2, \quad (10)$$

$$c = \sqrt{c}. \quad (11)$$

Суммирование в данном случае идет по всем вершинам переплетения. Также на каждой итерации считается коэффициент ускорения алгоритма, который зависит от ошибок на предыдущей и на текущей итерациях:

$$\lambda = \frac{c}{c_0}. \quad (12)$$

Если коэффициент ошибки меньше единицы, то есть ошибка на текущей итерации меньше, чем на предыдущей, то ошибка пересчитывается по следующей формуле:

$$c = \lambda c. \quad (13)$$

А если модуль разности между коэффициентами на двух следующих друг за другом итерациях стремится к постоянному значению, то есть меньше какого-то заданного параметра, то пересчитывается коэффициент ускорения, и весь алгоритм ускоряется с его новым значением [14]:

$$\lambda = \frac{\lambda}{1 - \lambda}, \quad (14)$$

$$R_i = R_i + \lambda(R_i - R_0). \quad (15)$$

Алгоритм работает до тех пор, пока ошибка на каждой итерации больше заданного малого параметра. Результатом работы алгоритма являются радиусы для каждой вершины двоякопериодического переплетения. Выбрав вершину с наибольшим числом соседей и поместив её в начало координат, постепенно изображаются на плоскости все точки пересечений. Наибольшим числом соседей при этом будет одна из вершин, поставленных в грани начальной проекции.

ГЛАВА 3. ИДЕАЛЬНЫЕ (ПЛОТНЕЙШИЕ) ПЕРЕПЛЕТЕНИЯ

Для нахождения идеальной формы узлов используется алгоритм SONO, который расшифровывается как *shrink on no overlap*, то есть алгоритм сжатие без пересечений. Основная идея данного метода заключается в том, что нить моделируется в виде трубки с постоянным радиусом, длина которой постепенно уменьшается. А цель состоит в том, чтобы найти такое состояние узла, при которой достигается минимум энергии внутренних взаимодействий. Данный алгоритм выделяет один из возможных заданий функции энергии – зависимость от плотности. Вводится масштабно-инвариантный коэффициент, который равен отношению длины узла к его диаметру. В ходе работы алгоритма последовательно выполняется четыре функции [10].

Первая функция разбивает весь узел на части равной длины и следит во время всей работы алгоритма, чтобы точки разбиения так и оставались равно удаленными. Функция называется «Control Leashes (CL)». Шаг разбиения зависит от радиуса, также от шага разбиения зависит время работы алгоритма: чем меньше шаг, тем больше времени необходимо для нахождения идеального состояния. На Рис. 31 представлен узел, разбитый на равные части, нить которого моделируется сферами.

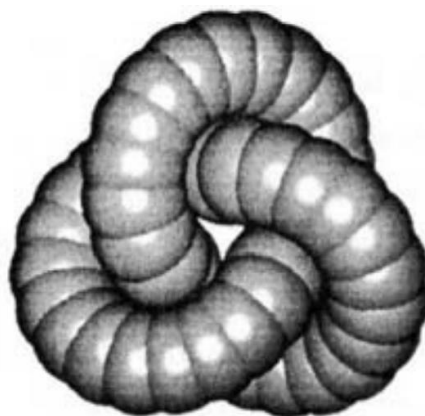


Рис. 31. Пример разбиения узла

Так как основное преимущество данного алгоритма заключается в том, что форма нити сохраняется на протяжении всего алгоритма, то есть нить не деформируется, то необходимо следить за соседями каждой точки узла, с которыми могут возникнуть пересечения при дальнейшей работе. Такая

функция называется «Find Neighbours (FN)». Для этого записывается массив соседей, в котором количество строк равно количеству точек разбиения узла. В каждой строке записываются номера вершин, которые находятся на расстоянии меньше заданного. Обычно за заданное расстояние принимают диаметр нити с малым коэффициентом. Массив соседей очищается на каждой итерации и заполняется заново.

Третья функция является продолжением второй функции, и она называется «Remove Overlaps (RO)». Если в ходе работы возникли пересечения, то есть массив соседей перестал быть нулевым, то эти пересечения необходимо удалить. Для того чтобы избавиться от пересечений, точки, которые пересекаются, расталкиваются друг от друга на такое расстояние, чтобы суммарное расстояние между ними было равно заданному. Здесь заданное расстояние такое же, как и в функции поиска соседей. Алгоритм данной функции таков: необходимо найти уравнение прямой, которая соединяет две эти точки, далее по этой прямой данные точки симметрично раздвигаются друг от друга на расстояние, равное половине разности между заданным расстоянием и текущим.

Последняя и основная функция всего алгоритма – умножение на масштабный коэффициент. Именно с помощью этой функции и моделируется затягивание узла. После того как убедились, что в узле нет пересечений, все умножается на масштабный коэффициент, который в начале алгоритма равен 0.99. После этого опять проверяют узел на наличие пересечений и, если они появились, то пробуют удалить третьей функцией. Но, если пересечения «растолкать» не удалось, то все составляющие узла умножают на вспомогательный коэффициент, который зависит от текущего значения масштабного коэффициента. Так повторяется до тех пор, пока все пересечения не будут удалены, либо пока масштабный коэффициент не будет равен 0.9999. Принимается именно такое условие выхода из алгоритма, то есть значение масштабного коэффициента 0.9999, потому что данное значение максимально

близко к единице и дальнейшие умножение на этот коэффициент не будут изменять данные узла.

Для двоякопериодических переплетений в данный алгоритм внесены некоторые изменения из-за их внутренней структуры. Более подробно про данный алгоритм и его особенности для «текстильных» переплетений будет рассказано в этой главе.

3.1. Особенности реализации SONO для двоякопериодических переплетений

Особенность алгоритма затягивания для двоякопериодических переплетений заключается в том, что помимо затягивания нити её надо еще и выпрямлять. Это необходимо для того, чтобы находить наиболее точную минимальную длину нити, необходимую для создания каждого переплетения, и, соответственно, максимальную плотность.

Именно поэтому функция CL, заменяется на функцию, которая реализует движение по градиенту, то есть по обратному направлению производных энергии нити в каждой точке. Также благодаря такому изменению алгоритма, минимум энергии достигается быстрее, другими словами, время работы алгоритма уменьшается.

3.2. Основные процедуры алгоритма

Рассмотри каждый этап алгоритма отдельно.

3.2.1. Подготовка переплетений к алгоритму

Перед тем, как начинать работу алгоритма, двоякопериодическое переплетение надо подготовить.

Для начала необходимо найти диаметр нити, с помощью которой переплетение будет моделироваться. Диаметр можно находить двумя способами. Для первого способа вводится предположение, что нить в начальной ячейке занимает половину её площади. Так как элементарные ячейки переплетений моделировались с единичной площадью, то нить должна иметь площадь равную $\frac{1}{2}$. Занимаемую нитью площадь можно найти через

произведение её длины на диаметр. Тогда диаметр будет равен половине от длины всей нити. Но данный метод не удобно использовать, так как на первой же итерации работы алгоритма появляются пересечения, которые надо удалять. Поэтому в данной работе используется второй способ нахождения диаметра нити. Его суть заключается в том, что находится минимальное расстояние между двумя ветвями переплетения. Это минимальное расстояние и есть диаметр нити. После использования этого способами не возникают никакие пересечения, поэтому можно сразу переходить к следующему этапу.

Следующим подготовительным этапом является умножение всех координат переплетения и границ элементарной ячейки на коэффициент, равный обратному значению диаметра. Это делается только для удобства, чтобы ячейка была больше и нагляднее были все результаты работы алгоритма. После данной процедуры диаметр нити становится равен единице.

Далее необходимо разделить все нити переплетения на части с равными длинами. Шаг деления зависит от диаметра нити. Чем меньше шаг деления, тем более гладкой и непрерывной моделируется нить. Если шаг выбрать слишком большой, то на нити будут видны неровности, из-за того, что сферы, которыми моделируется нить, недостаточно пересекают друг друга. Но при этом если выбрать слишком маленький шаг разбиения, то для расчёта потребуется много времени. В данной работе диаметр нити и шаг разбиения имеют соответственно следующие значения: $D = 1, h = \frac{D}{5}$.

Диаметр нити и шаг разбиения остается постоянным при дальнейшей работе алгоритма, данные переменные не умножаются ни на какие коэффициенты.

3.2.2. Алгоритм движения по градиенту

Как говорилось ранее, движение по градиенту необходимо для того, чтобы нить выпрямлялась и чтобы длина нити уменьшалась. Для этого записывается формула энергии для каждой точки переплетения:

$$E = (\vec{r}_i - \vec{r}_{i-1})^2 + (\vec{r}_{i+1} - \vec{r}_i)^2, \quad (16)$$

где \vec{r}_i – радиус-вектор i -ой точки, \vec{r}_{i-1} – радиус-вектор $(i - 1)$ -ой точки, \vec{r}_{i+1} – радиус-вектор $(i + 1)$ -ой точки. Радиус-вектор является функцией трех координат, то есть:

$$\vec{r} = \vec{r}(x, y, z). \quad (17)$$

Для того чтобы найти градиент энергии, нужно найти её производную по каждой координате:

$$\frac{\partial E}{\partial x_i} = -2(x_{i+1} - 2x_i + x_{i-1}) \quad (18)$$

$$\frac{\partial E}{\partial y_i} = -2(y_{i+1} - 2y_i + y_{i-1}) \quad (19)$$

$$\frac{\partial E}{\partial z_i} = -2(z_{i+1} - 2z_i + z_{i-1}) \quad (20)$$

Полученный градиент направлен в сторону увеличения энергии каждого узла, поэтому для достижения минимума энергии надо брать значения градиента с обратным знаком.

Для того чтобы минимизировать возникновение пересечений каждую компоненту градиента умножают на следующий коэффициент: $G_k = \frac{1}{4}$. Это позволяет контролировать движение нити с помощью градиента. В результате каждая координата нити пересчитывается по следующим формулам:

$$x_i = x_i + \frac{1}{2}(x_{i+1} - 2x_i + x_{i-1}) \quad (21)$$

$$y_i = y_i + \frac{1}{2}(y_{i+1} - 2y_i + y_{i-1}) \quad (22)$$

$$z_i = z_i + \frac{1}{2}(z_{i+1} - 2z_i + z_{i-1}) \quad (23)$$

После того как были найдены новые значения компонент переплетения, можно переходить к следующему этапу алгоритма.

3.2.3. Деформация ячейки

При деформировании элементарной ячейки, с помощью которой задается двоякопериодическое переплетение, её площадь остается

постоянной, поэтому всегда можно найти такую форму ячейки, что суммарная длина переплетения будет минимальна.

Квадрат длины нити можно записать следующим образом:

$$S = \sum_{i=1}^N x_i^2 + y_i^2. \quad (24)$$

Необходимо найти такую форму ячейки, что значение S будет минимальным.

Так как само переплетение инвариантно относительно поворотов, то можно считать, что ячейка деформируется только при растяжении и сдвиге. Тогда деформацию можно задавать матрицей следующего вида:

$$A = \begin{pmatrix} \alpha & \beta \\ 0 & \frac{1}{\alpha} \end{pmatrix}, \quad (25)$$

причем $\det A = 1$, $\alpha > 0$ – коэффициент растяжения, β – коэффициент сдвига. Визуализация деформации элементарной ячейки представлена на Рис. 32.

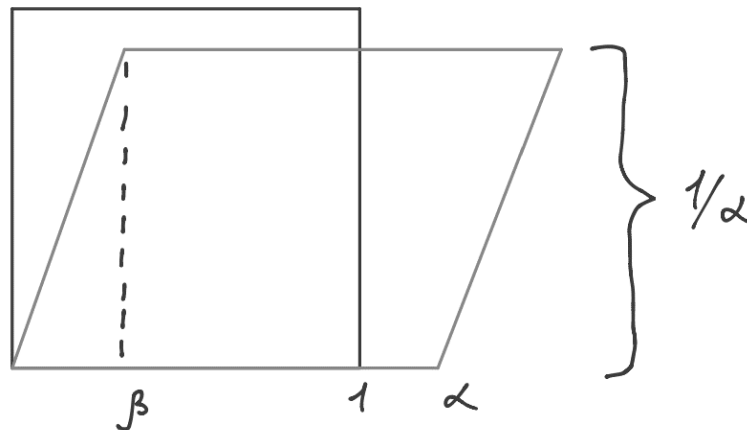


Рис. 32. Деформация элементарной ячейки

Координаты в деформированной ячейке записываются через матрицу деформации:

$$\begin{pmatrix} \tilde{x} \\ \tilde{y} \end{pmatrix} = A \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \alpha x + \beta y \\ \frac{y}{\alpha} \end{pmatrix}. \quad (26)$$

Тогда S можно переписать следующим образом, причем данная функция становится функцией от коэффициентов растяжения α и сдвига β :

$$S(\alpha, \beta) = \sum_{i=1}^N (\alpha x_i + \beta y_i)^2 + \frac{y_i^2}{\alpha^2} = S_1 \alpha^2 + 2S_2 \alpha \beta + S_3 \beta^2 + \frac{S_3}{\alpha^2}, \quad (27)$$

где $S_1 = \sum_{i=1}^N x_i^2$, $S_2 = \sum_{i=1}^N x_i y_i$, $S_3 = \sum_{i=1}^N y_i^2$. Для того чтобы найти минимальное значение S , необходимо найти экстремумы этой функции.

Условие экстремума:

$$\frac{\partial S}{\partial \alpha} = 0; \quad \frac{\partial S}{\partial \beta} = 0. \quad (28)$$

Подставив выражение для S , получаются следующие выражения:

$$\begin{cases} S_1 \alpha + S_2 \beta - \frac{S_3}{\alpha^3} = 0 \\ S_2 \alpha + S_3 \beta = 0 \end{cases}. \quad (29)$$

Решив данную систему, получаются выражения для коэффициентов:

$$\alpha = \left(\frac{S_3^2}{S_1 S_3 - S_2^2} \right)^{\frac{1}{4}}, \quad (30)$$

$$\beta = -\frac{S_2}{S_3} \alpha. \quad (31)$$

Вычислив новые координаты с новыми значениями коэффициентов растяжения и сжатия, получается новая форма элементарной ячейки, длина нити также уменьшилась.

3.2.4. Умножение на масштабный коэффициент

Следующим этапом алгоритма является умножение на масштабный коэффициент. Именно таким образом моделируется затягивание нити. В ходе всей работы алгоритма есть заданный параметр, которые не меняет своего значения: $\alpha = 0.01$.

Такой коэффициент необходим при регулировании масштабного коэффициента. Выражение для масштабного коэффициента выглядит следующим образом:

$$K = 1 - \frac{\alpha}{2^k}, \quad (32)$$

k – номер попытки удалить пересечения, то есть в начале алгоритма $k = 0$ и $K = 0.99$.

Регулировать масштабный коэффициент необходимо в том случае, если после его применения возникают новые пересечения, которые невозможно удалить. Для этого сначала номер попытки k увеличивается на единицу и затем вводится вспомогательный масштабный коэффициент:

$$K_h = 1 + \frac{\alpha}{2^k}. \quad (33)$$

Такой коэффициент вводится для того, чтобы после возникновений пересечений немного увеличить ячейку и избавиться от пересечений. После увеличения ячейки проверяется наличие пересечений. Если пересечения всё ещё остались, то их снова надо пробовать удалять. Номер попытки и вспомогательный коэффициент увеличивается до тех пор, пока пересечения не удалятся или пока масштабный коэффициент не примет значение $K = 0.9999$.

3.2.5. Удаление пересечений

Для того чтобы вычислить между какими точками двоякопериодического переплетения возникли пересечения, необходимо заполнить массив соседей. Для этого надо вычислить попарное расстояние между каждыми точками и сравнить с желанным расстоянием. Желаемое расстояние равно диаметру нити с малым параметром eps . Малый параметр вводится для того, чтобы при удалении пересечений оставлять небольшой запас между сферами, и на следующей итерации возникало как можно меньше пересечений.

После того как вновь заполнен массив соседей, надо найти число пересечений. Число пересечений LO равно половине от числа ненулевых элементов массива соседей. Также на данном этапе необходимо найти число пропускания. Это число равно количеству соседей от каждой точки, пересечения с которыми следует сохранить. Это делается, потому что нить моделируется как непрерывная кривая. Число пропускания вычисляется по следующей формуле:

$$Skipped = round\left(\frac{\pi D}{h}\right). \quad (34)$$

После нахождения числа пропускания необходимо обновить массив соседей, то есть в каждой строке массива удалить вершины, номера которых находятся в следующем промежутке: $[i - Skipped, i + Skipped]$, где i – номер рассматриваемой вершины.

Если массив соседей не нулевой, то необходимо удалять пересечения. Для этого надо найти оптимальный вектор перемещений для каждой вершины, у которой есть пересечения. Чтобы найти этот вектор, надо для каждого соседа найти модуль расстояния, на которое надо переместить вершину, чтобы удалить данное пересечение. Также необходимо найти направление перемещение. Для этого надо найти прямую, соединяющую две вершины, у которых есть пересечения, и её проекции на каждую ось нормировать. В результате получены направления перемещения для данной пары соседей.

После того как получены направления перемещений для всех соседей данной точки, эти направления надо сложить. В итоге получить оптимальное направление пересечение точки для каждой оси. Для того чтобы найти длину сдвига, нужно найти максимум расстояния, на которое надо отодвинуть точку, чтобы удалить пересечение:

$$R = \frac{D + eps - d}{2}. \quad (35)$$

d – текущее расстояние между точками. Значение расстояния делится на два, так как необходимо симметрично отодвинуть друг от друга обе точки, участвующие в пересечении.

В результате вектор перемещения равен:

$$u_x = (0.25 + eps)R_{max}V_x \quad (36)$$

$$u_y = (0.25 + eps)R_{max}V_y \quad (37)$$

$$u_z = (0.25 + eps)R_{max}V_z \quad (38)$$

где u_x, u_y, u_z – компоненты перемещения узла, R_{max} – максимальная длина сдвига, V_x, V_y, V_z – компоненты сдвига.

Алгоритм удаления пересечений следует повторять до тех пор, пока все пересечения не будут удалены.

ГЛАВА 4. РЕЗУЛЬТАТЫ

Первым результатом данной работы является алгоритм упаковки. На Рис. 33 представлены визуализации двоякопериодических переплетений в их начальном состоянии.

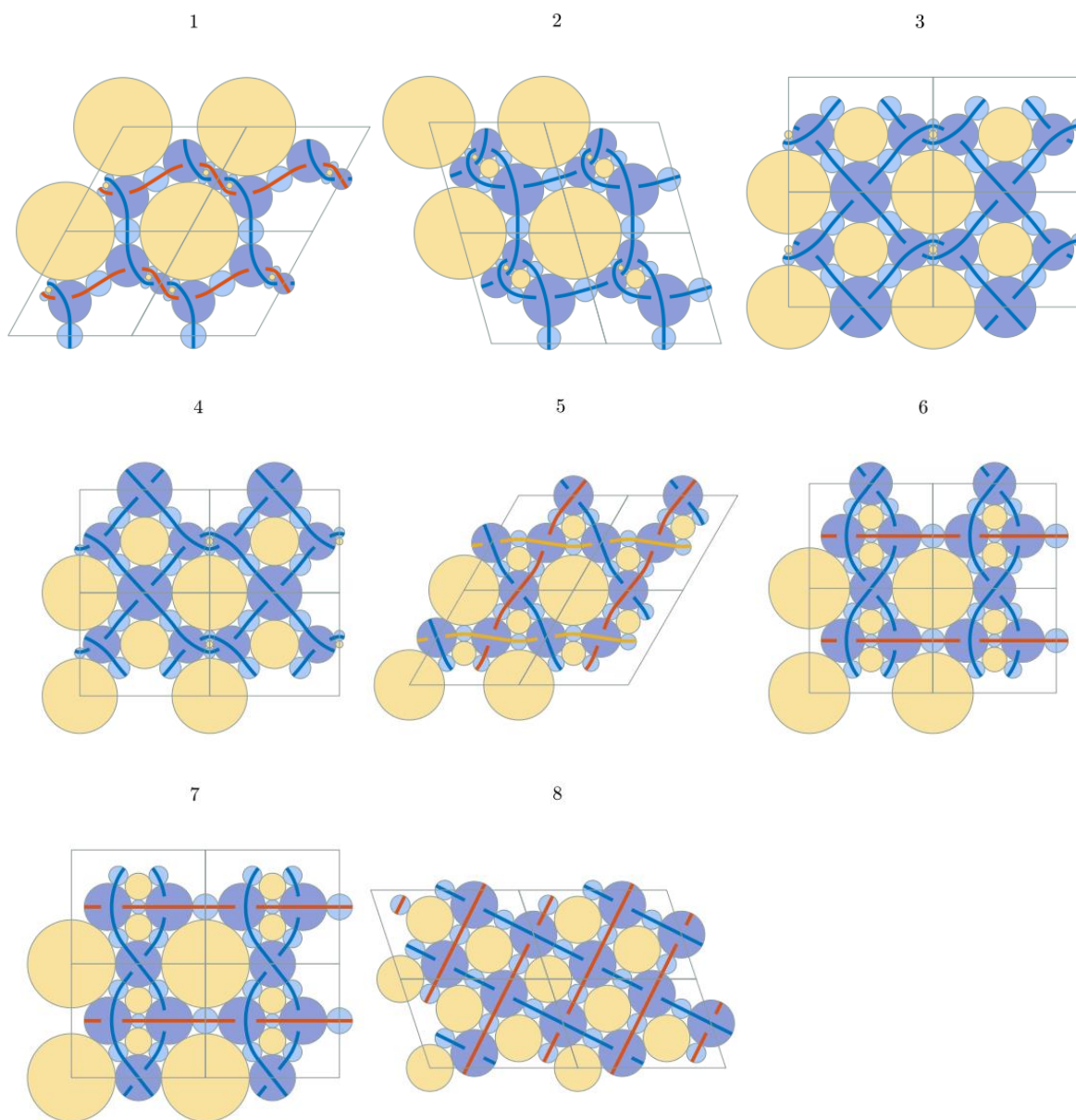


Рис. 33. Визуализация двоякопериодических переплетений

На Рис. 34 изображен пример решетки двоякопериодического переплетения.

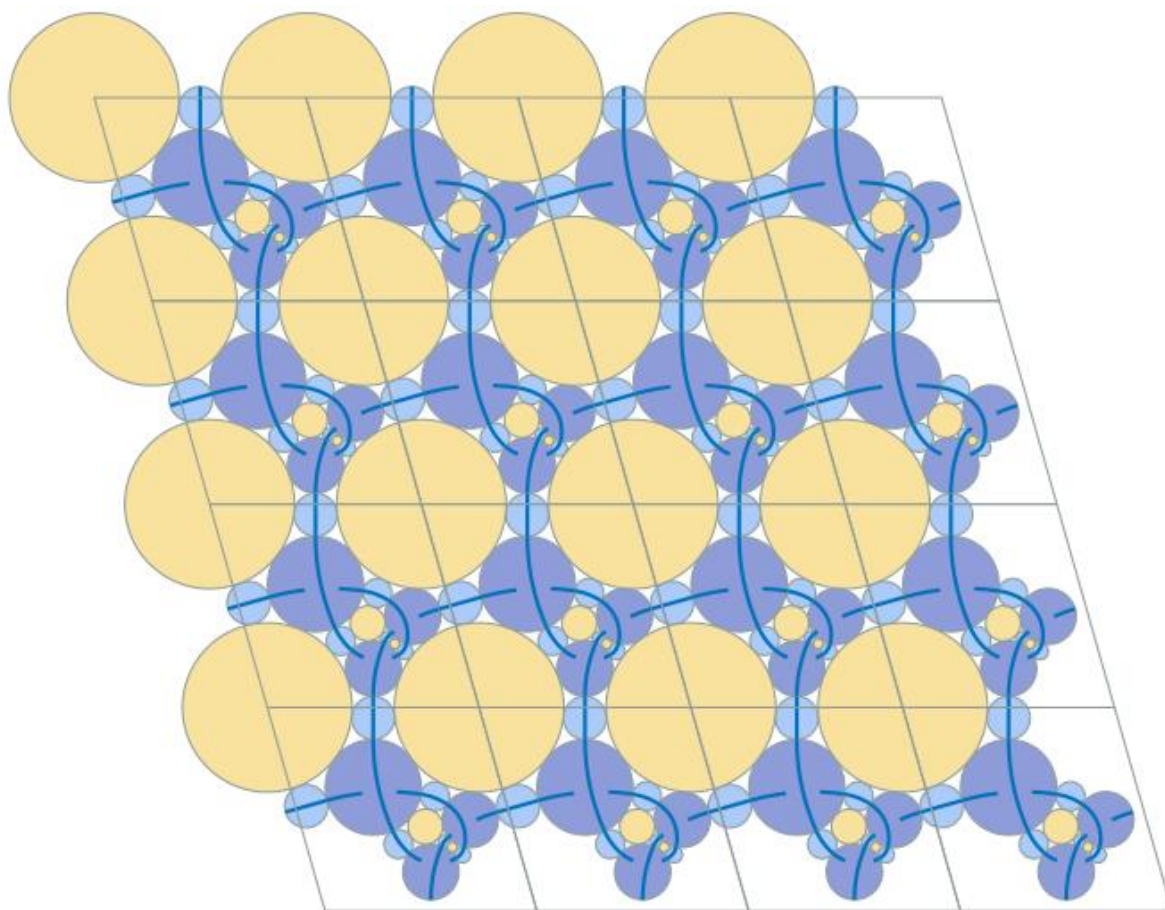


Рис. 34. Пример решетки двоякопериодического переплетения

Параллельно с данной работой, решалась задача получения таблиц всевозможных двоякопериодических переплетений. Именно эти таблицы планируется очистить от повторов с помощью алгоритма затягивания. На Рис. 35 представлен пример части таблицы для двоякопериодических переплетений с двумя перекрестками. На Рис. 36 – для переплетений с тремя перекрестками.

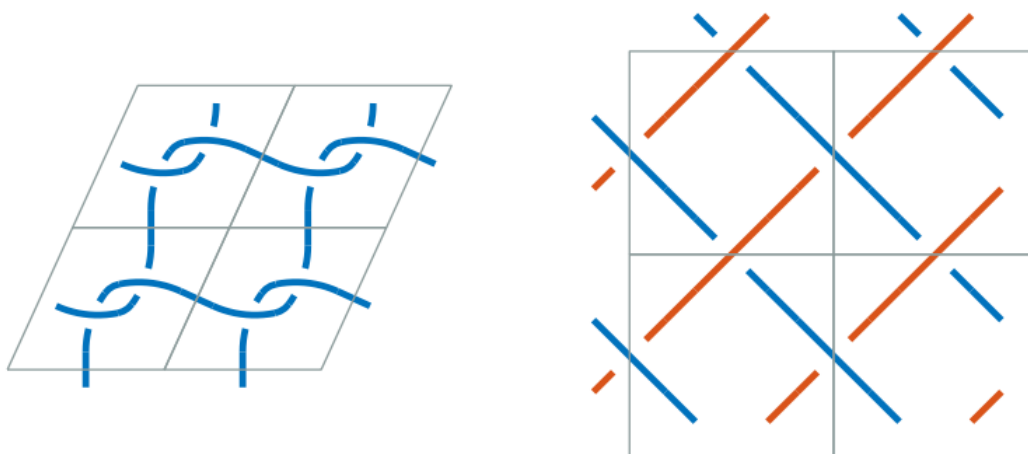


Рис. 35. Двоякопериодические переплетения с двумя перекрестками

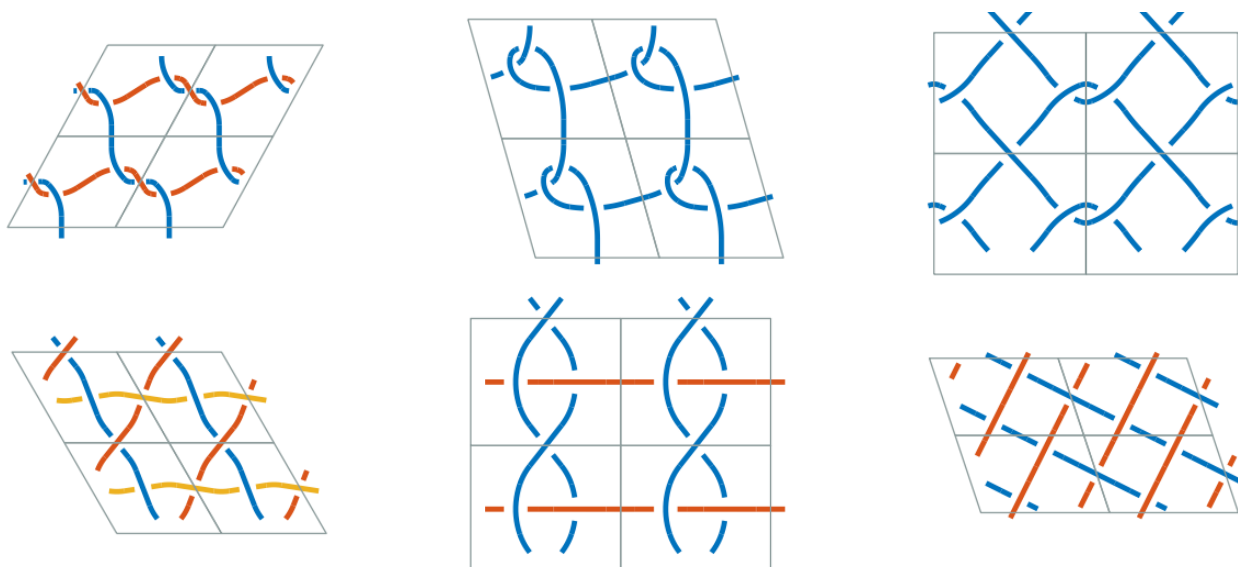


Рис. 36. Двойкопериодические переплетения с тремя перекрестками

На Рис. 37 представлен результат работы для случайно сгенерированного двойкопериодического переплетения.



Рис. 37. Результат работы алгоритма для случайно сгенерированного двойкопериодического переплетения ($D = 1$)

Для наглядности движения нитей, на Рис. 38 изображено идеальное состояние переплетения с диаметром нитей в два раза меньше действительного.

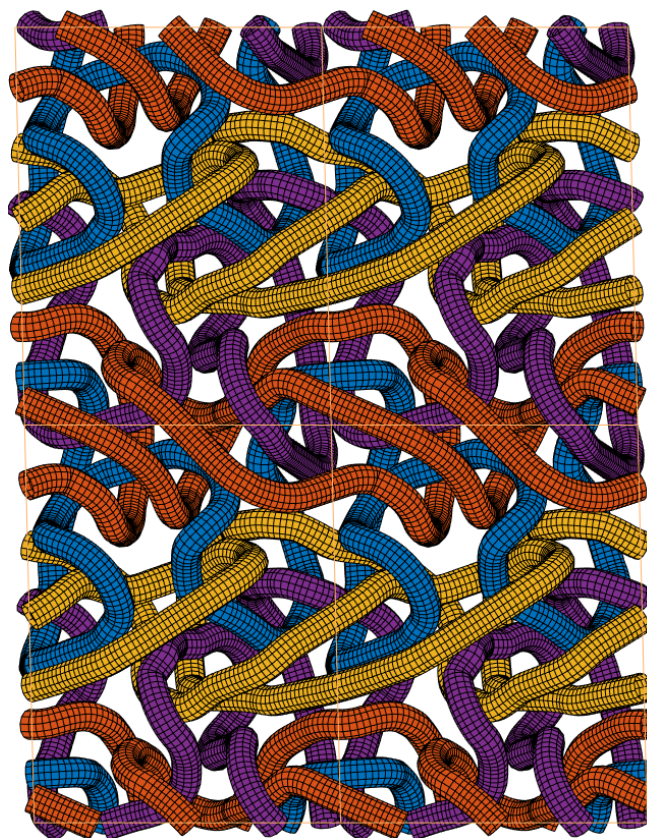


Рис. 38. Результат работы алгоритма для случайно сгенерированного двоякопериодического переплетения ($D = \frac{1}{2}$)

Идеальные состояния двоякопериодических переплетений с двумя перекрестками изображены на Рис. 39 (первое переплетение на Рис. 35) и на Рис. 40 (второе переплетение на Рис. 35).

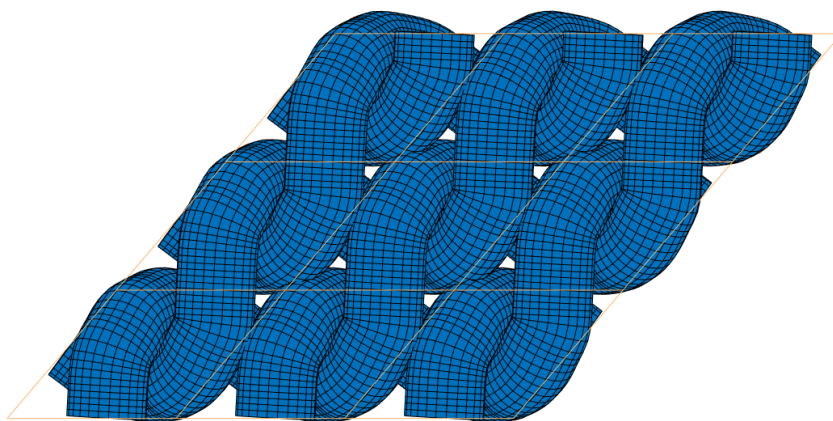


Рис. 39. Идеальное состояние двоякопериодического переплетения с двумя перекрестками

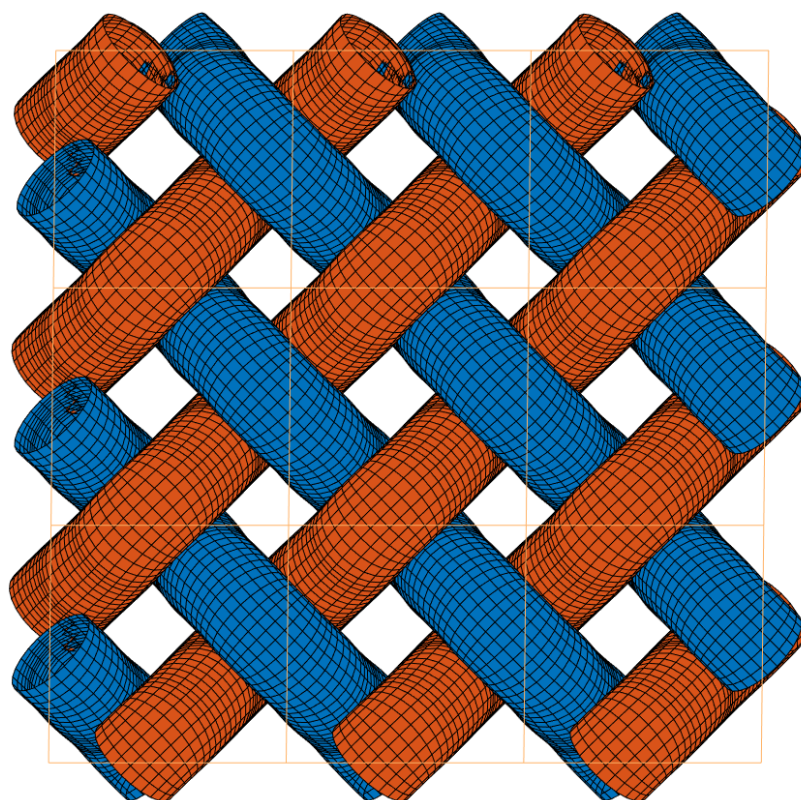


Рис. 40. Идеальное состояние двоякопериодического переплетения с двумя перекрестками

Ниже представлен результат работы алгоритма для каждого двоякопериодического переплетения с тремя перекрестками. Первый рисунок для каждого варианта соответствует реальному диаметру нити, на втором рисунке диаметр нити уменьшен в два раза, это нужно для того, чтобы убедиться, что переплетение нитей в ходе работы не нарушено.

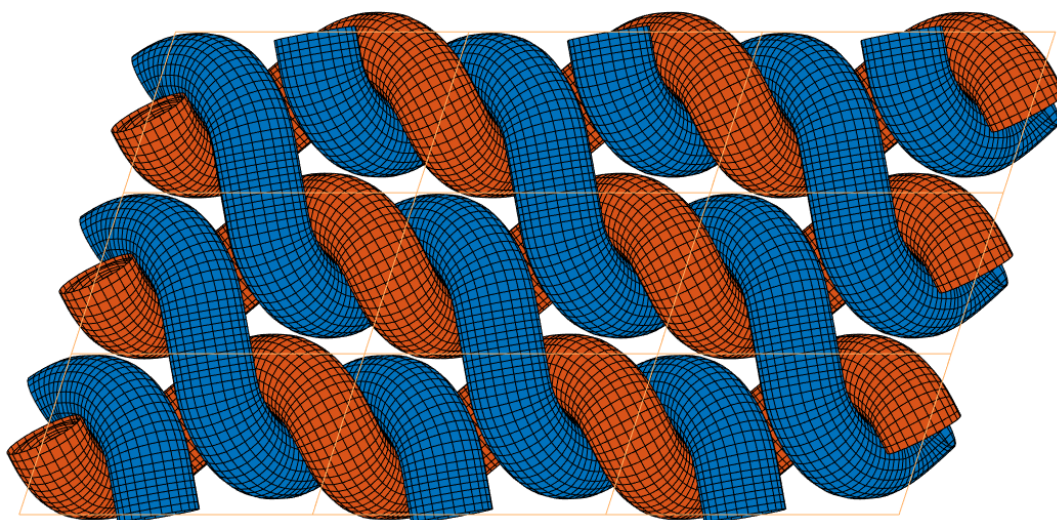


Рис. 41. Идеальное состояние двоякопериодического переплетения с тремя перекрестками
($D = 1$)

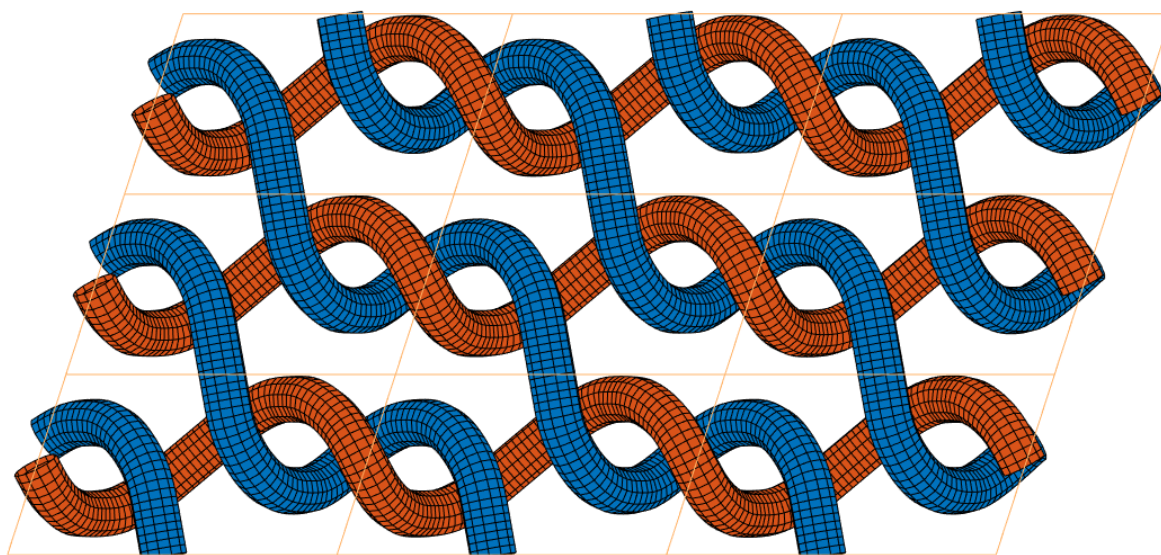


Рис. 42. Идеальное состояние дwoякопериодического переплетения с тремя перекрестками
 $(D = \frac{1}{2})$

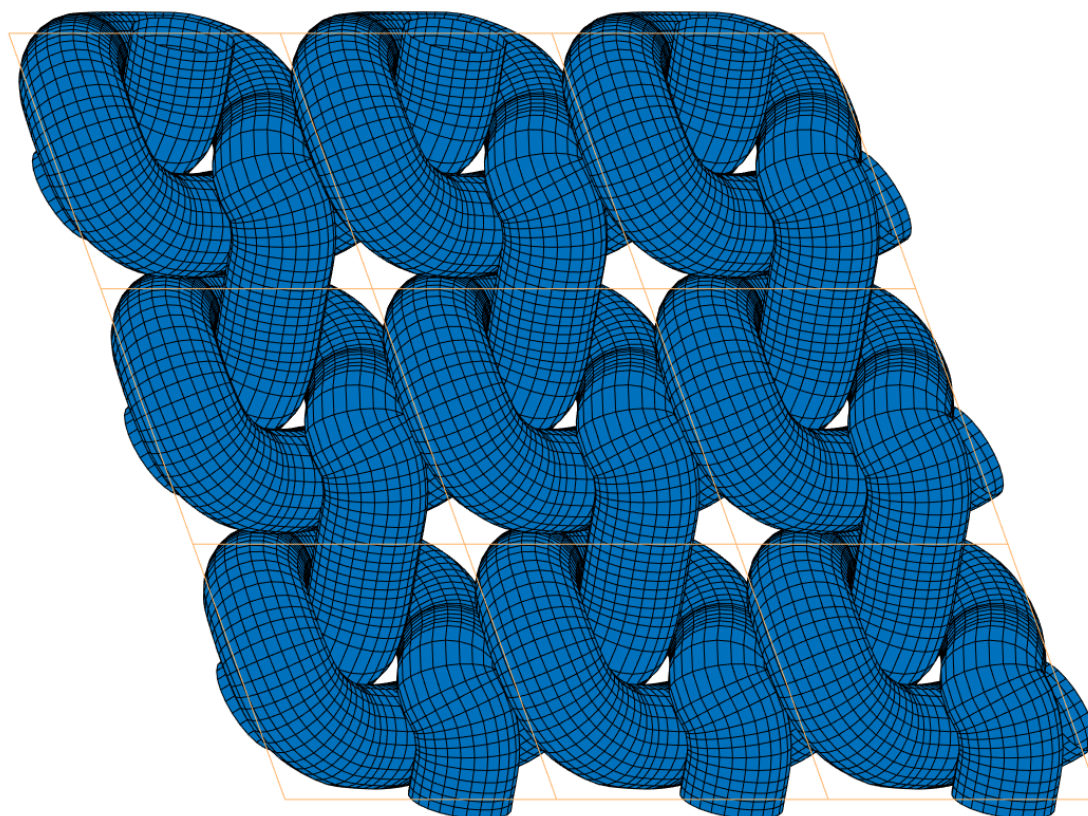


Рис. 43. Идеальное состояние дwoякопериодического переплетения с тремя перекрестками
 $(D = 1)$

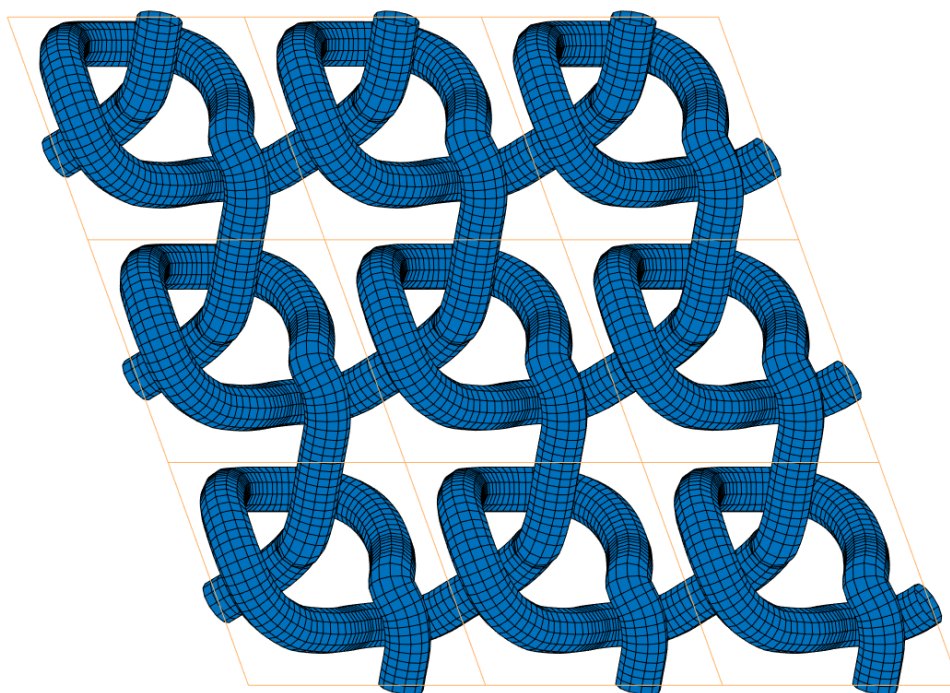


Рис. 44. Идеальное состояние двоякопериодического переплетения с тремя перекрестками
($D = \frac{1}{2}$)

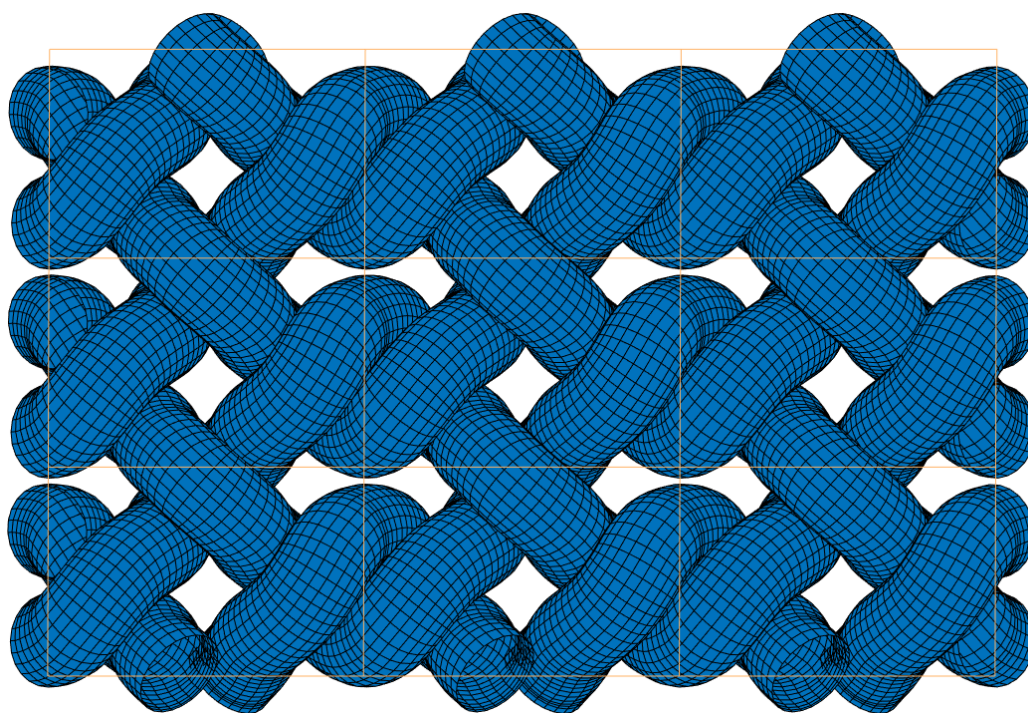


Рис. 45. Идеальное состояние двоякопериодического переплетения с тремя перекрестками
($D = 1$)

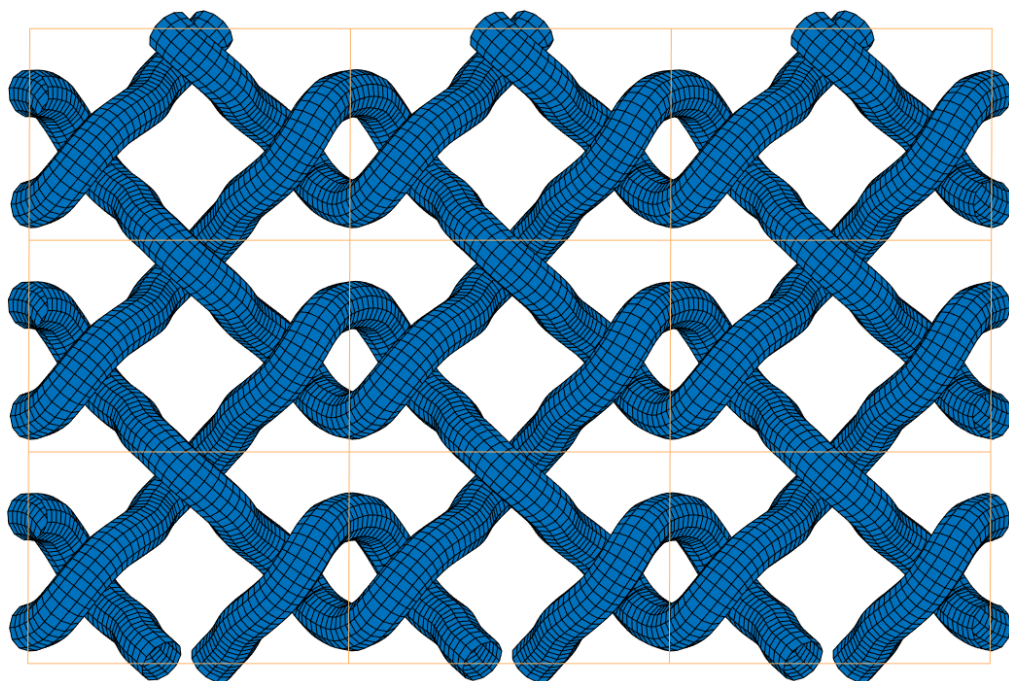


Рис. 46. Идеальное состояние двоякопериодического переплетения с тремя перекрестками
($D = \frac{1}{2}$)

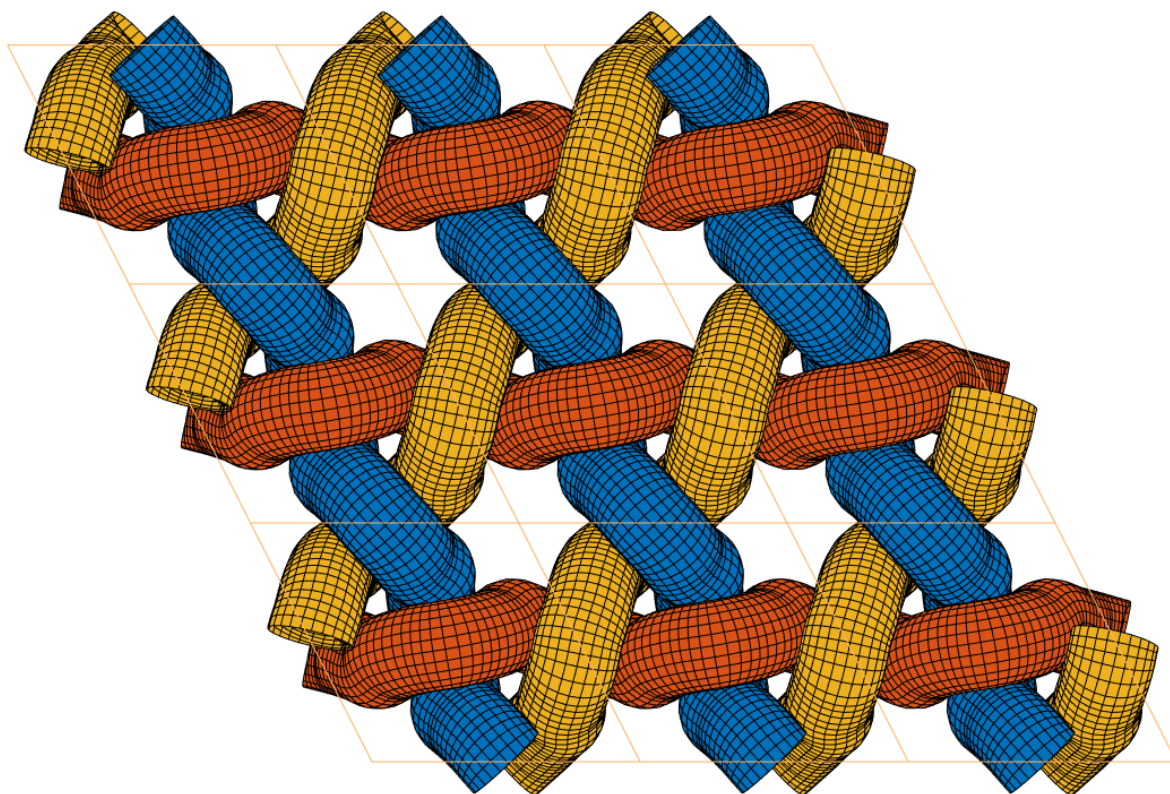


Рис. 47. Идеальное состояние двоякопериодического переплетения с тремя перекрёстками
($D = 1$)

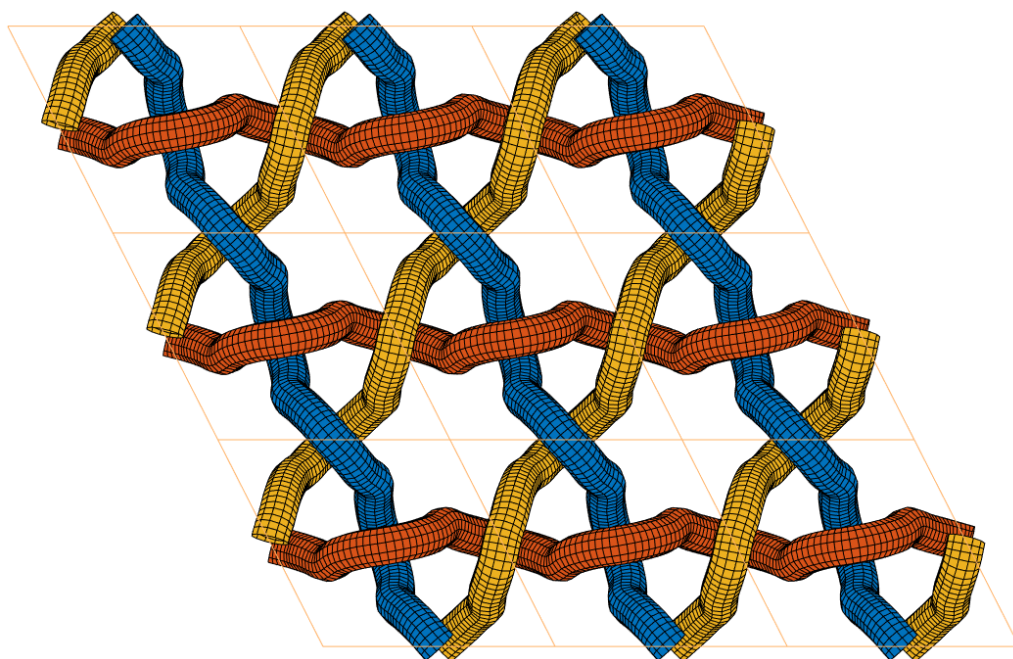


Рис. 48. Идеальное состояние двоякопериодического переплетения с тремя перекрёстками
 $(D = \frac{1}{2})$

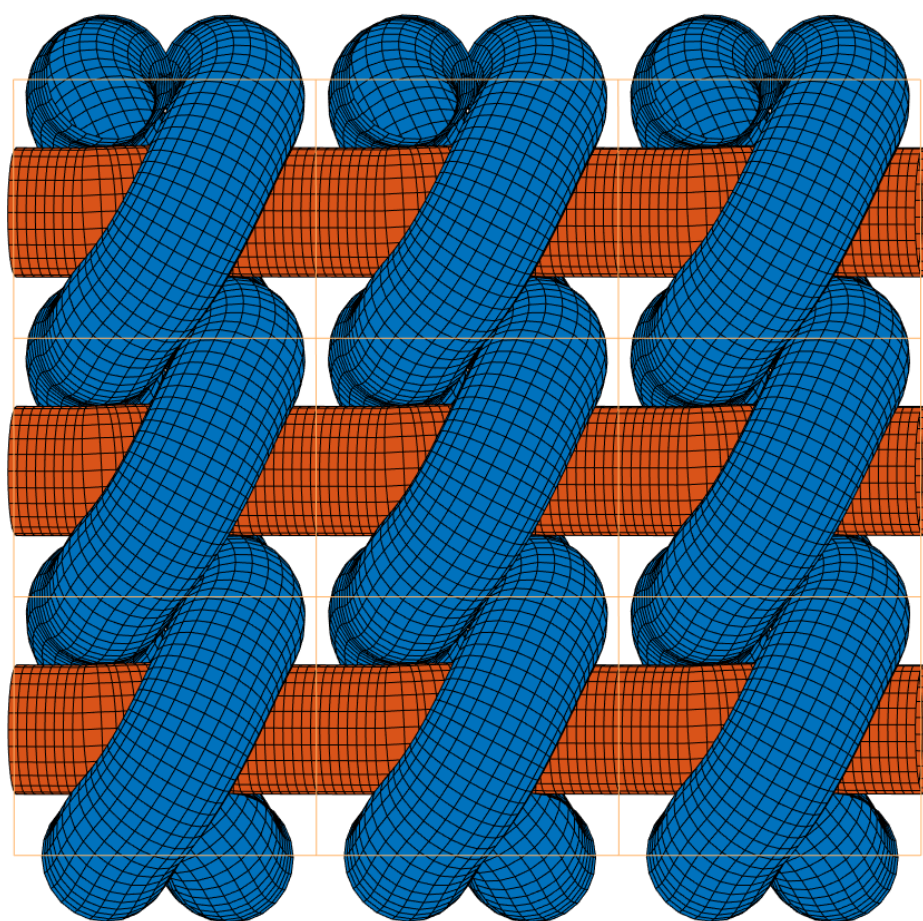


Рис. 49. Идеальное состояние двоякопериодического переплетения с тремя перекрёстками
 $(D = 1)$

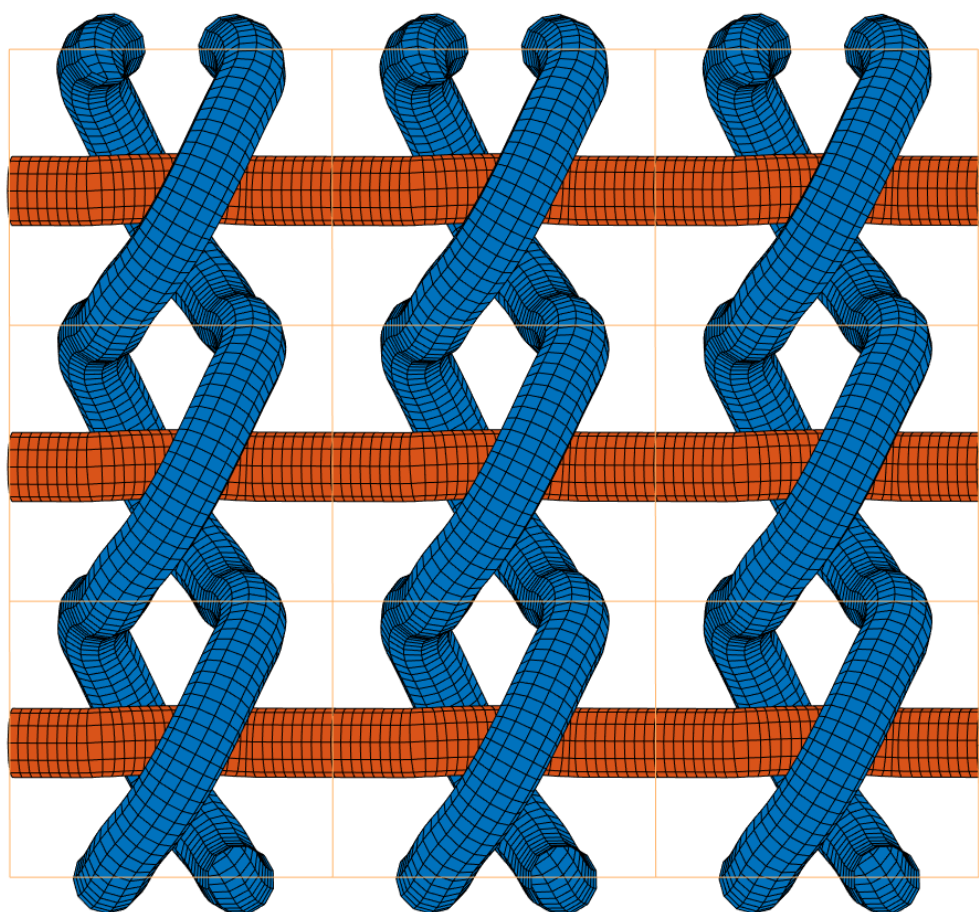


Рис. 50. Идеальное состояние двоякопериодического переплетения с тремя перекрёстками
 $(D = \frac{1}{2})$

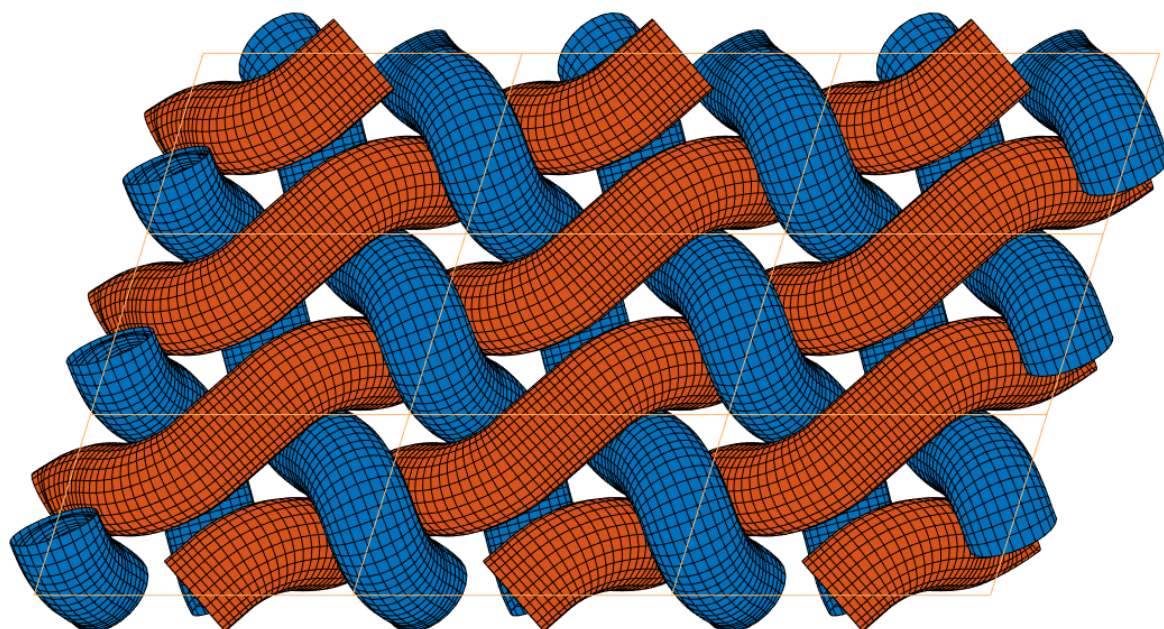


Рис. 51. Идеальное состояние двоякопериодического переплетения с тремя перекрёстками
 $(D = 1)$

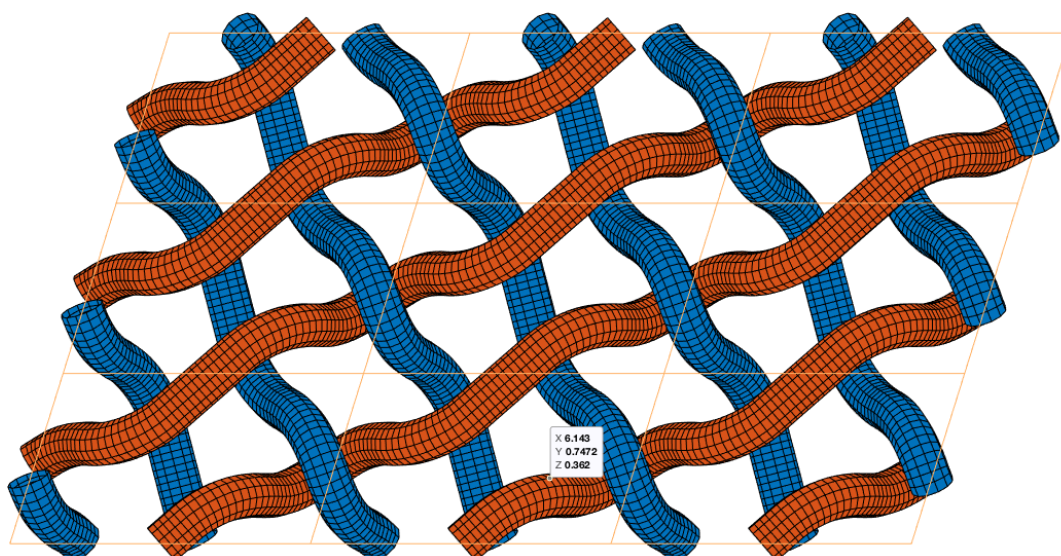


Рис. 52. Идеальное состояние двоякопериодического переплетения с тремя перекрёстками
 $(D = \frac{1}{2})$

Видно, что переплетение нитей для каждого случая сохраняется исходным и что каждая нить сохраняет свою первоначальную форму. Для каждого двоякопериодического переплетения элементарная ячейка деформируется по разному для того, чтобы достичь минимальной длины нити, то есть максимальной плотности.

ЗАКЛЮЧЕНИЕ

В ходе данной работы был разработан алгоритм для нахождения максимальной плотности двоякопериодических переплетений. Для этого были решены следующие задачи:

1. Создана программа, которая представляет начальное переплетение в матричном виде;
2. Создана программа, которая визуализирует начальное переплетение с помощью метода упаковки окружностей;
3. Разработан алгоритм затягивания двоякопериодических переплетений, основанный на алгоритме затягивания для узлов «SONO».

Результатами работы является визуализация плотнейших двоякопериодических переплетений и значение минимальной длины нити, необходимой для создания данного переплетения.

Дальнейшим развитием данной работы может являться работа с таблицами перечисления двоякопериодических переплетений, а именно проверка каждого переплетения на повторы. После этого можно начинать изучать механические свойства переплетений и использовать их на практике.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Звонкин А. К., Ландо С. К. Графы на поверхностях и их приложения. М.: МЦНМО, 2010, 480 с.
2. Кристофидес Н. ТЕОРИЯ ГРАФОВ. АЛГОРИТМИЧЕСКИЙ ПОДХОДМ.: Мир, 1978, 432 стр.
3. Мантуров В.О. Теория узлов. – Москва-Ижевск: Институт компьютерных исследований, 2005, 512 стр.
4. Прасолов В. В., Сосинский А. Б. Узлы, зацепления, косы и трехмерные многообразия. М.: МЦНМО, 1997, 352 с.
5. Cromwell P. Knots and links. Cambridge: Cambridge university press. 2004. 350 p.
6. Katritch V., Bednar J., Michoud D., Sharein R. G., Dubochet J. and Stasiak A., 1996, Geometry and physics of knots., Nature 384, 142-145.
7. Kauffman L. Virtual knot theory. European Journal of Combinatorics, 1999, 20, 663–690.
8. Kauffman, L. H., “Knots and Physics”, 2nd edn., World Scientific, London, UK, 1993.
9. Moffat H. K., 1990, The energy spectrum of knots and links., Nature 347, 367-369.
10. Stasiak A., Katritch V., Kauffman L. H. IDEAL KNOTS. Series on Knots and Everything - Vol. 19., CHAPTER 2, 1998.
11. Мешков В. Р., Мишунин А. С., Омельченко А. В. Перечисление альтернированных k -танглов. Настоящий сборник.
12. Мешков В. Р., Мишунин А. С., Омельченко А. В. Быстрая генерация неизоморфных вложенных букетов окружностей. Настоящий сборник.
13. Beardon A.F., Stephenson K., The uniformization theorem for circle packings, Indiana Univ. Math. J. 39 (1990) 1383– 1425.
14. Collins Charles R., Stephenson Kenneth, A circle packing algorithm, Department of Mathematics, University of Tennessee, 37996-1300 Knoxville, TN, USA

15. Grosberg, P., Shape and Structure in Textiles, *J. Textile Inst.* 57, T383–T394 (1966).