Crystallographic Surfaces

Joel Martin

Contents

Dedication

To Vanessa Robins and Joan Licata, for their never ending patience. To all the other honours students, I still have no idea what any of you are doing but it all sounds very clever. To my parents; I remember when you first taught me to count. And finally to Jess Liang, Alex McKay and Tess Johnson, for the various long walks and aimless discussions.

Where don't we find crystals?

Crystals nets are used extensively in the material sciences as models for highly symmetric atomic structures seen in nature. Such atomic structures were first postulated by scientists studying crystal formations. These formations possessed integer ratios in the angles between their faces and it was from these measurements that chemists predicted the symmetries of their atomic structures [6]. In the time since, more sophisticated research methods have been developed for determining these symmetries of crystals, such as x-ray crystallography, giving rise to the field of crystallography. The modern concept for a crystal now refers to material that produces sharp lines in diffraction experiment, which includes a wide class of materials such as salts and metals [6].

There is in particular a strong link between the symmetries of a crystal net and the emergent physical properties of the material [6]. Scientists determine the crystal structure of a material by comparing the result of x-ray diffraction to those expected from different hypothesised structures [6]. For this method to work scientists must have an effective means of enumerating possible crystal structures. There are any number of methods for enumerating crystal structures all of which present their own challenges and rewards [5].

EPINET is one such method developed at the ANU. This process requires the selection of a Triply Periodic Minimal Surfaces sitting in Euclidean 3-space, which acts as a filter over the range of possible crystals [7]. This however leaves an important open question as to how we quantify which crystals are left out by such a filter [6].

Motivated by this problem we here set up the idea of a crystallographic surface as a discrete model for TPMS. We then proceed to set up a theory for the enumeration of crystallographic surfaces modelled closely on T. Sunada's theory for the enumeration of crystal structures. This framework serves as both a two-dimensional extension of Sunada's theory as well as a potential framework within which the inverse problem of EPINET can be studied.

Theorems in Chapter 2 are from Sunada's book topological crystallography [8]. Theorems in Chapter 3 are from either Graphs on Surfaces [2], Dress's paper [1] or Olaf's paper [4]. Chapter 4 is original work developed based on all of the above sources with imput and guidence from Vanessa Robins.

Chapter 1

Motivation

In this chapter we lay out the basic ideas of crystallography, crystal enumeration with particular focus on EPINET in order to motivate our algorithm. We give no formal definitions or theorems in this chapter but instead focus on examples; building up the broader intuition that will motivate us through the mathematical work in later chapters.

1.1 The Adventures of Agnes, Barry and Clara, Episode I

In a far off land...

Barry was feeling very stressed. For reasons best known only to himself and Gauss he had many decades ago set up a farm in the hyperbolic plane. Farming open balls had been all very well at the time but now Barry was starting to get old, and in all this time a whole industry had grown out of his venture. Suddenly there was increasing demand for a whole range of new shapes and hyperbolic farming was starting to outpace Barry's skill set. It became apparent that he would need to divide his farm up into paddocks to separate the various shapes, but what's more he'd need a design that could grow with increasing demand. After much thought he decided he would attempt to construct a periodic design, something that could be repeated over and over as the size of the farm needed to increase. Designing this however was no easy task for a two-dimension being like Barry and so he decided to do what all smart people do and ask around for some good ideas.

Barry had a sister, whose name was Agnes for reasons Barry found too horrible to contemplate. She owned a farm in a hyperbolic surface of her own and he appreciated the odd letter they'd often send each other musing the nature of the hyperbolic farming industry. Barry decided in his next letter to write up a description of the problem. A few days later he received a reply and it turned out she was in exactly the same predicament but had been given some rather useful advice by a friend of a friend of hers named Clara. Clara was apparently an expert at this kind of reasoning and was always willing to offer out advice, so Agnes strongly encouraged Barry to send her a letter.

Barry sent her back a letter thanking her profusely and set to write this Clara a letter. Looking up her details in the directory he discovered something quite astonishing; she lived in a graph! Barry couldn't possibly imagine how a 1-dimensional creature could possibly solve a two-dimensional problem such as his. But he decided to approach her without prejudice and he wrote the letter. Despite his resolution while writing out the letter he couldn't help but ask her how she had been able to imagine his sister's two dimensional problem. He waited for about a good week before he got a reply and what he got was quite astonishing...

To be continued...

1.2 What is a crystal?

To a material scientist a crystal is a three-dimensional arrangement of atoms possessing three independent directions of translational symmetry forming a lattice symmetry group [8]. There are actually any number of ways to imagine a crystal [6]. One such method is using sphere packings [5]. (Picture of sphere packing) Systematic enumeration of crystalline structures has been the focus of a large body of mathematical work and for some time extensive work was done attempting to determine the possible sphere-packings of $\mathbb{R}^3$ [5]. The approach focused on here uses graphs.

(Picture demonstrating this)

In this vein, a crystal is not only a graph, but a graph embedding in $\mathbb{E}^3$. The important features of this embedding are both the positions of the vertices and the isometries of the graph embedding [8]. As stated above crystals are required to have three independent directions of translational symmetry. This symmetry group is referred to as the *lattice group* [8]. We can from this symmetry group determine the fundamental domain, the smallest section of the graph embedding from which the whole embedding can be reconstructed from the lattice symmetries.

We should stop to take note here that the crystals we are studying are in fact infinite continuations of real world crystal structures. We make this choice in favour of preserving the symmetries of crystals.

Example 1.1. By far the simplest crystal to consider, and one that we will return to frequently, is simple cubic [8]. (Picture of simple cubic.) This can be described as the set of coordinates in $\mathbb{R}^3$ with at least two integer coordinates. It's lattice group is simply $\mathbb{Z}^3 \subset \mathbb{R}^3$. It's fundamental domain is simply the unit cube. When attempting to understand some crystallographic problem it is often helpful to first consider the case of simple cubic in which results can usually be easily visualised.

Simple cubic only has one vertex in its unit cell however which can present a problem when attempting to visualise properties that are complicated by the crystal having more than one such vertex. In this case a simple example of a crystal with two vertices in its unit cell is diamond. (Picture of diamond) As it's name suggests diamond is the crystal structure of real world diamonds [8].

1.3 How do we find crystals?

So far the description of crystal we have given is quite simple but surprisingly we can ask some very complicated mathematical questions about them. T. Sunada in his book Topological Crystallography outlines some such problems such as random walks on crystals [8]. One problem of much importance in material science is how to enumerate crystals [5]. Note that this can be broken into two components.

First we have enumerating suitable graphs. These crystal graphs, referred to as topological crystals, are infinite graphs with a symmetry group isomorphic to $\mathbb{Z}^3$ identified . Secondly then we have the embedding of this graph. Note that for some given topological crystal there are an infinite number of ways of embedding this graph as a crystal. In some cases this has physical meaning where we find certain crystallographic compounds with the same topological crystal structure manifest different embeddings depending on the nature of the atoms, or the circumstances of the compounds creation [8].

Example 1.2. A good example of this is diamond. Below we show two embeddings of the topological crystal diamond. (Pictures)

In other cases however some embeddings are simply physically unrealistic due to features such as closest neighbours that don't share an edge, or angles between bonds that simply would never occur. This demonstrates the difference between the purely combinatorial or graph properties of crystals and their geometric properties.

T. Sunada in his book Topological Crystallography describes a method of enumerating crystals using the mathematics of covering space theory. This theory covers both aspects of crystal enumeration, the production of topological crystals and the construction of a 'maximally symmetric' embedding for this topological crystal. In particular this method is exhaustive over all topological crystals [8]. This maximally symmetric embedding is however limited by the topological symmetries of the topological crystals.

Mathematicians have explored a huge range of crystal enumeration methods [5]. There have been many approaches to this, in particular a paper by A. Dress gave rise to the idea that crystals could be enumerated through periodic tilings. In his paper Dress outlines the basis of a theory for enumerating periodic tilings of manifold known as Delaney-Dress symbol theory [1]. The basic idea is that the 1-skeleton of a periodic tiling of $\mathbb{R}^3$ is a crystal. Enumerating three-dimensional tilings however has been found to suffer from signifcant combinatorial explosion as well as redundancies produced by multiple tilings having the same 1-skeleton [3].

EPINET was an algorithm developed at ANU which uses two-dimensional tilings to produce crystals. It was inspired by the fact that crystal nets were known to sit on the p, d, and s surfaces. These surfaces are Triply-Periodic Minimal Surfaces in $\mathbb{R}^3$ and the EPINET algorithm was developed to enumerate through all periodic tilings of these surfaces that preserved the three translation symmetries of the surfaces. [7]

(Pictures of these surfaces and the crystals)

By enumerating tilings over these surfaces we can produce any number of crystals. These surfaces are not simply-connected which makes direct enumeration using Delaney-Dress symbol theory complicated. Fortunately DDS theory has the added bonus of providing a framework to lift tilings of a surface to a covering space. The problem can be simplified then by making a choice of covering map from the universal cover of these surfaces, the hyperbolic plane. In particular this covering map need to send isometries to isometries so that periodic tilings get send to periodic tilings. Then with a choice of map, a subgroup of the hyperbolic isometries are identified with the isometries of the surfaces and enumerating tilings is straightforward from there [7].

1.4 Are all crystals surfaces?

So far however there are gaps in the theoretical foundations of EPINET [6]. The gap we will be focusing on is which class of crystals does EPINET enumerate over. To put it simply we want to understand what crystals fundamentally cannot be produced from a TPMS.

We might wonder why it would be so difficult to prove that no embedding of a crystal on a TPMS exists. For our purposes we will only consider 2-cell embeddings, that is embeddings that can be described by 'gluing' faces to the crystal. This problem can then be thought of in two separate components. The first is a purely combinatorial question. If we imagine our crystal as a graph we ask if we can glue faces on a subsection of this graph to form the fundamental domain for a surface stretching over the entire graph.

The second question then is whether the surface formed has an embedding in $\mathbb{E}^3$ that preserves the lattice symmetries of the graph. We could require that this embedding be an extension of the graph embedding of the crystal. This is, however perhaps a little too strict since such embedding have straight edges which would likely get in the way of producing a minimal surface.

It is these two questions we shall focus on solving. The approach presented here was inspired by T. Sunada's method for crystal enumeration. In particular Chapter 2 provides an in depth explanation of Sunada's method along with examples. The idea for this method was that a crystal could be viewed as a Triply-Periodic Graph. Using this idea then we might attempt to extend Sunada's theory to two-dimensions to produce a theory for Triply-Periodic Surfaces. This theory will be developed in Chapter 3 partly based of J. Hopkins book Graphs on Surfaces. Chapter 4 then is dedicated to the development of an 'inverse algorithm' for EPINET using the theory of TPS developed in Chapter 3.

Chapter 2

Sunada's Method

In this chapter we outline Sunada's algorithm for enumeratin topological crystals and producing maximally symmetric embeddings. This chapter is based off Sunada's book topological crystallography [8].

2.1 The Adventures of Agnes, Barry and Clara, Episode II

In a far off land, Barry receives a letter from Clara, explaining how she sees higher dimensions...

Clara grew up with her mother and siblings in the real line. It had been a straightforward life, Clara had two directions to remember. Things changed quite dramatically when she'd had to move to university. It turned out that university folk were much fancier than the people in her home line and their campus was arranged in something called a graph. It was much in many ways like her home town except for these blasted junction points.

Each junction was connected to six corridors, organised into pairs by three colours. Clara's navigation skills were at first no match for this strange new structure. She found that everything made perfect sense if she stuck to only red corridors or only blue but once she mixed colours everything got very confusing. Somehow, inexplicably certain sequences of colours would lead her back to where she started, something she spend many nights contemplating at the university pub.

Over time she noticed a pattern; for each of these looping paths she would always travel along an even number of corridors of each colour. This led her to develop a theory. She imagined three different versions of herself, red, blue and green. Each version of her moved along her hometown in between a series of bus stops. They moved independently of one another however only one could move between bus stops at any time. The even pattern now made perfect sense. Each version had to move forwards and backwards the same number of times for all three to arrive where they started. Suddenly she found navigating her university was a piece of cake.

It was using this idea, she claimed that she began to understand the real world was three dimensional.

To be continued...

2.2 Combinatorial Graphs

We begin by setting our the basics of combinatorial graph theory as layed out in Sunada's work. This is perhaps a little basic, but the set up and terminology is useful for understanding Sunada's algorithm. No proves are reproduced here due to the simplicity of the content.

We begin by defining a graph.

Definition 2.1. A *graph* X is a pair (V, E) equipped with three maps, $r : E \to E$, $o : E \to V$, and $t : E \to V$, such that $o \circ r = t$ and $r \circ r$ gives the identity map.

Elements of V are referred to as *vertices* and elements of E are *directed edges. Undirected edges* are equivalence classes of edges under the relation $e \sim r(e)$. For some edge $e \in E$, we say that $o(e)$ is the *origin* of e, $t(e)$ is the *terminus* of e, and $r(e)$ is the *reflection* of e. e is a *looped edge* if $o(e) = t(e)$. e and e' are parallel edges if o(e) = o(e') and t(e) = t(e').

Definition 2.2. A *combinatorial graph* is a graph that has no parallel edges or looped edges.

In particular the reader will note that the topological crystals we will be dealing with will always be combinatorial, however this won't always be true for the quotient graphs we later introduce.

The following concepts are useful in defining certain topological properties of graphs.

Definition 2.3. The *neighbourhood* N_v of a vertex v is the set of all $e \in E$ such that $o(e) = v$.

Definition 2.4. The *degree* of a vertex v, deg v, is equal to $|N_v|$.

Theorem 2.5. *For a graph X = (V,E) the following holds true*

$$\sum_{v \in V} |N_v| = |E| \tag{2.1}$$

Definition 2.6. A *path* in a graph X is a sequence of edges $\{e_n\}_{n\in\{0,\dots,N\}}$, such that $o(e_i) = t(e_{i-1})$ for all $0 < i \leq N$. By abuse of notation $o(\{e_n\}_{n\in\{0,\dots,N\}}) = o(e_0)$ and $t(\{e_n\}_{\{0,\dots,N\}}) = t(e_N)$. A path is said to *connect* its origin and terminus.

We now introduce some topological concepts for graphs.

Definition 2.7. A *path-connected* graph is a graph such that any two vertices in a graph are connected via a path.

The reader should again note that all crystals considered here are path-connected.

Definition 2.8. A *graph morphism* is a map between the vertices of two graphs that also induces a map between the edges of the graph. A graph automorphism is a bijective graph morphism.

Definition 2.9. An *embedding* of a graph X in $\mathbb{R}^3$ is a map $f : X \to \mathbb{R}^3$, such that every $v \in V$ is mapped to a point $x \in \mathbb{R}^3$, and every $e \in E$ is mapped to a path in T from $f \circ o(e)$ to $f \circ t(e)$.

We now present one definition of a topological crystal we can already understand.

Definition 2.10. A topological crystal is an infinite graph X, with a subgroup $G(X) \sim \mathbb{Z}^3$ of Aut(X) called the lattice group.

This infinite graph is difficult to do calculation on, and so we will introduce the idea of quotient graphs.

Definition 2.11. A *quotient graph* X for a topological crystal T is X = T/H, where H is a subgroup of G(X) called the *generating subgroup*. If H = G(X) then we call X the smallest quotient graph and typically denote it Q.

Example 2.12. We might think back to our example of simple cubic. In this case V = $\mathbb{Z}^3$ and E = $V \times B$, where B is set of standard Euclidean basis vectors and their negatives. For $(v, b) \in E$ we let o((v,b)) = v and t((v,b)) = v + b. Finally r((v,b)) = (v+b,-b), and G(T) is the group of symmetries given by sending v to v + a, and (v,b) to (v+a,b), where $a \in \mathbb{Z}^3$. Note then any two vertices are related via a symmetry and so that T/G(T) is then given by V = 0, $E = 0 \times B^+$, where B^+ is the standard basis for Euclidean space, with o(e) = t(e) = 0 and r(e) = e.

(Picture)

Consider now instead diamond.

(Picture of Diamond)

Diamond's three translational basis vectors are as shown. This results in two vertices in its quotient graph. The quotient graph is shown below.

Sunada's method enumerates topological crystals using covering space theory so it is now helpful to discuss these concepts using the terminology introduced.

First we build up the concept of a covering graph.

Definition 2.13. A covering graph over X is a pair ($\tilde{X}$, p) such that $\tilde{X}$ is a graph and p is a surjective graph morphism from $\tilde{X}$ to X such that p restricted to N_v is a bijection.

The *degree* of a covering map is the cardinality of the preimage of any vertex or edge under the covering map, which can be shown to not depend on the choice of vertex or edge.

Definition 2.14. The deck transformation group $G(\tilde{X})$ of a covering graph is the subgroup of $Aut(\tilde{X})$ that preserves the covering map. The deck transformation is then a faithful group action over $\tilde{X}$.

We might notice that we used the same notation for the deck transformation group as the lattice symmetry group. We did this with good reason and now we make clear the relationship between covering graph theory and topological crystals.

Definition 2.15. A covering graph is said to be normal if the orbits under the group action are identical to the fibres under the covering map.

Theorem 2.16. *A graph with group action G is a normal covering space over its quotient graph X/G with $G(X) = G$.*

Hence we can now think of a topological crystal as a special kind of covering graph.

Example 2.17. Joel remember here to go through examples for Simple Cubic and Diamond.

We now seek to introduce the Galois Correspondence which is used in Sunada's method for topological crystal enumeration. We will first set this up using the fundamental group, and then using graph homology which is used more in practice.

We start then by building up the fundamental group.

Definition 2.18. A geodesic is a path that does not backtrack over itself.

Definition 2.19. A deformation retraction over paths is an equivalence relation where $Aer(e)B \sim AB$ Under this relation all paths are equivalent to a geodesic.

Definition 2.20. The fundamental group with base point x_0, $\pi_1(X, x_0)$ is the group of deformation retraction classes of loops under concatenation, with the empty path as the identity.

Note that for any x_0, $y_0 \in$ X, there is an isomorphism between $\pi_1(X, x_0)$ produced simply by conjugating by a path between the two points so when working from a purely algebraic standpoint we may refer to *the* fundamental group of a graph.

We now build up the Galois correspondence.

Theorem 2.21. *A path starting at x in X can be lifted up uniquely to a path in $\tilde{X}$ provided a starting point $x_0 \in \tilde{X}$ is chosen.*

Theorem 2.22. *The covering map induces an injective group homomorphism between the fundamental groups. This also tells us that $\pi_1(\tilde{X}, x_0)$ is isomorphic to some subgroup in $\pi_1(X, x)$ which we call the generating subgroup H.*

Definition 2.23. For a graph X a universal cover of X is the covering graph corresponding to the trivial subgroup. It follows then that this covering graph is simply connected.

Theorem 2.24. *Any two universal covers of X are isomorphic as graphs.*

We may now state the Galois correspondence.

Theorem 2.25. *There is a bijective correspondence between subgroups of $\pi_1(X, x)$ and covering spaces of X up to graph isomorphism. Furthermore given a normal covering space $p : \tilde{X} \to X$ the following equation holds true*

$$G(\tilde{X}) \cong \frac{\pi_1(X)}{p_*(\pi_1(\tilde{X}))}. \tag{2.2}$$

This theorem can be used to enumerate through all possible covering spaces over a finite graph. This is achieved by first computing the universal cover which is in practice quite easy to do. We can then quotient this universal by the subgroup of the deck transformation group corresponding to the generating subgroup of any covering graph to produce this covering graph.

Example 2.26. Joel remember to include examples here for simple cubic and diamond.

In particular this leads us to a new defintion of a topological crystal.

Definition 2.27. A topological crystal is an infinite-fold covering space of a graph such that the deck transformation group is a free abelian group of rank 3.

Note that since our covering spaces always have abelian deck transformations it is actually often more useful to use graph homology. In particular homology calculations are generally easier than calculations involving the fundamental group.

We begin by setting up the basics of graph homology.

Definition 2.28. The zeroth-order chain group $C_0(X)$ is the free abelian group of vertices. The first-order chain group $C_1(X)$ is the free abelian group of directed edges with $r(e) \sim e^{-1}$.

Definition 2.29. The boundary map of a graph is a homomorphism $C_1(X) \to C_0(X)$ that sends e to $t(e) - o(e)$.

Definition 2.30. The homology group of a graph is the kernel of the first-order boundary map.

We then note the following useful fact.

Theorem 2.31. *The homology group is simply the abelianisation of the fundamental group.*

Furthermore the first-order homology group is a free abelian group and hence we can define the betti number of a graph.

Definition 2.32. The betti number of a graph, $b_1(X)$, is the rank of its homology group.

Theorem 2.33. *For a graph X = (V,E) the following relationship holds true,*

$$|V| - \frac{|E|}{2} = 1 - b_1(X) \tag{2.3}$$

Definition 2.34. An abelian covering graph is a covering graph whose deck transformation group is abelian.

Considering then the Galois correspondence for the fundamental group we find the following definition useful.

Definition 2.35. For a graph X the universal abelian cover of X up to graph isomorphism is the covering graph corresponding to the commutator subgroup of $\pi_1(X, x_0)$.

Furthermore we note that if $\tilde{X}$ is the universal abelian cover then $G(\tilde{X}) \cong H_1(X)$.

We now wish to state the Galois correspondence for homology but first we will require the following two lemmas.

Theorem 2.36. *The covering map $p : \tilde{X} \to X$ induces a homomorphism $p^* : H_1(\tilde{X}) \to H_1(X)$.*

Theorem 2.37. *Let $p : \tilde{X} \to X$ be an abelian covering graph, and $\nu : \pi_0(X, x_0) \to G(X)$ be the homomorphism with kernel $p^*\pi_0(\tilde{X}, \tilde{x_0})$ from the Galois correspondence. If c is a closed path in X, and $\tilde{c}$ is a lift over c to $\tilde{X}$, then $t(\tilde{c}) = \mu(c)o(\tilde{c})$, where μ is the homomorphism $H_1(X) \to G(\tilde{X})$ induced by ν.*

From these the following follows.

Theorem 2.38. *Let $p : \tilde{X} \to X$ be an abelian covering graph, and $\nu : \pi_0(X, x_0) \to G(X)$ be the homomorphism with kernel $p^*\pi_0(\tilde{X}, \tilde{x_0})$. Then $ker\mu = p * H_1(\tilde{X})$, where μ is the homomorphism induced by ν.*

We then refer to the image of this homomorphism as the vanishing subgroup, H. Finally we obtain the Galois correspondence for abelian covering graphs.

Theorem 2.39. *There exists a bijective correspondence between subgroups of $H_1(X)$ and abelian covering spaces of X given by the generating subgroup. In particular*

$$G(\tilde{X}) \cong \frac{H_1(X)}{p_*(H_1(\tilde{X}))}. \tag{2.4}$$

And now we finally have enough theory to start crystal enumeration.

2.3 Enumerating Crystals with Covering Space Theory

We break the process of enumerating crystals into two distinct components. First the enumeration of topological crystals and the construction of a graph embedding. We first begin by enumerating possible starting quotient graphs, X = (V,E). We only consider finite connected graphs due to physical relevance. We start by fixing the betti number, $b_1(X)$. Note that to get a topological crystal we will need to have $b_1(X) \geq 3$.

We claim that we need only enumerate graphs where every vertex is at least degree 3. To justify this we note that we can delete a vertex of degree one without changing the betti number. Additionally since $b_1(X) \geq 3$ and our graph is path-connected we can delete a vertex of degree 2 and replace the two adjoined edges with a single connecting edge without changing the betti number. Hence any graph can be reduced to a graph with minimal degree 3.

Since $\sum_{v \in V} |N_v| = |E|$, we can deduce $3|V| \leq |E|$. Combining this with $|V| - \frac{|E|}{2} = 1 - b_1(X)$ we obtain $|V| \leq 2b_1(X) - 2$, and $|E| \leq 3b_1(X) - 3$. So given a fixed choice of $b_1(X)$ we have a finite number of possible vertices with degree at least 3 we can enumerate through all possible graphs that satisfy the above conditions.

The next step is once we have chosen a quotient graph we compute its first-order homology group. A simple way to do this is to construct an adjacency matrix. This is obtained by assigning each column in a matrix to a vertex and every row an edge. For each edge e a 1 is placed in the column corresponding to t(e) and -1 for o(e). This matrix represents the boundary map $C_1(X) \to C_0(X)$. We can then calculate the kernel for this boundary map using linear algebra techniques.

Example 2.40. Given the following graph X we can compute the adjacency matrix.

$$\partial_1 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ -1 & -1 & -1 & -1 \end{bmatrix}$$

Now we need only compute the kernel.

$$Nul(\partial_1) = b \begin{bmatrix} -1 \\ 1 \\ 0 \\ 0 \end{bmatrix} + c \begin{bmatrix} -1 \\ 0 \\ 1 \\ 0 \end{bmatrix} + d \begin{bmatrix} -1 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

From this we deduce that $H_1(X)$ is generated by the paths $a^{-1}b$, $a^{-1}c$ and $a^{-1}d$.

Now that we have $H_1(X)$ we need to look for possible generating subgroups. That is we want any subgroup $H \subset H_1(X)$ such that $\frac{H_1(X)}{H} \cong \mathbb{Z}^3$. There will in general be infinitely many such groups but we can introduce a constrait to enumerate over. If $x = \sum_{e \in E} a_e e$ is an element of the first-order chain group then let $|x| = \sum_{e \in E} |a_e|$. Let S be a $\mathbb{Z}$-basis of a generating subgroup. We then define the height of S, $h(S) = max_{s \in S}|s|$, and the height of a subgroup H,

$h(H) = min_{\text{S generates H}} h(S)$. We note that if we fix h, then there are finitely mainly S such that $h(S) \leq h$. Since H is determined by a choice of S it then follows that there are finitely many H such that $h(H) \leq h$. We can then use h as a constraint to enumerate over.

Example 2.41. In the case from above we observe that $H_1(X) \cong \mathbb{Z}^3$ so the only option for H is the trivial subgroup. We then have that the only crystal we can choose is the maximal abelian cover which is in fact diamond.

We could at this point attempt to construct an explicit representation for the topological crystal corresponding to H. In fact we will show that Sunada's method enables us to skip this step.

We have now a method for enumerating topological crystals and so the final step necessary is to construct an embedding. Before we defined an embedding for a graph, but for a topological crystal we also want to require the embedding to realise all lattice symmetries as isometries.

Definition 2.42. An embedding for a topological crystal T = (V,E) is a graph embedding ϕ that sends all edges to straight line segments and which induces a homomorphism $\rho : G(X) \to \mathbb{R}^3$, such that if $v \in V$, $\sigma \in G(X)$ then $\phi(\sigma v) = \phi(v) + \rho(\sigma)$.

In the case of Sunada's embedding we seek to construct an embedding that is 'maximally symmetric'. To make sense of this definition we first define the symmetry group of a topological crystal.

Definition 2.43. The symmetry group of a topological crystal is given by

$$Aut(\tilde{X}/Q) = \{f \in Aut(\tilde{X}|f \circ p = p \circ f', \text{for some } f' \in Aut(Q)\}.$$

Intuitively we can think of this as the set of graph symmetries that perserves the quotient graph. The definition of a maximally symmetrical embedding then follows thusly.

Definition 2.44. A maximally symmetric embedding of a topological crystal is an embedding, ϕ which induces a homomorphism $\rho : Aut(T/Q) \to \mathbb{R}^3$ such that if $v \in V$, $\sigma \in Aut(T/Q)$ then $\phi(\sigma v) = \phi(v) + \rho(\sigma)$.

Sunada's method allows us to construct a maximally symmetric embedding, with a few pathological cases where the method fails. Before introducing Sunada's method we first wish to show that if we want to prove that an embedding is maximally symmetric, it is sufficient to prove the following conditions.

$$\sum_{e \in N_v} \phi(e) = 0, \text{ for every } x \in V \tag{2.5}$$

$$\sum_{e \in E_0} \langle x, v_0(e) \rangle v_0(e) = cx, \tag{2.6}$$

where E_0 is a fundamental domain for the edges and c is a postive constant.

If an embedding satisfies the first condition we call it a *harmonic realisation.* If an embedding satisfies both conditions we call it a *standard realisation.* We will first begin by proving that two harmonic embeddings are related via an affine transformation. We will then use this to prove that two standard realisations are related via a similitude. Finally we will show that if $v \in V$, $\sigma \in Aut(T/Q)$, and *phi* at standard realisation, then $\phi(\sigma T)$ is also standard realisation which proves that ϕ is a maximally symmetric.

First we prove that harmonic realisations are unique up to affine transformation. For this we use the following Lemma.

Theorem 2.45. *Two harmonic realisations that share the shame induced homomorphism $\phi : G(X) \to \mathbb{R}^3$ are related via a translation.*

Proof To prove this we begin by showing that a harmonic realisation minimises the expression

$$\sum_{e\in E_0} ||\phi(e)||^2. \tag{2.7}$$

Let ϕ be a harmonic realisation and ϕ' another embedding of the same topological crystal sharing the same induced homomorphism and let $f = \phi' - \phi$.

$$\begin{aligned}
\sum_{e\in E_0'} ||\phi_0'(e)|| &= \sum_{e\in E_0} (||\phi_0(e)||^2 + 2\langle \phi(e), f(t(e)) - f(o(e))\rangle + ||f(t(e)) - f(o(e))||^2) \\
&= \sum_{e\in E_0} (||\phi(e)||^2 + 2\langle \phi(e), f(t(e))\rangle - 2\langle \phi(e), f(o(e))\rangle + ||f(t(e)) - f(o(e))||^2) \\
&= \sum_{e\in E_0} (||\phi(e)||^2 + 2\langle \phi(-e), f(t(-e))\rangle - 2\langle \phi(e), f(o(e))\rangle + ||f(t(e)) - f(o(e))||^2) \\
&= \sum_{e\in E_0} (||\phi(e)||^2 - 4\langle \phi(e), f(o(e))\rangle + ||f(t(e)) - f(o(e))||^2) \\
&= \sum_{e\in E_0} ||\phi(e)||^2 + \sum_{v\in V_0} \sum_{e\in N_v} \langle \phi(e), f(v)\rangle + \sum_{e\in E_0} ||f(t(e)) - f(o(e))||^2 \\
&= \sum_{e\in E_0} ||\phi(e)||^2 + \sum_{e\in E_0} ||f(t(e)) - f(o(e))||^2 \\
&\geq \sum_{e\in E_0} ||\phi(e)||^2
\end{aligned} \tag{2.8}$$

Hence we have that a harmonic realisation minimises the above expression. Consider that if the other embedding is also a harmonic realisation then we must require

$$\begin{gathered}
\sum_{e\in E_0} ||f(t(e)) - f(o(e))||^2 = 0 \\
f(t(e)) = f(o(e))
\end{gathered} \tag{2.9}$$

Since this is true for all e, this tells us that f is constant and hence is a translation.

Theorem 2.46. *The harmonic realisations are unique up to affine transformations.*

Consider two harmonic realisations ϕ and ϕ' with induced homomorphisms, ρ and ρ'. We can find a matrix A such that $A\rho = \rho'$. Consider then the embedding $A\phi$. Note that $\sum_{e\in N_v} A\phi(e) = A\sum_{e\in N_v}\phi(e) = 0$. $A\phi$ is then a harmonic realisation with the induced homomorphism ρ'. Using the previous result then we have that $\phi' = A\phi + b$.

We now prove that two standard realisations are related by a similitude.

Theorem 2.47. *The standard realisations of a topological crystal are unique up to similar transformations.*

Proof We will prove this by first showing that a standard realisation minimises the term $vol(D)^{-\frac{2}{3}}\sum_{e\in E_0}||v_0(e)||^2$, where D is the parallelopiped formed by the generating set of $\rho(G(\tilde{X})$. Suppose ϕ is a standard realisation and ϕ' is a harmonic realisation of the same topological crystal. Since both are harmonic realisations we know that $\phi' = A\phi + b$. We make a choice of coordinates to obtain $\phi = (\phi_1, \phi_2, \phi_3)$,

$$\begin{aligned}\sum_{e\in E_0'}||\phi'(e)||^2 &= \sum_{e\in E_0}||A\phi(e)||^2\\ &= \sum_{e\in E_0}\sum_{i=1}^{3}\sum_{j=1}^{3}A_{ij}\phi_j\sum_{k=1}^{3}A_{ik}\phi_k\\ &= \sum_{i=1}^{3}\sum_{j=1}^{3}\sum_{k=1}^{3}cA_{ij}A_{ik}\delta_{jk}\\ &= ctrace(A^TA)\end{aligned} \tag{2.10}$$

We then consider the following inequality, the proof of which is given in Sunada's book,

$$tr(A^TA) \leq d(det(A^TA)) \tag{2.11}$$

And in particular the equality is realised if and only if $A = \lambda I$. Applying this we obtain

$$\begin{aligned}\sum_{e\in E_0'}||\phi'(e)||^2 &\geq cddet(A^TA)^{\frac{1}{3}}\\ &= cddet(A)^{\frac{2}{3}}\end{aligned} \tag{2.12}$$

We now consider the second requirement for ϕ being a standard realisation can be written as

$$\sum_{e\in E_0}\phi_i(e)\phi_j(e) = c\delta_{ij} \tag{2.13}$$

This then gives us

$$\sum_{e\in E_0} \phi_i(e)^2 = 3c$$

$$\sum_{e\in E_0'} ||\phi'(e)||^2 \geq \sum_{e\in E_0} (||\phi(e)||^2)\frac{vol(D')^{\frac{2}{3}}}{vol(D)^{\frac{2}{3}}}$$

$$vol(D')^{-\frac{2}{3}} \sum_{e\in E_0'} ||\phi'(e)||^2 \geq vol(D)^{-\frac{2}{3}} \sum_{e\in E_0} ||\phi(e)||^2 \tag{2.14}$$

Hence we have that a standard realisation minimises this value. Suppose both embeddings are standard realisations. Thus we require that $A^T A = \lambda I$ which gives us that A must be a scalar times an orthogonal matrix. Finally this tells us that ϕ' and ϕ are related via a similitude.

And now the final step is to prove that a standard realisation is indeed maximally symmetric.

Theorem 2.48. *A standard realisation is maximally symmetric.*

Proof Suppose that ϕ is a standard realisation of a topological crystal with covering map $p : T \to Q$, with induced homomorphism ρ. Let $g \in Aut(T/Q)$. We want to show $g \circ \phi$ is a standard realisation of the same topological crystal. The first step is to prove that it is an embedding of the same topological crystal. Let $\sigma \in G(T)$. First we consider that

$$\begin{aligned}(g\circ\phi)(\sigma v) &= \phi(g\sigma g^{-1} g v)\\ &= \phi(gv) + \rho(g\sigma g^{-1})\\ &= (g\circ\phi)(v) + \rho(g\sigma g^{-1})\end{aligned} \tag{2.15}$$

We now wish to show that $gG(T)g^{-1} = G(T)$. Suppose $\sigma \in G(T)$.

$$\begin{aligned}(g\sigma g^{-1})\circ p &= g^{-1}\circ\sigma\circ g\circ p\\ &= g^{-1}\circ\sigma\circ p\circ g'\\ &= g^{-1}\circ p\circ g'\\ &= g^{-1}\circ g\circ p\\ &= p\end{aligned} \tag{2.16}$$

Hence $g\sigma g^{-1} \in G(T)$ and $gG(T)g^{-1} = G(T)$ as required. This finally gives us that $g \circ \phi$ is an embedding of the same topological crystal. Lastly we need only show that $g \circ \phi$ is a standard realisation. First we observe

$$\begin{aligned}\sum_{e\in N_v} \phi(g(e)) = \sum_{e\in N_{g^{-1}(v)}} \phi(e)\\ = 0\end{aligned} \tag{2.17}$$

and then we observe

$$\begin{aligned}\sum_{e\in E_0}\langle x,\phi(g(e))\rangle &= \sum_{e\in g^{-1}(E)}\langle x,\phi(g(e))\rangle \\ &= cx\end{aligned} \tag{2.18}$$

And so we see here that $g \circ \phi$ is a standard realisation of the same topological crystal as ϕ and hence $g \circ \phi$ must be equal ϕ composed with a similitude. By sending g to this similitude we obtain our induced homomorphism and hence ϕ is maximally symmetric.

2.4 Sunada's Algorithm

We will now describe Sunada's algorithm for constructing standard realisations of topological crystals. Given a topological crystal T with quotient graph X we can construct $C_1(X,\mathbb{R})$. This is a vector space and we define on it the following inner product.

$$\langle e,e'\rangle = \begin{cases} 1, & \text{if } e=e' \\ -1, & \text{if } e=r(e') \\ 0, & \text{otherwise}\end{cases} \tag{2.19}$$

In describing this algorithm we will abuse notation and image paths as directly sitting in $C_1(X)$ which we in turn imagine sitting in $C_1(X,\mathbb{R})$. Using this inner product we now let p_0 be the orthogonal projection onto the subspace perpendicular to the subspace generated by H, in $H_1(X,\mathbb{R})$.

We now make a choice of vertex, v in the topological crystal and let $\phi(v)=0$. We now choose a path x'_v originating at v for each vertex v'. We then assign $\phi(v')=p_0(p^*(x'_v))$. We then let $\phi(e)$ be sent to the straight line segment joining o(e) and t(e).

Theorem 2.49. *Sunada's method does not depend on the choice of path x'_v.*

Proof Let x and y both originate at v and terminate at v'. We then consider the loop xy^{-1}. Note that since this is a loop it is contained in $H_1(T)$. We then have that $p^*(xy^{-1})\in H$ which tells us that xy^{-1} is perpendicular to the image of the orthogonal projection and hence sent to 0.

$$\begin{aligned} p_0(xy^{-1}) &= 0 \\ p_0(x)-p_0(y) &= 0 \\ p_0(x) &= p_0(y)\end{aligned} \tag{2.20}$$

Hence we have proven that Sunada's method does not depend on the choice of path.

Theorem 2.50. *If Sunada's method produces an injective function then this function is a standard realisation.*

Proof First we show that Sunada's method produces an embedding of the topological crystal. Suppose $\sigma \in G(T)$. Let y be a path between *sigmav* and v.

$$\begin{aligned} \phi(\sigma v') &= p_0 p^*(x_{\sigma v'}) \\ &= p_0 p^*(y + x_{v'}) \\ &= p_0(p^* y) + p_0(p^* x_{v'}) \end{aligned} \tag{2.21}$$

Note that $p^*(y)$ is a loop based at p(v). Hence either $p_0 p^*(x'_v) = 0$ or $p^*(x'_v)$ is in the image of the orthogonal projection in which case since it gets send to a the image of ϕ the induced homomorphism. Hence

$$\phi(\sigma v') = \phi(v') + b \tag{2.22}$$

where b is in the image of ϕ. Now we show that Sunada's method is harmonic. Let $c \in H_1(X, \mathbb{Z})$. First consider the case that c does not pass through the vertex. Then

$$\begin{aligned} \langle e, c\rangle &= 0, \text{for all } e \in E_0 \\ \langle \sum_{e \in E_0} e, c\rangle &= 0 \end{aligned} \tag{2.23}$$

Now consider the case that c does pass through the vertex. Then it for every edge component that terminates on the vertex we must have one that originates on the vertex. Hence

$$\langle \sum_{e \in E_0} e, c\rangle = 0 \tag{2.24}$$

Hence we have that

$$\langle \sum_{e \in E_0} e, c\rangle = 0, \text{for all } c \in H_1(X, \mathbb{Z}). \tag{2.25}$$

Since $H_1(X, \mathbb{Z})$ spans $H_1(X, \mathbb{R})$ we then have that $\sum_{e \in E_0} e$ is orthogonal to $H_1(X, \mathbb{R})$ finally giving us that

$$p_0(\sum_{e \in E_0} e) = 0 \tag{2.26}$$

Finally we show that Sunada's method is standard.

Example 2.51. Consider the nice example of diamond. The quotient graph is shown below.

We can choose a basis for $C_1(X, \mathbb{R})$, $\{a, b, c, d\}$. We now wish to choose a basis for $H_1(X, \mathbb{R})$ given by $\{\frac{a-b}{\sqrt{2}}, \frac{b-c}{\sqrt{2}}, \frac{a+b-c-d}{2}\}$. Note that this is an orthonormal basis. We then set atom 1 to the origin. We can then express every other atom via a word in $\{a, b, c, d\}$. We then need simply to project each element in $\{a, b, c, d\}$ into $H_1(X, \mathbb{R})$. We then get the following $a \to \begin{pmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ \frac{1}{2} \end{pmatrix}, b \to \begin{pmatrix} \frac{-1}{\sqrt{2}} \\ 0 \\ \frac{1}{2} \end{pmatrix}, c \to \begin{pmatrix} 0 \\ \frac{1}{\sqrt{2}} \\ \frac{-1}{2} \end{pmatrix}, d \to \begin{pmatrix} 0 \\ \frac{-1}{\sqrt{2}} \\ \frac{-1}{2} \end{pmatrix}$.

So now we have that all atoms in diamond are realised as integer combinations of these vectors. This is a particularly nice example as diamond in fact does exhibit this maximally symmetric embedding in nature.

Note that Sunada's method is harmonic and hence minimises the term $\sum_{e\in E_0} |\phi(e)|^2$, which essentially minimises the lengths of edges given a fixed choice of lattice translations. This minimisation can collaspe vertices with degree 1, and also collide vertices that share all their neighbours.

Chapter 3

Crystallographic Surfaces

In this chapter we set up the theory of Crystallographic Surfaces closely following Sunada's theory of Topological Crystals. This serves as both a two-dimensional extension of Sunada's theory as well as a model for computing Triply-Periodic Minimal Surfaces for our inverse EPINET algorithm. The algorithmic approach and layout is based off Sunada's book Topological Crystallography [8]. Some of the theorems and defintions are based off B. Mohar and C. Thomassen's book Graphs on Surfaces [2]. Further definitions are based off A. W. M. Dress's paper [1] and O. D. Friedrichs paper [4].

3.1 The Adventures of Agatha, Barry and Clara, Episode III

Barry was quite struck by Clara's story and he wrote a letter explaining back to her his paddock as a graph that sat inside a pairs of numbers which he understood to be a coordinate system. He didn't really feel that a coordinate system was very useful however in designing a paddock structure in the first place and he stressed this in his letter.

It was many days and nights before she replied but finally she did and the answer to his problem was pleasantly simple. It said to imagine himself at the centre of each paddock performing a Barry-centric subdivision. This would break each paddock up into triangles, but as the letter pointed out each triangle could be assigned an edge of exactly one of three colours based on a simple rule.

This in terms allowed Barry to simplify his picture so that each triangle was the vertex of a graph which had three edges each one a different colour leading to each of its three neighbours. It turned out that Barry could rotate around either a corner in the fence, a length of fence or a paddock simply by choosing two colours and alternating them repeatedly. If Barry could imagine a graph that satisfied these simply conditions then it could be built up into a paddock structure. Hence he now had a simple method of creating his paddock structure.

It turned out his sister had been given the same advice and sent him back some plans she had been drawing up for herself. Her plans were confusing to Barry however, it seemed that certain loops in her graph didn't correspond to the alternating colour paths...

3.2 Crystallographic Surfaces

We start by defining the notion of a 2-cell complex as a purely combinatorial object that we will use as the basis for our conception of a surface.

Definition 3.1. A 2-cell complex is a triplet (V, E, F) such that (V, E) is a graph along with functions $\partial_f : \mathbb{Z}/n_f\mathbb{Z} \to E$, and $\rho : F \to F$ such that

$$\begin{gathered} \partial_f(i+1) \subset N(t(\partial_f(i))) \\ Im(\partial_f) = Im(\partial_{\rho f}) \\ \text{If } \partial_f(i) = \partial_{\rho f}(j) \text{ then } \partial_f(i+1) = \partial_{\rho f}(j-1) \end{gathered} \tag{3.1}$$

We refer to $S^1 =$ (V, E) as the graph of the 2-cell complex. Elements of F are called orientated faces. $Im\delta_f$ is referred to as the boundary of the orientated face f. A non-orientated face is an equivalence class under the equivalence relation $f \sim \rho f$.

Example 3.2. As a simple example consider the 2-boquet graph with two faces with boundaries a and b.

Definition 3.3. The neighbourhood of an orientated edge N_e is the set of orientated faces f such that $e \in Im\delta_f$.

We now introduce an analogous concept to covering graph theory for 2-cell complexes.

Definition 3.4. A 2-cell complex is a map $p : \tilde{S} \to S$ which is a covering graph when restricted to the graphs $\tilde{S}^1$ and S^1 and

$$\begin{aligned} \rho p(f) &= p(\rho f) \\ Im\delta_{p(f)} &= p(Im\delta_f) \end{aligned} \tag{3.2}$$

p is a bijection on N_e.

The deck transformation group $G(\tilde{S})$ is the same as $G(\tilde{S}^1)$ for the covering graph.

Definition 3.5. The fundamental group with basepoint $v_0 \in V$ for a 2-cell complex (V, E, F) is given by

$$\pi_1(S, x_0) = \frac{\pi_1(S^1, v_0)}{H}, \tag{3.3}$$

where H is the normal subgroup generated by all δ_f conjugated by paths in (V,E) joining δ_f to v_0.

Theorem 3.6. *For a 2-cell complex $S = (V,E,F)$ and basepoint $v_0 \in V$, there is a bijective correspondence between subgroups of $\pi_1(S, v_0)$ and covering 2-cell complexes of S. Furthermore for each covering 2-cell complex $p : \tilde{S} \to S$, with basepoint $\tilde{v}_0 \in p^{-1}(v_0)$, we have*

$$G(X) \cong \frac{\pi_1(S, v_0)}{p_*(\pi_1(\tilde{S}, \tilde{v}_0))} \tag{3.4}$$

Proof We already have the Galois correspondence for the covering graph $p : \tilde{S}^1 \to S^1$. Noting that by definition $G(\tilde{S}) = G(\tilde{S}^1)$. Hence letting H be the normal subgroup generated by boundary cycles conjugated by appropriate paths, and similarly for $\tilde{H}$ we obtain

$$\begin{aligned} G(X) &\cong \frac{\pi(S^1, v_0)}{p^*\pi(\tilde{S}^1, \tilde{v}_0)} \\ &\cong \frac{\pi(S^1, v_0)/H}{p^*\pi(\tilde{S}^1, \tilde{v}_0)/H} \\ &\cong \frac{\pi(S^1, v_0)/H}{p^*(\pi(\tilde{S}^1, \tilde{v}_0)/\tilde{H})} \\ &\cong \frac{\pi(S, v_0)}{p^*\pi(\tilde{S}, \tilde{v}_0)} \end{aligned} \tag{3.5}$$

We now set up homology for 2-cell complexes.

The zeroth- and first-order chain groups of a 2-cell complex are simply be the zeroth- and first-order chain groups of its graph. Similarly the first-order boundary map of a 2-cell complex is the first-order boundary map of its graph.

Definition 3.7. The second-order chain group of a 2-cell complex, $C_2(X)$ is the free abelian group over orientated faces with the identification $\rho f = -f$.

Definition 3.8. The second-order boundary map of a 2-cell complex is given by

$\partial_2 = \sum_{e \in Im(\partial_f)} e$.

Definition 3.9. The first-order homology group of a 2-cell complex S is given by

$$H_1(S) = ker\partial_1 / Im\partial_2. \tag{3.6}$$

We then still obtain the Galois correspondence for abelian covering 2-cell complexes.

Theorem 3.10. *There exists a bijective correspondence between subgroups of $H_1(S)$ and abelian covering 2-cell complexes of S, furthermore for an abelian covering 2-cell complex $p : \tilde{S} \to S$ we have the following*

$$G(\tilde{S}) \cong \frac{H_1(S)}{p^* H_1(\tilde{S})} \tag{3.7}$$

Proof This proof follows from the case for graphs analogous to the case for the fundamental group.

We now introduce a helpful concept called the characteristic.

Definition 3.11. For a finite 2-cell complex the characteristic χ is given by $\chi = rankC_0 - rankC_1 + rankC_2$.

We now introduce our equivalent of a crystal for 2-cell complexes.

Definition 3.12. A crystallographic 2-cell complex is an abelian covering 2-cell complex over a finite surface $p : \tilde{S} \to S$ such that $G(\tilde{S})$ is a free abelian group of rank 3.

We now note that for our purposes we want to focus on triply-periodic surfaces and so we now define an orientable surface as a special kind of 2-cell complex.

Definition 3.13. A 2-cell complex is an orientable surface when

$|N_e| = 2$, for all edges e.

For every vertex v, there exists a permutation σ_v on N_v such for each $e \in N_v$, there is a face neighbouring both e and $r(\sigma_v e)$. Furthermore an orientated surface is non-regular if there exists an edge e and an orientated face f such that $N_e = \{f, \rho f\}$.

Example 3.14. Consider again the 2-boquet graph with three faces with boundaries a, b and ab.

For a non-regular example consider the below graph with three faces attached to a, c and abc.

Note that since an edge has exactly two neighbouring orientated faces it follows that both faces can be recovered from the rotation scheme. We can use this to produce a method of enumerating orientated surfaces over a finite graph.

Definition 3.15. A rotation scheme on a finite graph is a collection of permutations σ_v on each N_v.

Theorem 3.16. *A rotation scheme induces an orientated surface on a graph.*

Proof Select a vertex v, and then select an orientated edge e in its neighbourhood. Let this be the first edge in our path. We can then select the vertex t(e) and then the orientated edge $\sigma_{t(e)} r(e)$. We then add this new edge to our path and keep going. This is a finite graph so we must eventually reach our original edge. At this point we have a cycle which we identify as the boundary of a new face f. We can then repeat this procedure starting from a different vertex, edge pair, remembering not to repeat the same cycle twice until we have all faces.

Definition 3.17. For a finite orientable surface the genus is given by $g = 1 - \frac{\chi}{2}$

Theorem 3.18. *For a finite orientable surface $rankH_1(S) = 2g$*

Proof

$$\begin{aligned} rankH_1(S) &= rankker\delta_1 - rankIm\delta_2 \\ &= rankC_1(S) - Im\delta_1 - rankC_2(S) + 1 \\ &= rankC_1(S) - rankC_0 - rankC_2(S) + 2 \\ &= 2 - \chi \\ &= 2g \end{aligned} \tag{3.8}$$

Here will denote a finite surface of genus g by M_g.

3.3 Enumerating Crystallographic Surfaces

We now wish to construct a theory for enumerating crystallographic surfaces analogous to Sunada's method.

The first step is to enumerate over finite graphs X. For each graph X we enumerate through all choices of rotation schemes to produce a finite surface M_g.

Once we have a choice of surface M_g we can enumerate through all crystallographic surfaces over this surface by calculating the homology group $H_1(M_g)$ and enumerating through all subgroups to produce a covering surface.

This provides a complete process for enumerating crystallographic surfaces. It now remains to develop a theory for embedding these surfaces.

Definition 3.19. An embedding of a crystallographic surface in $\mathbb{R}^3$ is a collection of injective functions $f_V : V \to \mathbb{R}^3$, $f_E \to P(\mathbb{R}^3)$ and $f_F : F \to P(\mathbb{R}^3)$ such that, $f_e(e)$ is a line segment joining $f_v(o(e))$ and $f_v(t(e))$, and $f_f(f)$ is a disc bounded by $f_e(\delta(f))$.

Note that this definition has a lot of degrees of freedom associated with how we embed the vertices, edges and faces. We wish to have a more combinatorial process and so we introduce the concept of flags.

Definition 3.20. A flag on a CW-complex is a triplet (v, e, f) such that

$$\begin{aligned} \text{either } v = o(e) \text{ or } v = t(e) \\ e \in Im\delta_f \end{aligned} \tag{3.9}$$

We now introduce a more combinatorial notion of an embedding.

Definition 3.21. An barycentric embedding of a crystallographic surface is collection of injective functions, $f'_V : V \to \mathbb{R}^3$, $f'_E : E \to \mathbb{R}^3$ $f'_f : F \to \mathbb{R}^3$ such that for any two flags c and c', the triangles formed by f'(c) and f'(c') only intersect on at most a shared boundary edge.

We will now show that this concept indeed deserves the term embedding.

Theorem 3.22. *A barycentric embedding of a crystallographic surface induces a graph embedding of the same surface.*

Proof The induced function f_V is simply f'_V. f_E maps e to the union of the straight line segment joining $f'_V(o(e))$ and $f'_E(e)$ with the straight line segment joining $f'_E(e)$ and $f'_V(t(e))$. f_F maps f to the union of the triangles formed by f(c), where c is a flag.

Since this embedding is described in terms of 'flags' we wish to derive a combinatorial structure over these flags to make this process more combinatorial. Our first step is to define what we will refer to as reflection.

Theorem 3.23. *If we fix the edge of a flag and either the vertex or face, then there is only one choice for the remaining element.*

Proof We will consider both possible cases. First consider the case that the vertex and edge are fixed. We then have two choices for the face from the neighbourhood of the edge. Suppose instead we fix the edge and face. We then have two choices for the vertex as either the origin or terminus of the edge.

We will now define reflections on our flags in terms of an action of the free Coexter group with three generators.

Definition 3.24. The reflection group action on a surface is the action of $\langle \sigma_i \rangle$ on the set of flags on a surface where

$\sigma_0 c$ switches the vertex,

$\sigma_1 c$ switches to the adjacent edge with the same face in the rotation scheme of the vertex.

$\sigma_2 c$ switches the face.

We now show that these actions deserve the term reflection.

Theorem 3.25. *For all i, $\sigma_i^2 = 1$.*

Proof We claim that the case follows from the previous theorem for σ_0 and σ_2. For σ_1 we consider that this action switches to the adjacent edge with the same face in the rotation scheme of the vertex. Note that there is a unique choice of such edge since by the definition of a surface each oriented edge belongs to two oriented faces, one shared with $r(\sigma e)$ and one shared with $\sigma^{-1} r(e)$.

Definition 3.26. For the set of flags on a surface the function $m_{ij} : C \to \mathbb{N}$, $i, j \in \{0, 1, 2\}$ is given by

$m_{ij}(C) = min\{m | (\sigma_i \sigma_j)^m c = c\}$

Theorem 3.27. $m_{ij}(C) = 2$ *when* $|i - j| > 1$.

This object can now be used for

Definition 3.28. A standard barycentric embedding is an barycentric embedding which is standard when interpreted as a graph embedding as defined by Sunada.

Note that there is a representation of this embedding using the Delaney-Dress symbol.

Note however that a non-regular surface does not have a barycentric embedding.

Example 3.29.

Note that we can form a representation of $C_1(X', \mathbb{R})$ using $C \times \sigma$. However there is no convenient way of determining $H_1(X, \mathbb{R})$.

Definition 3.30. For a surface, the Delaney-Dress graph, X^D is formed by assigning a vertex to each flag and an edge to each reflection between flags. We note that the each edge has a number $\{0, 1, 2\}$ which we refer to as the colour of the edge.

Theorem 3.31. *If $p : \tilde{S} \to S$ is a covering surface, then there is a corresponding covering graph $p : \tilde{X}^D \to X^D$.*

Proof This follows from the observation that a flag is a ordered triplet of vertex, edge and face all of which are sent via covering map.

Definition 3.32. An orbit is a simple loop through the Delaney-Dress graph made up of an alternating pair of colours.

Theorem 3.33. *There is a one-to-one correspondence between orbits and the elements of a surface.*

Proof Note that a pair of colours corresponds to switching between a pair of elements for a flag. The remaining element therefore is preserved. This defines the forward correspondence. Work out details later.

Definition 3.34. The lattice group of the Delaney-Dress graph of a covering surface is the deck transformation group of the induced covering graph.

Definition 3.35. A Delaney-Dress embedding is a function $f : C \to \mathbb{R}^3$ such that the lattice group is send to a group of lattice isometries.

Theorem 3.36. *An embedding of a Delaney-Dress graph induces a barycentric embedding on a crystallographic surface.*

Proof First we consider a regular surface. In this case we simply embed each surface element at the centre of its orbit.

Definition 3.37. The standard Delaney-Dress embedding of a regular crystallographic surface is the embedding induced by the standard realisation of it's Delaney-Dress symbol.

This is the definition we shall use. Note that by minimising the gaps between centres of faces we are essentially utilising a discrete variant of minimal surfaces.

Chapter 4

Inverting EPINET

In this chapter we seek to apply the theory of crystallographic surfaces to the inverse problem for EPINET.

4.1 The Adventures of Agatha, Barry and Clara, Episode IV

Barry thought maybe he was simply confused. So he tried plotting out the graph in two-dimensional coordinates only to find that the structure simply didn't work. Not anyway he tried it.

Barry wrote to Clara concerned for his sister's sanity. Clara wrote back explaining that Agatha's hyperbolic surface was different to his and that he could understand this if only he thought back to how Clara had been confused by the unexpected loops in her space.

Barry pondered this. The alternating colour loops he expected. The others were quite strange but he realised that she was right he could in fact find six directions sorted into three pairs, that could be alternated to generate all of these loops.

He realised then he could describe her space with three coordinates instead of the standard two using these three paths and placing all other points conveniently around these three array. The resulting structure made perfect sense and it seemed as if Clara was right and that their whole omniverse was in fact three dimensional. This was too much for Barry and so he decided to go out for a drink at the pub.

4.2 Surfaces from Crystals

With our theory set up we can now present the inverse problem from EPINET as follows. For a crystal T, can we find an embedding for a crystallographic surface S with T as it's graph.

$$\begin{tikzcd}T \arrow[r, hook] & S \arrow[r, hook] & \mathbb{R}^3\end{tikzcd}$$

So we seek to amend our enumeration algorithm for crystallographic surfaces to those with T as their graph.

The first step in our algorithm was to enumerate through finite graphs. In this case however we desire that these finite graphs can be lifted to T. We note then from covering graph theory that all such graphs cover Q, the quotient graph of T.

$$\begin{tikzcd}T \arrow[r] & X \arrow[r] & Q\end{tikzcd}$$

To do this we choose normal subgroups of G(T), generating subgroups H, and quotient T by these groups. Once such a graph X is obtained the next stage is to enumerate over rotation schemes of X. Note however that before we chose the rotation schemes and this placed a restriction on the graphs we could lift to. In this case we already have a chosen graph to lift to so clearly we will need to restrict our choice of rotation scheme.

In particular a 2-cell complex lifts to a covering graph of its graph if and only if $C_2(X) \subset H$. Restricting to this condition we can then enumerate through until we find a finite surface that lifts to a crystallographic surface with T as its graph.

$$\begin{tikzcd}T \arrow[r] \arrow[d, hook] & X \arrow[r] \arrow[d, hook] & Q \\ S \arrow[r] & M_g & \end{tikzcd}$$

Our next step is to find an embedding for this surface.

$$\begin{tikzcd}S \arrow[r, hook] & \mathbb{R}^3\end{tikzcd}$$

Here we will use the standard Delaney-Dress embedding. This notion of embedding can handle non-regular surfaces, is easily computable, and most importantly the final embedding has a representation in terms of the Delaney-Dress symbol which can be used to determine collisions.

We can now set up the precise details relevant for calculations.

Definition 4.1. Let the dual graph of 2-cell complex be given by $X^D = (V, E)$ where V = C the set of chambers and $E = C \times \sigma$.

Definition 4.2. Let $C_1(X^D, \mathbb{R})$ be the free abelian group with real coefficients generated by $C \times \{\sigma_0, \sigma_1, \sigma_2\}$ with the identification $(\sigma_i C, \sigma_i) = (C, \sigma_i)^{-1}$.

Following Sunada's method we then need to define his inner product on $C_1(X^D, \mathbb{R})$. Our goal is to construct an orthogonal projection from $C_1(X^D, \mathbb{R})$ to the orthogonal subspace of $H_1(X^D, \mathbb{R})$ to H^D. Here we will break this into two steps, the first projection onto $H_1(X^D, \mathbb{R})$ and then the second orthogonal to H^D.

We begin first by describing how to project onto $H_1(X^D, \mathbb{R})$. We want to choose an orthonormal basis for our calculations and for this we make a orientation for the edges in $C_1(X^D, \mathbb{R})$ as well as an ordering for the purpose of matrix computations. Note that for any homomorphism $C_1(X^D) \to C_0(X^D)$ there is a natural linear map $C_1(X^D, \mathbb{R}) \to C_0(X^D, \mathbb{R})$. We then claim that it is simple to construct a matrix representation of the induced boundary map $\delta_1 : C_1(X^D, \mathbb{R}) \to C_0(X^D, \mathbb{R})$ using our orthonormal basis. This is given by the fact that $\delta_1(C \times \sigma_i) = C$.

Once we have computed a representation of δ_1 we then compute a basis for the kernal of this map to produce basis for $H_1(X^D, \mathbb{R})$. Once we have this basis it is a simple matter to apply Gram-Schimdt to construct an orthonormal basis for $H_1(X^D, \mathbb{R})$, e_{ij}. We can then representation the orthogonal projection map $C_1(X^D, \mathbb{R}) \to H_1(X^D, \mathbb{R})$ by $\sum_{j=1}^{b_1(X^D)} e_{ij}e_{jk}$.

We now describe a procedure to project orthogonal to H^D. For this we will define a natural way to extend the quotient map $q : C_1(X) \to C_1(Q)$ to a linear map $q^D : C_1(X^D, \mathbb{R}) \to C_1(Q, \mathbb{R})$.

First we define a homomorphism $\psi : C_1(X^D) \to C_1(X)$.

Definition 4.3. Let $\psi : C_1(X^D) \to C_1(X)$ be given by

$$\psi(c \times \sigma_i) = \begin{cases} e & i = 0 \\ 0 & \text{otherwise} \end{cases} \tag{4.1}$$

where e is the shared edge between c and $\sigma_i c$.

This homomorphism was motivated by imagining the Delaney-Dress graph as made up of vertex orbits with σ_0 edges connecting them. We note them that we can obtain the original graph X simply by collapsing the vertex orbits to a point and then the pairs of σ_0 edges to a single edge joining the vertices. We then obtain the following result.

Theorem 4.4. *The following diagram is commutative.*

$$\begin{tikzcd}
C_1(T^D) \arrow[r] \arrow[d] & C_1(X^D) \arrow[d] \\
C_1(T) \arrow[r] & C_1(D)
\end{tikzcd}$$

Proof We claim that this result follows from the fact that covering maps preserve the colours of edges.

Theorem 4.5. *The image of H^D under ψ is equal to H.*

Proof This proof follows easily from the above diagram and the fact that H^D and H are the images of the homology groups under their respective covering maps.

We now define the following homomorphism.

Definition 4.6. The homomorphism $f : C_1(X^D) \to C_1(Q)$ is given by $f = q \circ \psi$.

Finally we can state the following crucial result.

Theorem 4.7. *If T is the maximal abelian cover of Q then the kernal of q^D restricted to $H_1(X^D)$ is precisely H^D.*

Proof This follows from the previous results easily by noting that since T is the maximal abelian cover of Q, $H_1(T)$ is mapped to 0 in $H_1(Q)$.

We now consider the linear map induced by the homomorphism $f : H_1(X^D) \to H_1(Q)$. The kernal of this map is the subspace generated by H^D. Note then by the universal quotient property the image of this map is isomorphic as a vector space to the orthogonal projection we seek to compute. We might worry about the fact that the image is not isomorphic as a inner product space however this is not hugely important here since the image is related to our projection via a affine transformation which is sufficient for our purposes.

For the final stage we first make a choice of vertex in X^D to be our origin. Delaney-Dress graphs typically come with a choice of vertex ordering which increases with a breath first search through the graph so it is convenient to choose the vertex 1 as the origin. We can then compute the set of orbits and for each orbit determine the minimally numbered vertex. We can then easily find a path from this vertex back to 1 simply by travelling backwards along vertexes of decreasing number order. We can then easily compute the projection of a path back to 1 in order to embed all cycles in single cell. We can then simply shift this cell using translations to produce the entire embedding.

4.3 A Worked Example

This algorithm was implemented for simple cubic and a surface resembling the p surface was produced. The code can be seen in the appendix and pictures seen in figures 4.1 and 4.2.

4.4 Intersections

There is still another problem to consider here, namely, how would we go about determining whether there are any intersections between triangles in our triangulated surface.

Each triangle can be represented by $\{\alpha x_1 + \beta x_2 + \gamma x_3 | \alpha + \beta + \gamma \leq 1\}$, where x_1, x_2, x_3 are the embeddings for the barycentre points of the vertex, edge and face of the flag corresponding to that triangle.

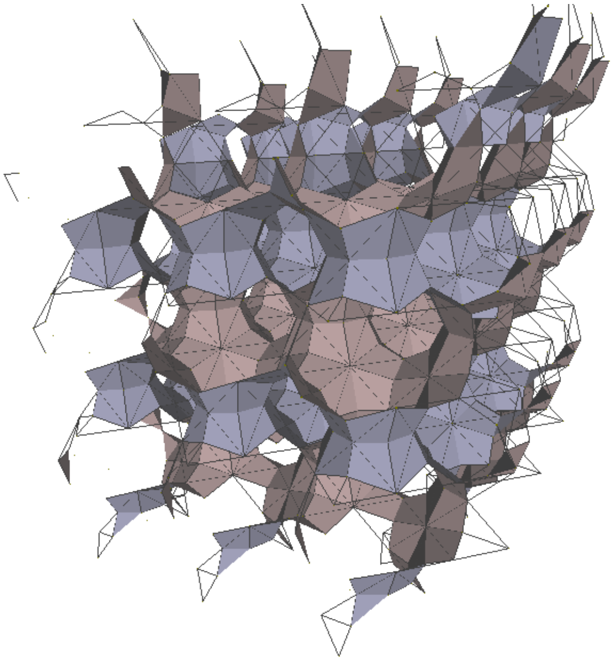


Figure 4.1: The final surface computed for simple cubic.

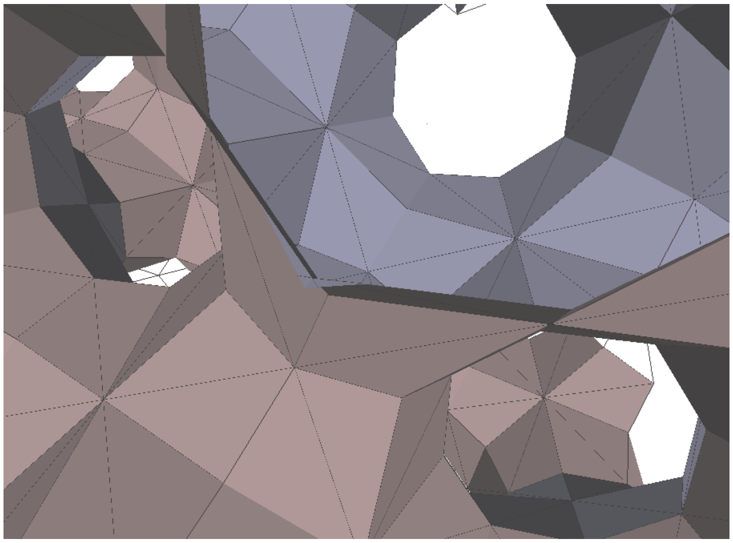


Figure 4.2: A closed up view of a ’collar’.

Our idea to make this easier is to represent our potential embedding as a homomorphism acting on the Coexter group. First we introduce the following notation.

Definition 4.8.

$$(c, (\sigma_{i_k})_{k\in\{0,\ldots,n\}}) = (c, \sigma_{i_1})(\sigma_{i_1} c, \sigma_{i_2})...(c, \sigma_{i_n}) \tag{4.2}$$

Consider then that in our algorithm we fix a starting vertex on the Delaney-Dress graph. Let our starting vertex be c. We then assign all vertices, c' on the Delaney-Dress graph a path $x_{c'}$. All of these paths start at c, and hence using the above notation can be represented as $(c, (\sigma_{i_k})_{k\in\{0,\ldots,n\}})$ or simply in the form $(\sigma_{i_k})_{k\in\{0,\ldots,n\}})$. We might then be tempted to identify this form directly with the Coexter group, but we immediately have to be careful due to the fact the that we created an abelian group over these (c, σ_i) elements.

To deal with this we note the following.

Theorem 4.9. *Every simple path has a unique representation in the form $(c, \sigma_{i_k\, k\in\{0,\ldots,n\}})$, where $((c, \sigma_{i_k}))_{k\in\{0,\ldots,n\}}$ is a path.*

Proof We first imagine a simple path as an element of the first-order chain group of the Delaney-Dress graph. This is in an abelian group so we can order the generators however we wish. This ordering is a path ordering if for $e_i = (c', \sigma_i)$ we have that $o(e_i) = t(e_{i-1})$. We then choose this ordering for the generators. Note then that because this path is simple if $o(e_i) = o(e_j)$, then i = j and hence there is only one way to form this ordering and this gives us a unique representation.

Since the algorithm is independent of choice of path we can then choose simple paths for all vertices. We can then represent the Delaney-Dress embedding as a homomorphism on the Delaney-Dress graph to a collection of vectors in $\mathbb{R}^3$.

```
A = {2, 25, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 44, 27, 29, 31,
  33, 35, 37, 39, 41, 43, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64,
  66, 68, 70, 72, 74, 76, 78, 80, 82, 84, 86, 88, 90, 92, 94, 96};
B = {3, 96, 6, 78, 76, 10, 35, 32, 14, 86, 83, 18, 66, 64, 22, 46,
  24, 26, 95, 30, 54, 51, 34, 38, 62, 59, 42, 90, 88, 45, 49, 74,
  72, 53, 57, 82, 79, 61, 65, 69, 94, 92, 73, 77, 81, 85, 89, 93};
F = {4, 5, 22, 8, 9, 12, 13, 16, 17, 20, 21, 25, 43, 28, 29, 32,
  33, 36, 37, 40, 41, 44, 47, 48, 51, 52, 55, 56, 59, 60, 63, 64,
  67, 68, 71, 72, 75, 76, 79, 80, 83, 84, 87, 88, 91, 92, 95, 96}
```

This stores the compressed form of the Delaney - Dress symbol.

```
G := {}; n := 1;
While[ n < 97, If[! MemberQ[A, n], AppendTo[G, n]]; n++]
```

```
H := {}; n := 1; While[n < 97, If[! MemberQ[B, n], AppendTo[H, n]];
 n++]
```

```
J := {}; n := 1; While[n < 97, If[! MemberQ[F, n], AppendTo[J, n]]; n++]
```

These generate the missing numbers from the compressed form in the appropriate order.

```
   K := {}; n := 1;
While[n < 97, If[MemberQ[A, n], AppendTo[K, G[[FirstPosition[A, n]]]],
  AppendTo[K, A[[FirstPosition[G, n]]]]]; n++]; m := 1; While[m < 97,
 If[MemberQ[B, m], AppendTo[K[[m]], First[H[[FirstPosition[B, m]]]]],
  AppendTo[K[[m]], First[B[[FirstPosition[H, m]]]]]]; m++];
k := 1; While[k < 97, If[MemberQ[F, k],
  AppendTo[K[[k]], First[J[[FirstPosition[F, k]]]]],
  AppendTo[K[[k]], First[F[[FirstPosition[J, k]]]]]]; k++]
```

This determine the uncompressed form of the Delaney-Dress symbol by comparing elements from the previous two lists to determine adjacencies.

```
L := {}; n := 1; While[n < 97, AppendTo[L, 0]; n++]
```

```
M := {};
m := 1;
P := {};
While[m < 4, AppendTo[P, {}];
 n := 1;
 While[n < 97, If[K[[n]][[m]] > n,
   AppendTo[M, ReplacePart[ReplacePart[L, n → -1 ], K[[n]][[m]] → 1]];
   AppendTo[P[[m]], {n, K[[n]][[m]]}]];
  n++];
 m++];
```

```
Q := Transpose[M]
```

This generates a the first-order boundary map of the Delaney-Dress graph as well as an ordering on the edges as a reference for the choice of basis in all future matrix calculations.

```
R := NullSpace[Q]
```

This calculates the homology group of the Delaney-Dress graph.

```
S := {};
n := 1;
m := 1; While[n < 13, While[m < 97, If[! MemberQ[S, m, 2], k := 1;
   AppendTo[S, {m}];
   While[k < 8, AppendTo[S[[n]], P[[If[Mod[k, 2] == 1, 1, 2]]][[FirstPosition[
          P[[If[Mod[k, 2] == 1, 1, 2]]], S[[n]][[k]]][[1]]]][[If[FirstPosition[
           P[[If[Mod[k, 2] == 1, 1, 2]]], S[[n]][[k]]][[2]] == 1, 2, 1]]]];
     k++] n++];
  m++]]
```

```
T := {};
n := 1;
m := 1; While[n < 9, While[m < 97, If[! MemberQ[T, m, 2], k := 1;
   AppendTo[T, {m}];
   While[k < 12, AppendTo[T[[n]], P[[If[Mod[k, 2] == 1, 2, 3]]][[FirstPosition[
          P[[If[Mod[k, 2] == 1, 2, 3]]], T[[n]][[k]]][[1]]]][[If[FirstPosition[
           P[[If[Mod[k, 2] == 1, 2, 3]]], T[[n]][[k]]][[2]] == 1, 2, 1]]]];
     k++] n++];
  m++]]
```

```
U := {};
n := 1;
m := 1; While[n < 25, While[m < 97, If[! MemberQ[U, m, 2], k := 1;
   AppendTo[U, {m}];
   While[k < 4, AppendTo[U[[n]], P[[If[Mod[k, 2] == 1, 3, 1]]][[FirstPosition[
          P[[If[Mod[k, 2] == 1, 3, 1]]], U[[n]][[k]]][[1]]]][[If[FirstPosition[
           P[[If[Mod[k, 2] == 1, 3, 1]]], U[[n]][[k]]][[2]] == 1, 2, 1]]]];
     k++] n++];
  m++]]
```

These generate lists of vertex, edge and face orbits, in other words the alternating colour loops. For future reference.

```
L1 := {}; n := 1; While[n < 145, AppendTo[L1, 0]; n++]
```

```
V := {}; n := 1;
While[n < 13, m := 1; AppendTo[V, {}]; While[m < 9, AppendTo[V[[n]],
   ReplacePart[L1, (FirstPosition[P[[If[Mod[m, 2] == 1, 1, 2]]], S[[n]][[m]]][[
        1]] + If[Mod[m, 2] == 1, 0, 48]) → If[FirstPosition[
         P[[If[Mod[m, 2] == 1, 1, 2]]], S[[n]][[m]]][[2]] == 1, 1, -1]]];
  m++];
 n++]
```

```
V1 := Total[V, {2}]
```

```
W := {}; n := 1; While[n < 9, m := 1;
 AppendTo[W, {}];
 While[m < 13, AppendTo[W[[n]], ReplacePart[L1, (FirstPosition[P[[If[Mod[m, 2] ==
           1, 2, 3]]], T[[n]][[m]]][[1]] + If[Mod[m, 2] == 1, 48, 96]) → If[
      FirstPosition[P[[If[Mod[m, 2] == 1, 2, 3]]], T[[n]][[m]]][[2]] == 1, 1, -1]]];
  m++];
 n++]
```

```
W1 := Total[W, {2}]
```

```
X := {};
n := 1;
While[n < 25, m := 1;
 AppendTo[X, {}];
 While[m < 5, AppendTo[X[[n]], ReplacePart[L1, (FirstPosition[P[[If[Mod[m, 2] == 1,
          3, 1]]], U[[n]][[m]]][[1]] + If[Mod[m, 2] == 1, 96, 0]) → If[
      FirstPosition[P[[If[Mod[m, 2] == 1, 3, 1]]], U[[n]][[m]]][[2]] == 1, 1, -1]]];
  m++];
 n++]
```

```
X1 := Total[X, {2}]
```

These generate vectors in the first-order homology group with real coefficients corresponding to the alternating colour loops. These are used but were ultimately found to be redundant.

```
Y = Join[V1, W1, X1, R]
```

This creates a list of vectors that span the first-order homology group, made up of, first the vectors computed in the last part, and then the basis vectors computed before from the boundary map.

```
Z = Orthogonalize[Y]
```

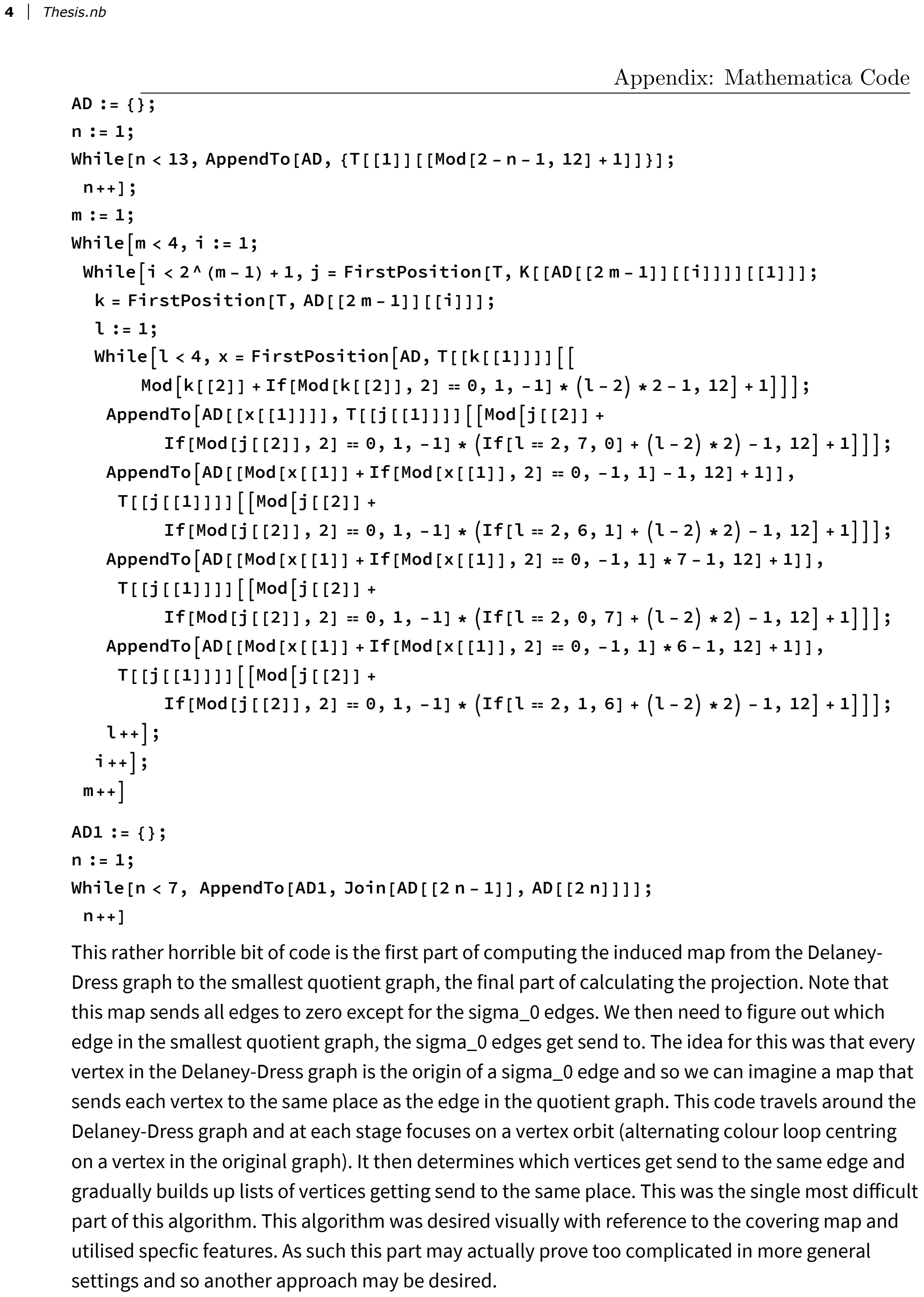
```
AA := {};
n := 1; While[n < 94, If[! (Z[[n]] == L1), AppendTo[AA, Z[[n]]]];
 n++]
```

This applies Gram-Schidmt to the previous list of vectors and then removes the zero vectors produced. The benefit is that with this basis we know the first ‘x’ number of vectors correspond to the subspace spanned by the alternating coloured loops. So if we then select all of the other vectors we will have a basis for a space perpendicular to this coloured subspace. This is useful since the coloured subspace is contained in H and hence allows us not instead of projecting onto the first-order homology group we can project onto a subspace that is closer to the final space we wish to project to. Unfortunately this turned out to not save any time whatsoever and was completely useless. But it didn’t hurt so oh well.

```
AB := {}; n := 1; While[n < 7, AppendTo[AB, AA[[50 - n]]]; n++]
```

```
AC = Transpose[AB].AB
```

These collect the vectors corresponding to the subspace we wish to project to and computes the projection map.



```
AD := {};
n := 1;
While[n < 13, AppendTo[AD, {T[[1]][[Mod[2 - n - 1, 12] + 1]]}];
 n++];
m := 1;
While[m < 4, i := 1;
 While[i < 2^(m - 1) + 1, j = FirstPosition[T, K[[AD[[2 m - 1]][[i]]]][[1]]];
  k = FirstPosition[T, AD[[2 m - 1]][[i]]];
  l := 1;
  While[l < 4, x = FirstPosition[AD, T[[k[[1]]]][[
      Mod[k[[2]] + If[Mod[k[[2]], 2] == 0, 1, -1] * (l - 2) * 2 - 1, 12] + 1]]];
   AppendTo[AD[[x[[1]]]], T[[j[[1]]]][[Mod[j[[2]] +
        If[Mod[j[[2]], 2] == 0, 1, -1] * (If[l == 2, 7, 0] + (l - 2) * 2) - 1, 12] + 1]]];
   AppendTo[AD[[Mod[x[[1]] + If[Mod[x[[1]], 2] == 0, -1, 1] - 1, 12] + 1]],
    T[[j[[1]]]][[Mod[j[[2]] +
        If[Mod[j[[2]], 2] == 0, 1, -1] * (If[l == 2, 6, 1] + (l - 2) * 2) - 1, 12] + 1]]];
   AppendTo[AD[[Mod[x[[1]] + If[Mod[x[[1]], 2] == 0, -1, 1] * 7 - 1, 12] + 1]],
    T[[j[[1]]]][[Mod[j[[2]] +
        If[Mod[j[[2]], 2] == 0, 1, -1] * (If[l == 2, 0, 7] + (l - 2) * 2) - 1, 12] + 1]]];
   AppendTo[AD[[Mod[x[[1]] + If[Mod[x[[1]], 2] == 0, -1, 1] * 6 - 1, 12] + 1]],
    T[[j[[1]]]][[Mod[j[[2]] +
        If[Mod[j[[2]], 2] == 0, 1, -1] * (If[l == 2, 1, 6] + (l - 2) * 2) - 1, 12] + 1]]];
   l++];
  i++];
 m++]
```

```
AD1 := {};
n := 1;
While[n < 7, AppendTo[AD1, Join[AD[[2 n - 1]], AD[[2 n]]]];
 n++]
```

This rather horrible bit of code is the first part of computing the induced map from the Delaney-Dress graph to the smallest quotient graph, the final part of calculating the projection. Note that this map sends all edges to zero except for the sigma_0 edges. We then need to figure out which edge in the smallest quotient graph, the sigma_0 edges get send to. The idea for this was that every vertex in the Delaney-Dress graph is the origin of a sigma_0 edge and so we can imagine a map that sends each vertex to the same place as the edge in the quotient graph. This code travels around the Delaney-Dress graph and at each stage focuses on a vertex orbit (alternating colour loop centring on a vertex in the original graph). It then determines which vertices get send to the same edge and gradually builds up lists of vertices getting send to the same place. This was the single most difficult part of this algorithm. This algorithm was desired visually with reference to the covering map and utilised specfic features. As such this part may actually prove too complicated in more general settings and so another approach may be desired.

```
AE := {};
n := 1;
While[n < 145, AppendTo[AE, {FirstPosition[AD1,
      P[[Quotient[n - 1, 48] + 1]][[Mod[n - 1, 48] + 1]][[1]]][[1]], FirstPosition[
      AD1, P[[Quotient[n - 1, 48] + 1]][[Mod[n - 1, 48] + 1]][[2]]][[1]]}];
 n++]
```

```
AE1 = Drop[AE, {49, 144}]
```

```
AF := {};
n := 1;
While[n < 49, If[! MemberQ[AF, AE1[[n]]] && ! MemberQ[AF, Reverse[AE1[[n]]]],
   AppendTo[AF, AE1[[n]]]];
 n++]
```

This code then determines a representation for the edges in terms in the lists determined previously.

```
AG := {}; n := 1; While[n < 4, AppendTo[AG, 0] n++]
```

```
AH := {};
n := 1;
While[n < 145, If[MemberQ[AF, AE[[n]]],
    AppendTo[AH, ReplacePart[AG, FirstPosition[AF, AE[[n]]][[1]] → 1]],
    If[MemberQ[AF, Reverse[AE[[n]]]], AppendTo[AH, ReplacePart[AG,
        FirstPosition[AF, Reverse[AE[[n]]]][[1]] → -1]], AppendTo[AH, AG]]];
```

```
AI := Transpose[AH]
```

This code uses the previous representation to compute the induced map.

```
AJ := AI.AC
```

Here we compose the two maps.

```
AK = Transpose[AJ]
```

```
n := 1;
AL := {};
While[n < Length[S] + 1, AppendTo[AL, {Min[S[[n]]]}];
 m := 1;
 While[AL[[n]][[m]] ≠ 1, If[K[[AL[[n]][[m]]]][[1]] < AL[[n]][[m]],
    AppendTo[AL[[n]], K[[AL[[n]][[m]]]][[1]]],
    If[K[[AL[[n]][[m]]]][[2]] < AL[[n]][[m]], AppendTo[AL[[n]],
      K[[AL[[n]][[m]]]][[2]]], AppendTo[AL[[n]], K[[AL[[n]][[m]]]][[3]]]]];
  m++];
 n++]
```

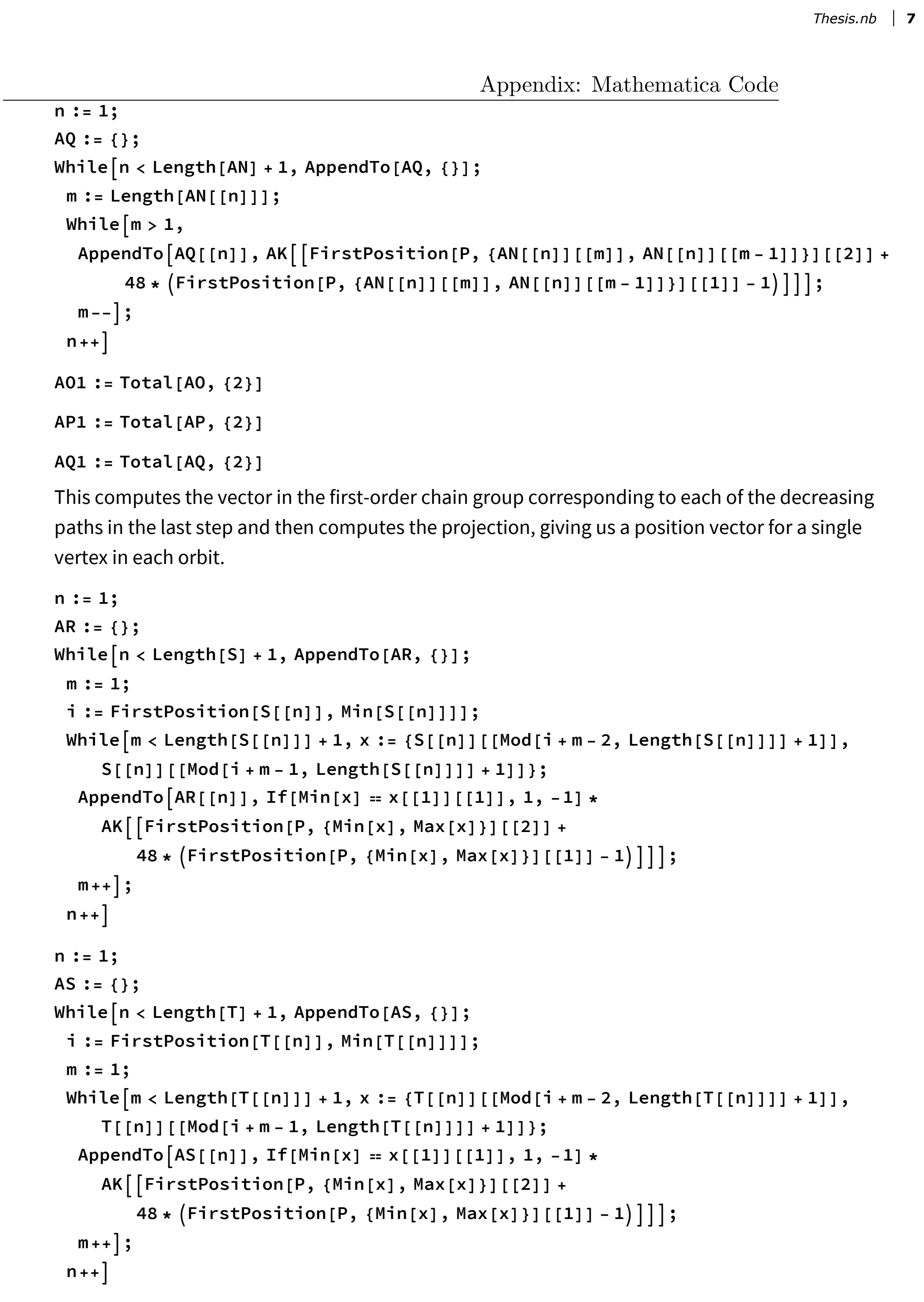
```
n := 1;
AM := {};
While[n < Length[T] + 1, AppendTo[AM, {Min[T[[n]]]}];
 m := 1;
 While[AM[[n]][[m]] ≠ 1, If[K[[AM[[n]][[m]]]][[1]] < AM[[n]][[m]],
   AppendTo[AM[[n]], K[[AM[[n]][[m]]]][[1]]],
   If[K[[AM[[n]][[m]]]][[2]] < AM[[n]][[m]], AppendTo[AM[[n]],
     K[[AM[[n]][[m]]]][[2]]], AppendTo[AM[[n]], K[[AM[[n]][[m]]]][[3]]]]];
  m++];
 n++]
```

```
n := 1;
AN := {};
While[n < Length[U] + 1, AppendTo[AN, {Min[U[[n]]]}];
 m := 1;
 While[AN[[n]][[m]] ≠ 1, If[K[[AN[[n]][[m]]]][[1]] < AN[[n]][[m]],
   AppendTo[AN[[n]], K[[AN[[n]][[m]]]][[1]]],
   If[K[[AN[[n]][[m]]]][[2]] < AN[[n]][[m]], AppendTo[AN[[n]],
     K[[AN[[n]][[m]]]][[2]]], AppendTo[AN[[n]], K[[AN[[n]][[m]]]][[3]]]]];
  m++];
 n++]
```

This computes a takes each orbit (alternating colour loop) and find the smallest index, before computing a path back to the vertex numbered 1, along a decreasing path.

```
n := 1;
AO := {};
While[n < Length[AL] + 1, AppendTo[AO, {}];
 m := Length[AL[[n]]];
 While[m > 1,
  AppendTo[AO[[n]], AK[[FirstPosition[P, {AL[[n]][[m]], AL[[n]][[m - 1]]}][[2]] +
      48 * (FirstPosition[P, {AL[[n]][[m]], AL[[n]][[m - 1]]}][[1]] - 1)]]];
  m--];
 n++]
```

```
n := 1;
AP := {};
While[n < Length[AM] + 1, AppendTo[AP, {}];
 m := Length[AM[[n]]];
 While[m > 1,
  AppendTo[AP[[n]], AK[[FirstPosition[P, {AM[[n]][[m]], AM[[n]][[m - 1]]}][[2]] +
      48 * (FirstPosition[P, {AM[[n]][[m]], AM[[n]][[m - 1]]}][[1]] - 1)]]];
  m--];
 n++]
```



```
n := 1;
AQ := {};
While[n < Length[AN] + 1, AppendTo[AQ, {}];
 m := Length[AN[[n]]];
 While[m > 1,
  AppendTo[AQ[[n]], AK[[FirstPosition[P, {AN[[n]][[m]], AN[[n]][[m - 1]]}][[2]] +
      48 * (FirstPosition[P, {AN[[n]][[m]], AN[[n]][[m - 1]]}][[1]] - 1)]]];
  m--];
 n++]
```

```
AO1 := Total[AO, {2}]
```

```
AP1 := Total[AP, {2}]
```

```
AQ1 := Total[AQ, {2}]
```

This computes the vector in the first-order chain group corresponding to each of the decreasing paths in the last step and then computes the projection, giving us a position vector for a single vertex in each orbit.

```
n := 1;
AR := {};
While[n < Length[S] + 1, AppendTo[AR, {}];
 m := 1;
 i := FirstPosition[S[[n]], Min[S[[n]]]];
 While[m < Length[S[[n]]] + 1, x := {S[[n]][[Mod[i + m - 2, Length[S[[n]]]] + 1]],
    S[[n]][[Mod[i + m - 1, Length[S[[n]]]] + 1]]};
  AppendTo[AR[[n]], If[Min[x] == x[[1]][[1]], 1, -1] *
    AK[[FirstPosition[P, {Min[x], Max[x]}][[2]] +
       48 * (FirstPosition[P, {Min[x], Max[x]}][[1]] - 1)]]];
  m++];
 n++]
```

```
n := 1;
AS := {};
While[n < Length[T] + 1, AppendTo[AS, {}];
 i := FirstPosition[T[[n]], Min[T[[n]]]];
 m := 1;
 While[m < Length[T[[n]]] + 1, x := {T[[n]][[Mod[i + m - 2, Length[T[[n]]]] + 1]],
    T[[n]][[Mod[i + m - 1, Length[T[[n]]]] + 1]]};
  AppendTo[AS[[n]], If[Min[x] == x[[1]][[1]], 1, -1] *
    AK[[FirstPosition[P, {Min[x], Max[x]}][[2]] +
       48 * (FirstPosition[P, {Min[x], Max[x]}][[1]] - 1)]]];
  m++];
 n++]
```

```
n := 1;
AT := {};
While[n < Length[U] + 1, AppendTo[AT, {}];
 i := FirstPosition[U[[n]], Min[U[[n]]]];
 m := 1;
 While[m < Length[U[[n]]] + 1, x := {U[[n]][[Mod[i + m - 2, Length[U[[n]]]] + 1]],
    U[[n]][[Mod[i + m - 1, Length[U[[n]]]] + 1]]};
  AppendTo[AT[[n]], If[Min[x] == x[[1]][[1]], 1, -1] *
    AK[[FirstPosition[P, {Min[x], Max[x]}][[2]] +
       48 * (FirstPosition[P, {Min[x], Max[x]}][[1]] - 1)]]];
  m++];
 n++]
```

This code computes paths around edge orbit from the chosen point to every other point, computes their first-order chain group element and then the projection.

```
n := 1;
AU := {};
While[n < Length[AR] + 1, AppendTo[AU, {}];
 m := 1;
 While[m < Length[AR[[n]]] + 1,
  AppendTo[AU[[n]], AO1[[n]] + Sum[AR[[n]][[i]], {i, m}]];
  m++];
 n++]
```

```
n := 1;
AV := {};
While[n < Length[AS] + 1, AppendTo[AV, {}];
 m := 1;
 While[m < Length[AS[[n]]] + 1,
  AppendTo[AV[[n]], AP1[[n]] + Sum[AS[[n]][[i]], {i, m}]];
  m++];
 n++]
```

```
n := 1;
AW := {};
While[n < Length[AT] + 1, AppendTo[AW, {}];
 m := 1;
 While[m < Length[AT[[n]]] + 1,
  AppendTo[AW[[n]], AQ1[[n]] + Sum[AT[[n]][[i]], {i, m}]];
  m++];
 n++]
```

This computes the positions of all vertices in the Delaney-Dress graph.

```
n := 1;
AX := {};
While[n < Length[AU] + 1, AppendTo[AX, (1 / Length[AU[[n]]]) * Total[AU[[n]]]];
 n++];
m := 1;
While[m < Length[AV] + 1, AppendTo[AX, (1 / Length[AV[[m]]]) * Total[AV[[m]]]];
 m++];
i := 1;
While[i < Length[AW] + 1, AppendTo[AX, (1 / Length[AW[[i]]]) * Total[AW[[i]]]];
 i++]
```

This code then computes the possibles of all barycentres of all vertices, edges and faces from the cycles described previously.

```
n := 1;
x := Length[AX];
AX1 := {};
While[n < 4, m := 1;
 While[m < 4, i := 1;
  While[i < 4, j := 1;
   While[j < x + 1, AppendTo[AX1, AX[[j]] + {2 * n - 4, 2 * m - 4, 2 * i - 4}];
    j++];
   i++];
  m++];
 n++]
```

This code then shifts these points around by lattice translations to produce a larger array of points.

```
AY := Join[AU, AV, AW]
```

```
n := 1;
AY1 := {};
While[n < 4, m := 1;
 AppendTo[AY1, {}];
 While[m < 4, i := 1;
  AppendTo[AY1[[n]], {}];
  While[i < 4, j := 1;
   AppendTo[AY1[[n]][[m]], {}];
   While[j < Length[AY] + 1, k := 1;
    AppendTo[AY1[[n]][[m]][[i]], {}];
    While[k < Length[AY[[j]]] + 1,
     AppendTo[AY1[[n]][[m]][[i]][[j]], AY[[j]][[k]] + {2 * n - 4, 2 * m - 4, 2 * i - 4}];
     k++];
    j++];
   i++];
  m++];
 n++]
```

```
AY2 := Flatten[AY1, 3]
```

This code shifts the orbits in the Delaney-Dress around the by the same lattice translations.

```
n := 1;
AZ := {};
While[n < Length[AY2] + 1, m := n + 1;
 While[m < Length[AY2] + 1,
  If[Length[Intersection[AY2[[n]], AY2[[m]]]] ≠ 0, AppendTo[AZ, {n - 1, m - 1}]];
  m++];
 n++]
```

This code determine which orbits share points and hence which vertices have an edge in the triangulation.

```
str = OpenWrite[]
```

```
n := 1;
While[n < Length[AX1] + 1, WriteLine[str, StringRiffle[AX1[[n]]]];
 n++]
```

```
n := 1;
While[n < Length[AZ] + 1, WriteLine[str, StringRiffle[AZ[[n]]]];
 n++]
```

```
Close[str]
```

This last bit of code translates all this information into a PLY file. This ply file was then loaded in meshlab to produce the image seen here.

Bibliography

[1] A.W.M.Dress. Presentations of discrete groups, acting on simply connected manifolds, in terms of parametrized systems of coexter matrices - a systematic approach. *Advances in Mathematics*, 63:196–212, 1987.

[2] B.Mohar and C.Thomassen. *Graphs on Surfaces.* The John Hopkins University Press, 2001.

[3] O.D.Friedrichs A.W.M.Dress D.H.Huson J.Klinowski and A.L Mackay. Systematic enumeration of crystalline networks. *Nature*, 400(6745):644–647, Aug 1999.

[4] O.D.Friedrichs. Data structures and algorithms for tilings i. *Theoretical Computer Science*, 303:431–445, 2003.

[5] O.D.Friedrichs and M.O'Keeffe. Identification of and symmetry computation for crystal nets. *Acta Crystallographica*, 59:351–360, 2003.

[6] Vanessa Robins. The Geometry and Topology of Crystals: From Sphere-Packing to Tiling, Nets, and Knots (Invited Talk). In Boris Aronov and Matthew J. Katz, editors, *33rd International Symposium on Computational Geometry (SoCG 2017)*, volume 77 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 1:1–1:1, Dagstuhl, Germany, 2017. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[7] V.Robins S.J.Ramsden and S.T.Hyde. 2d hyperbolic group induce three-periodic euclidean reticulations. *The European Physical Journal B*, 39:365–375, 2004.

[8] T. Sunada. *Topological Crystallography.* Springer, 2013.