2008

Boundary Cycles in Random Triangulated Surfaces

Kevin Fleming
Harvey Mudd College

Boundary Cycles in Random Triangulated Surfaces

Kevin Fleming

Nicholas Pippenger, Advisor

Francis Su, Reader

May, 2008

HARVEY MUDD
COLLEGE
Department of Mathematics

Abstract

Random triangulated surfaces are created by taking an even number, n, of triangles and arbitrarily ”gluing” together pairs of edges until every edge has been paired. The resulting surface can be described in terms of its number of boundary cycles, a random variable denoted by h. Building upon the work of Nicholas Pippenger and Kristin Schleich, and using a recent result from Alex Gamburd, we establish an improved approximation for the expectation of h for certain values of n. We use a computer simulation to exactly determine the distribution of h for small values of n, and present a method for calculating these probabilities. We also conduct an investigation into the related problem of creating one connected component out of n triangles.

Contents

List of Figures

List of Tables

Acknowledgments

I would like to thank my advisor, Professor Nicholas Pippenger, for all of his guidance, insight, and encouragement throughout the course of my thesis. I would also like to thank Professor Francis Su for taking on the responsibilities of being my second reader, and Claire Connelly for her invaluable sample thesis files.

Chapter 1

Introduction

This paper is concerned with the study of the properties of random triangulated surfaces. Such a surface can be constructed in the following manner: begin with an even number, n, of oriented triangles with labeled edges (in this paper, n will always refer to the number of triangles we start with). Randomly select two edges (the edges need not be on different triangles) and identify them together—that is, “glue” them to each other, respecting the orientation of the triangles. Remove these two edges from consideration, so that they may no longer be identified with any edge. We may think of the triangles as flexible, so that any two selected edges can always be identified together, regardless of their positions in the surface being created. After repeating this process until all edges have been identified (note that we require n to be even so that all edges can be paired up), the result will be a random triangulated surface, consisting of one or more closed components. There are several equivalent ways to model these surfaces, each of which we will adopt and use for various purposes as necessary.

1.1 The Ribbon-Graph Model

In order to discuss the idea of boundary components, it is helpful to introduces as the *ribbon-graph model*, as described by Pippenger and Schleich [5]. This model is quite similar to the one previously described. We begin with an even number of oriented triangles and randomly select pairs of edges to identify together. Instead of gluing them together, we attach a ribbon between them with an orientation such that a cycle is formed between the two selected edges and the two sides of the ribbon. We continue this process until all edges have been identified. The resulting surface will correspond to a random triangulated surface; if we replace each ribbon with a simple gluing, we will be left with a triangulated surface.

As an example, consider the following figure, representing the process of creating a random triangulated surface for the case $n = 2$.

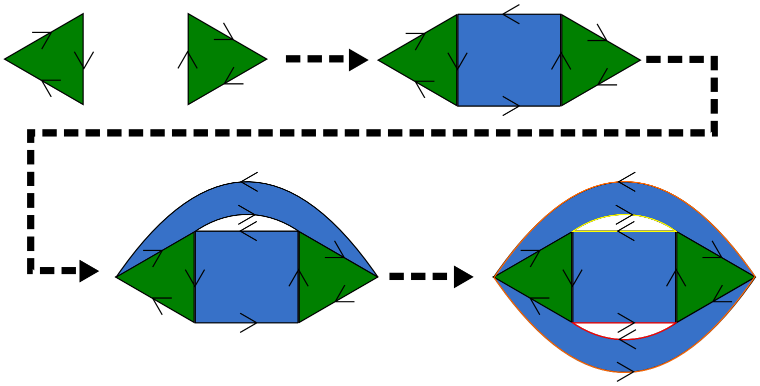

Figure 1.1: Creating a surface from two triangles.

Initially, we have two oriented triangles. Suppose we then select the two edges nearest the center of the figure. We add a ribbon between them with the appropriate orientation. We then pair up the two upper edges, followed by the two lower edges. All edges have been paired, so we are done with our process. We have created surface from two triangles.

1.1.1 Boundary Cycles

A main focus of this paper is the investigation of a property of random triangulated surfaces known as the number of *boundary cycles*. A boundary cycle can best be described using the ribbon-graph model, in conjunction with our previous example. Imagine that the interior of the triangles and the ribbons are filled with some material, and everywhere else is empty space. The boundary cycles are the cycles formed by the oriented ribbon edges that separate the material from the empty space. Referring back to figure 1.1, the three boundary cycles in the final surface are highlighted in red, orange, and green.

The number of boundary cycles in a random triangulated surface made from n triangles, which we will denote h, is a random variable [5]. We can therefore study its probabilistic properties, such as its expected value and distribution.

1.2 The Permutation Model

Rather than thinking about triangles and ribbons, we can also view this problem in terms of permutations. If ρ is a permutation of $3n$ elements composed entirely of 2-cycles (transpositions), σ is a permutation of $3n$ elements composed entirely of 3-cycles, and π is a uniformly distributed random permutation of $3n$ elements, then the number of cycles in $\pi\rho\pi^{-1}\sigma$ has the same distribution as h, the number of boundary cycles in a random triangulated surface from n triangles represented by the permutations [5]. In this case, the elements of σ represent the triangles, and the elements of $\pi\rho\pi^{-1}$ represent the edge identifications. We will refer to the probability distribution of h for permutations constructed in this manner as the $\rho\sigma$ distribution.

1.3 The Cubic Graph Model

Furthermore, we may also view this problem in terms of a cubic graph on n vertices. In this case, the vertices correspond to the triangles, and the edges in the graph correspond to identifications of the sides of the triangles. Suppose that for each vertex, the three edges incident to that vertex are randomly cyclically ordered. Then, when we embed the graph in a surface of minimum genus, the number of cycles (keeping in mind the cyclic orderings of the edges) that bound faces of the graph will have the same distribution as the number of boundary cycles in a random triangulated surface on n triangles [5].

Chapter 2

Previous Work

2.1 An Estimate for the Expected Number of Cycles

Pippenger and Schleich [5] proved the following theorem on the expected value of h, the number of boundary cycles in a random triangulated surface:

Theorem 2.1 *As n tends to infinity through even integers, we have*

$$Ex[h] = \log n + O(1).$$

Their proof involves finding bounds on p_k, the probability that, in a random triangulated surface made of n triangles and described with the ribbon-graph model, a given vertex of a triangle is found in a boundary cycle of length k (length being defined as the number of ribbon edges in a cycle). Specifically, they proved and used the following lemma, which we will use in our later proofs:

Lemma 2.1 *For $1 \leq k \leq n$, we have*

$$p_k \leq \frac{1}{3n-2k+1}\left(1+\frac{k}{3n-2k+5}\right).$$

Based on a computer simulation of random triangulated surfaces, they conjectured, but did not prove, the following:

Conjecture 2.1

$$Ex[h] = \log(3n) + \gamma + o(1),$$

where $\gamma = 0.57721\ldots$ is Euler's constant.

2.2 An Estimate for the Expectation of the Square of h

Pippenger and Schleich [5], in the course of proving an estimate for the variance of h, proved the following:

Theorem 2.2

$$Ex[h^2] = (\log n)^2 + O(\log n).$$

They accomplished this by using the identity

$$Ex[h^2] = \sum_{1 \leq k \leq 3n} \sum_{1 \leq k' \leq 3n} Ex[h_k \cdot h_{k'}],$$

where h_k (respectively, $h_{k'}$) is the number of cycles of length k (respectively, k') in the random triangulated surface. They broke the sum into several parts by considering separate cases and finding estimates for each part. We will later see that a similar approach can be used to prove that

$$Ex[h^l] = O((\log n)^l),$$

where l is any positive integer.

2.3 Closeness to the Uniform Distribution

After the work of Pippenger and Schleich, Alex Gamburd [2] obtained a relevant and applicable result in a paper on random Belyi surfaces. He proved the following theorem:

Theorem 2.3 *Consider a random r-regular graph on m vertices with random orientation. Let A_N denote the alternating group of degree N, and let C_r denote the conjugacy class of A_N consisting of the product of N/r disjoint r-cycles. If $N = rm$, P_r is the probability measure on A_N supported by C_r, and U is the uniform distribution on A_N, then for $r \geq 3$ and as m approaches infinity,*

$$||P_r * P_2 - U|| \to 0.$$

Here

$$||f - g|| = \max_{A \subseteq G} |f(A) - g(A)|$$

is a total variation distance.

Specifically, he proved that

$$||P_r * P_2 - U||^2 = O(N^{-5/2+7/r}) + O(N^{-2}).$$

If we take the square root of both sides of the equation, we arrive at

$$||P_r * P_2 - U|| = O(N^{-5/4+7/2r}) + O(N^{-1}).$$

For our study of random triangulated surfaces, we are considering random cubic graphs, and so $r = 3$, and the approximation becomes $O(N^{-1/12})$. Note that the number of vertices in the graph, m, corresponds to the number of triangles we are considering, n (with an edge between two vertices if they are connected by a ribbon in the ribbon model). Thus we have $N = 3n$. So if we let $Pr_{\rho\sigma}[h = k]$ denote the probability of a total of k cycles under the $\rho\sigma$ distribution and $Pr_U[h = k]$ denote the probability of a total of k cycles under the uniform distribution, then Gamburd's result gives us the following theorem:

Theorem 2.4

$$\left|Pr_{\rho\sigma}[h = k] - Pr_U[h = k]\right| = O\left(\frac{1}{n^{1/12}}\right).$$

This bound, combined with knowledge about the uniform and $\rho\sigma$ distributions, will allow us to prove a weaker version of Conjecture 2.1. As Gamburd's result involves the alternating group A_N, requiring even permutations, we can prove Conjecture 2.1 for values of n for which $n/2$ is even. Thus, we prove the following theorem as one of the main results of this thesis.

Theorem 2.5 *If $n/2$ is even, then*

$$Ex[h] = \log(3n) + \gamma + o(1),$$

where $\gamma = 0.57721\ldots$ is Euler's constant.

Chapter 3

A Proof for the Improved Approximation of Ex[h]

We can prove Theorem 2.5 by combining Gamburd's result, Theorem 2.4, with several of the results from [5]. First, note that the expected value of h for U, the uniform distribution on A_{3n}, is given by

$$Ex_U[h] = H_{3n} = \log(3n) + \gamma + O\left(\frac{1}{n}\right), \tag{3.1}$$

where H_{3n} is the $3n$th harmonic number [1]. Note that for $\alpha > 0$ and $n > 0$,

$$O\left(\frac{1}{n^{\alpha}}\right) = o(1).$$

Therefore, in order to prove Theorem 2.5 it is sufficient to prove that

$$\left|Ex_{\rho\sigma}[h] - Ex_U[h]\right| = O\left(\frac{1}{n^{\alpha}}\right) \tag{3.2}$$

for some $\alpha > 0$.

Note that 3.2 can be rewritten as

$$\left|\sum_{1 \le k \le 3n} k(Pr_{\rho\sigma}[h = k] - Pr_U[h = k])\right| = O\left(\frac{1}{n^{\alpha}}\right). \tag{3.3}$$

We might be tempted to apply the bound in Theorem 2.4 to the left side of the equation. Were we to do so, however, note that we would be adding up $3n$ terms, each of which consists of the product of k, which can be as large as $3n$, and $Pr_{\rho\sigma}[h = k] - Pr_U[h = k]$, which can be as large as $O(n^{-1/12})$. Therefore, simply applying Theorem 2.4 gives us the approximation

$$\left|Ex_{\rho\sigma}[h] - Ex_U[h]\right| = O(n^{2-(1/12)}).$$

As $2 - (1/12) > 0$, this is not sufficient to prove 2.5. Instead, we will split the left side of 3.3 into two sums, and prove the following:

$$\left|\sum_{1 \leq k \leq n^\beta} k(Pr_{\rho\sigma}[h = k] - Pr_U[h = k])\right| = O\left(\frac{1}{n^\alpha}\right), \tag{3.4}$$

$$\left|\sum_{n^\beta < k \leq 3n} k(Pr_{\rho\sigma}[h = k] - Pr_U[h = k])\right| = O\left(\frac{1}{n^\alpha}\right). \tag{3.5}$$

The idea is to look at the terms before and after a cutoff point, n^β, separately. For $k \leq n^\beta$, we can straightforwardly apply the bound in Theorem 2.4. We are adding up n^β terms, and k can be at most n^β, so applying Theorem 2.4, we have something close to 3.4:

$$\left|\sum_{1 \leq k \leq n^\beta} k(Pr_{\rho\sigma}[h = k] - Pr_U[h = k])\right| = O\left(\frac{n^{2\beta}}{n^{1/12}}\right). \tag{3.6}$$

This approach raises the question of how β should be chosen. It will depend on the particular bound that we want to establish (in other words, it will depend on the value of α). Note that in order to obtain 3.4, it is necessary to have $2\beta - \frac{1}{12} < -\alpha$. As long as we ensure that this inequality holds when choosing α and β, this approximation will be good enough to allow us to prove 3.4.

For $k > n^\beta$, however, we cannot apply Theorem 2.4 in this manner, as the resulting approximation will be too large, for the same reasons discussed previously. Instead, we will examine the actual probability that $h > n^\beta$ in both the $\rho\sigma$ and uniform distributions, and show that the probabilities are small enough to prove 3.5.

First we will consider the $\rho\sigma$ distribution. If l is a positive integer, then we have

$$Pr_{\rho\sigma}[h > n^\beta] = Pr_{\rho\sigma}[h^l > n^{\beta l}].$$

Next, recall that Markov's inequality states that for a non-negative-valued random variable X and constant a,

$$Pr[X > a] \leq \frac{Ex[X]}{a}.$$

Applying this to our situation, replacing X with h^l (note that h^l will not take on negative values) and replacing a with $n^{\beta l}$, we obtain

$$Pr_{\rho\sigma}[h^l > n^{\beta l}] \leq \frac{Ex[h^l]}{n^{\beta l}}. \tag{3.7}$$

We can then apply the following Lemma, to be proven later:

Lemma 3.1

$$Ex[h^l] = O((\log n)^l).$$

Applying Lemma 3.1 to 3.7, we obtain, for $k > n^\beta$, the following approximation of the probability that $h > n^\beta$ for the $\rho\sigma$ distribution:

$$Pr_{\rho\sigma}[h > n^\beta] = \frac{O((\log n)^l)}{n^{\beta l}}. \tag{3.8}$$

If we have $k > n^\beta$, then clearly

$$Pr_{\rho\sigma}[h = k] \leq Pr_{\rho\sigma}[h > n^\beta],$$

and therefore, for $k > n^\beta$, we obtain the bound

$$Pr_{\rho\sigma}[h = k] \leq \frac{O((\log n)^l)}{n^{\beta l}}. \tag{3.9}$$

Temporarily turning our attention to the uniform distribution, we can use a Chernoff bound to obtain a good approximation. Note that if $x \geq 1$, and $k > n^\beta$, it is true that

$$\frac{x^k}{x^{n^\beta}} \geq 1,$$

and so we have

$$\begin{aligned} Pr_U[h > n^\beta] &= \sum_{n^\beta < k \leq 3n} Pr_U[h = k] \\ &\leq \frac{1}{x^{n^\beta}} \sum_{n^\beta < k \leq 3n} Pr_U[h = k] x^k \\ &\leq \frac{1}{x^{n^\beta}} f(x), \end{aligned}$$

where $f(x)$ is the probability generating function for h. In order to find $f(x)$, first note that the number of permutations of $3n$ elements that contain exactly k cycles is given by the unsigned Stirling number of the first kind,

$\begin{bmatrix} 3n \\ k \end{bmatrix}$ [1]. The generating function for these numbers (for a fixed value of $3n$) is given by the rising factorial function, $(x)^{(3n)}$, defined as

$$(x)^{(3n)} = x(x+1)(x+2)\cdots(x+3n-1).$$

Recall, however, that 2.4 depends on the permutations being members of the alternating group, and so we will only consider the terms with even exponents in the generating function. Finally, in order to find the probability generating function, we must divide each coefficient by the total number of permutations under consideration, which is $(3n)!/2$. Therefore we have

$$f(x) = \frac{(x)^{(3n)} + (-x)^{(3n)}}{2} \frac{2}{(3n)!}.$$

Then, in order to obtain a bound on $Pr_U[h > n^\beta]$, we can set $x = 2$. This produces

$$Pr_U[h > n^\beta] \leq \frac{1}{2^{n^\beta}} \frac{(2)^{(3n)} + (-2)^{(3n)}}{2} \frac{2}{(3n)!}.$$

Noting that $(-2)^{(3n)} = (-2)(-1)(0)(1)\cdots = 0$, and using the identity

$$(2)^{(3n)} = \frac{(2+3n-1)!}{(2-1)!} = (3n+1)!,$$

we have

$$Pr_U[h > n^\beta] \leq \frac{1}{2^{n^\beta}} \frac{(3n+1)!}{(3n)!} = \frac{3n+1}{2^{n^\beta}}.$$

Noting that any exponential term in n will dominate the linear term $3n+1$, we can write

$$Pr_U[h > n^\beta] = O\left(\frac{1}{2^{n^{\beta/2}}}\right). \tag{3.10}$$

We now have bounds on both $Pr_{\rho\sigma}[h > n^\beta]$ and $Pr_U[h > n^\beta]$. Let us choose α, β, and l such that

$$\beta l \geq \alpha + 3 \tag{3.11}$$

and

$$\beta < \frac{1}{24} - \frac{\alpha}{2}. \tag{3.12}$$

Now we can establish the bound 3.5.

Note that the difference of $Pr_{\rho\sigma}[h = k]$ and $Pr_U[h = k]$ is bounded by their sum, and so we have

$$\begin{aligned}&\left|\sum_{n^\beta < k \leq 3n} k(Pr_{\rho\sigma}[h = k] - Pr_U[h = k])\right| \\ \leq &\sum_{n^\beta < k \leq 3n} k \cdot Pr_{\rho\sigma}[h = k] + \sum_{n^\beta < k \leq 3n} k \cdot Pr_U[h = k].\end{aligned}$$

Noting that in each sum we are adding up fewer than $3n$ terms, and that k can be at most $3n$, we have

$$\begin{aligned}&\sum_{n^\beta < k \leq 3n} k \cdot Pr_{\rho\sigma}[h = k] + \sum_{n^\beta < k \leq 3n} k \cdot Pr_U[h = k] \\ \leq &9n^2 \cdot Pr_{\rho\sigma}[h = k] + 9n^2 \cdot Pr_U[h = k].\end{aligned}$$

Referring to 3.9 and 3.10 to bound $Pr_{\rho\sigma}[h = k]$ and $Pr_U[h = k]$, we have

$$\begin{aligned}&9n^2 \cdot Pr_{\rho\sigma}[h = k] + 9n^2 \cdot Pr_U[h = k] \\ \leq &\frac{9n^2 \cdot O((\log n)^l)}{n^{\beta l}} + 9n^2 O\left(\frac{1}{2^{n^{\beta/2}}}\right).\end{aligned}$$

Noting that a linear term will dominate the logarithmic term and the exponential term will dominate a polynomial term, and applying 3.11, we have

$$\begin{aligned}&\frac{9n^2 \cdot O((\log n)^l)}{n^{\beta l}} + 9n^2 O\left(\frac{1}{2^{n^{\beta/2}}}\right) \\ \leq &\frac{9n^3}{n^{\beta l}} + 9n^2 O\left(\frac{1}{n^{\beta l}}\right) \\ \leq &\frac{9}{n^{\beta l - 3}} + 9n^3 \frac{1}{n^{\beta l}} \\ = &\frac{9}{n^{\beta l - 3}} + \frac{9}{n^{\beta l - 3}} \\ = &\frac{18}{n^{\beta l - 3}} \\ = &O\left(\frac{1}{n^\alpha}\right).\end{aligned}$$

Thus we have established the bound 3.5.

Note that when 3.12 is applied to 3.6, we establish the bound 3.4. Finally, combining 3.4 and 3.5 yields the desired bound in 3.2. Note that α can be chosen arbitrarily close to $\frac{1}{12}$ from below. The closer α is to $\frac{1}{12}$, the smaller β is, becoming arbitrarily close to 0, and the larger l is.

3.1 A Bound on the Moments of h

We now return our attention to Lemma 3.1. In order to prove this bound on the moments of h, we use an inductive version of Pippenger and Schleich's [5] proof of Theorem 2.2, and use their result that

$$Ex[h^2] = O((\log n)^2)$$

as the base case.

Supposing that

$$Ex[h^{l-1}] = O((\log n)^{l-1}),$$

consider $Ex[h^l]$. We can write this as

$$\sum_{1\leq k_1 \leq 3n} \cdots \sum_{1\leq k_l \leq 3n} Ex[h_{k_1} \cdots h_{k_l}],$$

where h_{k_i} is the number of cycles of length k_i. As in [5], we can reduce the summation ranges. In this case we sum up to ϵn instead of $3n$, choosing ϵ so that the extra summands are absorbed into one big O term. Thus we have

$$\sum_{1\leq k_1 \leq 3n} \cdots \sum_{1\leq k_l \leq 3n} Ex[h_{k_1} \cdots h_{k_l}] \leq \sum_{1\leq k_1 \leq \epsilon n} \cdots \sum_{1\leq k_l \leq \epsilon n} Ex[h_{k_1} \cdots h_{k_l}] + O(Ex[h^{l-1}]),$$

and, keeping in mind our supposition, we have

$$\sum_{1\leq k_1 \leq 3n} \cdots \sum_{1\leq k_l \leq 3n} Ex[h_{k_1} \cdots h_{k_l}] \leq \sum_{1\leq k_1 \leq \epsilon n} \cdots \sum_{1\leq k_l \leq \epsilon n} Ex[h_{k_1} \cdots h_{k_l}] + O((\log n)^{l-1}).$$

Thus, to prove Lemma 3.1, it suffices to show that

$$\sum_{1\leq k_1 \leq \epsilon n} \cdots \sum_{1\leq k_l \leq \epsilon n} Ex[h_{k_1} \cdots h_{k_l}] = O((\log n)^l).$$

As in [5], we can rewrite this sum as

$$\sum_{c_1} \cdots \sum_{c_l} \sum_{1\leq k_1 \leq \epsilon n} \cdots \sum_{1\leq k_l \leq \epsilon n} \frac{Pr[c_1 \text{ in } k_1, \ldots, c_l \text{ in } k_l]}{k_1 \cdots k_l},$$

where we are summing over all vertices, and "c_i in k_i" represents the state of vertex c_i being in a cycle of length k_i.

As in [5], we partition the sum into cases based on which of the vertices we are summing over are in the same or different triangles or cycles. The dominant sum is the case for which all of the vertices are in different cycles and different triangles, and it suffices to consider this sum to establish the desired estimate. For the case in which all of the vertices are in different triangles, we have

$$\begin{aligned}
&\sum_{c_1}\sum_{c_2 \not\sim c_1}\cdots\sum_{c_l \not\sim c_1,\ldots,c_l \not\sim c_{l-1}}\ \sum_{1\le k_1\le \epsilon n}\cdots\sum_{c\le k_l\le \epsilon n}\frac{Pr[c_1 \text{ in } k_1,\ldots,c_l \text{ in } k_l]}{k_1\cdots k_l}\\
&=\binom{n}{l}\times l!\times 3^l\sum_{1\le k_1\le \epsilon n}\ \sum_{k_1\le k_2\le \epsilon n}\cdots\sum_{k_{l-1}\le k_l\le \epsilon n}\frac{Pr[c_1 \text{ in } k_1,\ldots,c_l \text{ in } k_l]}{k_1\cdots k_l},
\end{aligned}$$

where $c_1 \not\sim c_2$ means "vertex c_1 is not adjacent to vertex c_2". The factors that replace the sums over the vertices represent choosing the l triangles that will contain the vertices, permuting them in any way, and then choosing one vertex from each triangle to be the one under consideration.

Now for the case where all of the vertices are in disjoint cycles, we have

$$\begin{aligned}
&\sum_{1\le k_1\le \epsilon n}\cdots\sum_{k_{l-1}\le k_l\le \epsilon n}\frac{Pr[c_1 \text{ in } k_1, c_2 \text{ in disjoint } k_2,\ldots,c_l \text{ in disjoint } k_l]}{k_1\cdots k_l}\\
&\le (3n)\sum_{k_1}\frac{Pr[c_1 \text{ in } k_1]}{k_1}(3n)\sum_{k_2}\frac{Pr[c_2 \text{ in disjoint } k_2|c_1 \text{ in } k_1]}{k_2}\times\cdots \qquad (3.13)\\
&\qquad\times(3n)\sum_{k_l}\frac{Pr[c_l \text{ in disjoint } k_l|c_1 \text{ in } k_1,\ldots,c_{l-1} \text{ in disjoint } k_{l-1}]}{k_l}.
\end{aligned}$$

Thus, we have a product of l terms that each look like $3n$ times a sum of probabilities. Taking as a representative example the first term and applying Lemma 2.1, we have

$$(3n)\sum_{k_1}\frac{Pr[c_1 \text{ in } k_1]}{k_1}\le(3n)\sum_{k_1}\frac{1}{3n-2k_1+1}\left(1+\frac{k_1}{3n-2k_1+5}\right)\frac{1}{k_1}.$$

We can write this as

$$(3n)\sum_{k_1}\frac{1}{3n-2k_1+1}\times\frac{1}{k_1}+(3n)\sum_{k_1}\frac{1}{3n-2k_1+1}\times\frac{k_1}{3n-2k_1+5}\times\frac{1}{k_1}.$$

For both of these terms, the $3n$ in front of the sum and the $3n$ in the denominator will obliterate each other. Furthermore, in the second term, the k_1 in the numerator will be canceled by the k_1 in the denominator, so this term will be $O(1)$. In the first term, there is no k_1 in the numerator, so the k_1 in the denominator remains, and thus we have a sum over k_1 of $\frac{1}{k_1}$, which is $O(\log n)$.

The cases for $k_2, k_3, \ldots, k_l$ are similar. Thus each factor in the right side of 3.13 looks like $O(\log n) + O(1)$, and multiplying them gives $O((\log n)^l)$, completing our proof of Lemma 3.1.

Chapter 4

Computing the Exact Distribution of h for Small Values of n

One aspect of h that has not undergone in-depth examination is its exact distribution. In other words, given a value of n, exactly how many ways are there to identify edges together to obtain a surface with h boundary cycles, for all possible values of h? The case for $n = 2$ is easily solvable by hand, but the cases for all other values of n call for a computer simulation due to their sizes.

4.1 How to Create an Efficient Algorithm

Using a computer, we can implement an algorithm that goes through the process of identifying pairs of edges until all edges have been used. When a closed surface is obtained, we note its number of boundary cycles. Once we account for all possible surfaces that can be created, we will have the exact distribution of h for the value of n that was used. The total number of ways to identify all $3n$ edges in some order to create a closed surface is $(3n-1)!! = (3n-1) \times (3n-3) \times \cdots \times 3 \times 1$, as explained in [5]. An implementation of a "brute force" algorithm that constructed each of these $(3n-1)!!$ surfaces, one at a time, would only be able to compute the distribution in a reasonable amount of time for the first few values of n. A more efficient method is to recognize when steps in the identification process are equivalent; that is, when they lead to the same surface. By watching for this and controlling which edges we identify at each step, we can calculate the exact distribution of h for much larger values of n, compared to the brute force algorithm.

Note that any identification of two edges falls into exactly one of three categories. We will refer to these categories as options 1, 2, and 3.

1. The two edges are in the same cycle. This shortens the length of the cycle by two (if the edges are next to each other) or splits the cycle into two (if they are not next to each other). The genus of the connected component containing the edges remains unchanged.

2. The two edges are in different cycles in the same connected component. The two cycles merge into one, and the genus of the connected component increases by one.

3. The two edges are in different connected components. The two components merge into one. The cycles containing the two edges also merge into one, as in option 2, and the new component contains all the other cycles from the original components. Furthermore, the genus of the new component is equal to the sum of the genera of the original components.

Some of this information (in particular, options 1 and 2) was adapted from a recurrence relationship for a similar problem presented in [4].

For each of these options, given one fixed edge that will be identified with some other chosen edge, there can be multiple identification partners that will result in the same surface.

1. For option 1, choosing a partner that is i edges away from the fixed edge, traveling clockwise around the cycle, results in the same surface as choosing a partner that is i edges away from the fixed edge, traveling counter-clockwise.

2. For option 2, choosing as a partner any edge from a cycle of a given length results in the same surface as choosing any edge in any other cycle (in the same connected component) of the same length.

3. For option 3, choosing as a partner any edge in a cycle of a given length in some connected component results in the same surface as choosing any edge in a cycle of the same length in an identical connected component (that is, one that has the same cycle lengths and genus).

This idea of equivalent identification partners can greatly reduce the number of surfaces that must be constructed. For example, take the first step of the process. We have an edge that we want to find a partner for. Essentially we have two choices: pick a partner from the same triangle (this corresponds to option 1), or pick a partner from a different triangle (option 3). Note that option 2 is not a possibility here since none of the components have multiple cycles. If we were doing this by brute force, we would have to consider separately the two equivalent surfaces that result from choosing a partner from the same triangle, as well as the $3(n-1)$ equivalent surfaces that result from choosing a partner from a different triangle. By considering equivalent partners, we only have to consider two surfaces: the one that results from option 1, and the one that results from option 3.

Of course, we have to take into account the fact that there are only two ways to obtain the former surface, while there are $3(n-1)$ ways to obtain the latter. This introduces the idea of giving each surface a weight, which counts the total number of ways to start with n triangles and identify edges in some order to arrive at the surface in question. In the example above, the former surface would have a weight of 2, while the latter would have a weight of $3(n-1)$. In order to introduce some more useful terms and further set the stage for the algorithm that will shortly be introduced, we now introduce a useful way to visualize this problem.

4.1.1 Traversing a Graph

The process of constructing a surface by identifying edges can be mapped to taking a path from the source of a directed graph to one of its sinks. Every node in this graph represents a surface, and one node will be the parent of another if there is a way to identify two edges in the parent surface and thereby obtain the child surface. Thus, each edge in the graph represents an identification of two triangle edges with each other. The source of the graph represents the initial surface consisting of n triangles, the sinks represent the closed surfaces with no edges left to identify, and all other nodes represent "intermediate" surfaces, where some, but not all, edges have been identified. Also associated with each node is a weight. The weight of the source is 1, while the weights of the other nodes have the meaning introduced above. The graph for the $n = 2$ case is pictured below, where edge identifications are shown by either placing the two edges on top of each other or connecting them with a blue line. Note that in this graph, the left sink represents a sphere, while the right sink represents a torus. Also note that it is possible for one node to have multiple parents.

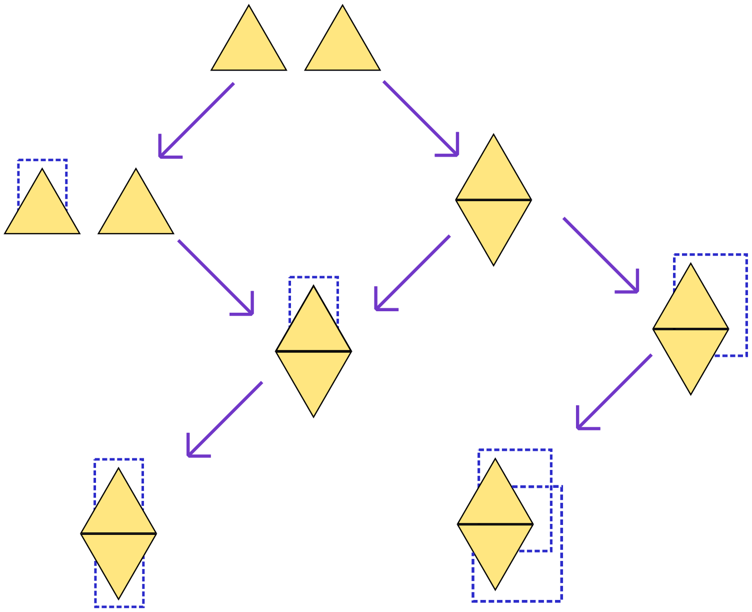

Figure 4.1: Graph for $n = 2$.

The weights for the non-source nodes can be calculated with the following formula, where $w(\text{node})$ is the weight of a node, and $t(\text{parent}, \text{child})$ is the number of ways to identify two edges in the parent node and thereby obtain the child node:

$$w(c) = \sum_{p \text{ is a parent of } c} w(p)t(p,c).$$

This formula makes intuitive sense when considering the meaning of weight: if, for example, a child has only one parent, there are a ways to get from the source to the parent, and there are b ways to get from the parent to the child, then there are ab ways to get from the source to the child. For multiple parents we simply add the weights.

Once the weights for all the nodes have been calculated, the distribution of h can be found by examining the weights of the sinks. These will tell us how many ways there are to arrive at any particular closed surface by identifying edges.

One more idea that simplifies the algorithm is that of an "active component". This refers to the concept of always choosing one of the two edges to be identified from a particular connected component (referred to as the active component). It then remains to pick this edge's identification partner. When all edges of the active component have been identified, the component is closed, and one of the remaining non-closed components becomes the active one. The reason this simplifies things is that it makes option 3 much easier to handle. Note that initially, every connected component is a triangle. With options 1 and 2, the active component changes in some way, but the other components are unaffected. With option 3, one of the other components becomes "absorbed" into the active one, which grows, and everything else stays the same. The implication of this is that *every non-closed, non-active connected component will always be a triangle*. Therefore, at any step in the process, there is always only one distinct surface that can be made with option 3, and the number of ways to make this surface is equal to three times the number of triangles remaining. A further implication of this is that any surface, be it source, sink, or intermediate, can be completely described by three things: a list of the genera of the closed components, a list of the lengths of the cycles in the active component, and the number of remaining triangles.

Finally, it is worth noting that in order to accurately calculate the weights for every node, the graph must be constructed in a breadth-first manner. In other words, first we consider the source, then all the children of the source, then all the children of those nodes, and so on. This is because it is necessary to have knowledge of the weights of every member of a "generation" of surfaces (that is, all of the surfaces that result after a fixed number of identifications) before finalizing the weights of the next generation. A benefit of this approach is that after we use one generation to obtain the next, the older generation is obsolete and can be "deleted" to save space. Ultimately, all that is necessary to calculate the distribution of h are the sinks of the graph and their weights.

4.2 Description of an Efficient Algorithm to Calculate the Distribution of h

With these methods for improving the efficiency of the process of constructing closed surfaces, we are ready to use them in an algorithm. The following is a description of an algorithm that takes into account the ideas of equivalent identification partners, surface weights, and active components. The details of each kind of edge identification (how the active component changes, and so on) are left out, but can be found above. An implementation of this algorithm in C++ can be found at `http://math.hmc.edu/seniorthesis/archives/2008/kfleming/`, or by searching online for “h_distribution.cpp”.

```
create a queue of surfaces and add the initial surface ($n$ triangles);
while the queue is not empty do
    pop the first surface off the queue (call it "parent");
    if there are edges left in parent to be identified then
        choose a fixed edge from the active component;
        for each edge in the same cycle as the fixed edge, traveling in one
        direction, until the halfway point is reached do
            construct the child surface by identifying the fixed edge
            and the chosen edge (option 1);
            if the chosen edge is not the edge at the halfway point then
                $w(\text{child}) := w(\text{child}) + w(\text{parent}) * 2$;
            else
                $w(\text{child}) := w(\text{child}) + w(\text{parent})$;
            end
            if child is not already in the queue then
                add child to the queue;
            end
        end
        for each cycle of distinct length in the active component besides the
        one containing the fixed edge do
            count how many other cycles in the active component
            have the same length;
            construct the child surface by identifying the fixed edge
            and one of the edges in the chosen cycle (option 2);
            $w(\text{child}) := w(\text{child}) + w(\text{parent}) *$
            length of chosen cycle $*$ copies of chosen cycle;
            if child is not already in the queue then
                add child to the queue;
            end
        end
        construct a child surface by identifying the fixed edge with an
        edge in one of the remaining triangles (option 3);
        $w(\text{child}) := w(\text{child}) + w(\text{parent}) * 3 *$ number of triangles;
        if child is not already in the queue then
            add child to the queue;
        end
        remove parent's node from the graph;
    end
end
for each closed surface (sink) do
    surface chi := 0;
    for each closed component in the surface do
        surface chi := surface chi + (2 - 2 * genus of component);
    end
    record the fact that a surface with (surface chi + n/2) boundary
    cycles can be made in $w(\text{surface})$ ways;
end
```

Figure 4.2: Improved algorithm to caculate the exact distribution of h.

Chapter 5

The Distribution of Boundary Cycles and Spherical Numbers

With the implementation of the previously described algorithm, we can examine the exact distribution of h for small values of n. The remainder of this paper will be devoted to the investigation of these results.

5.1 Results of the Implementation

This improved algorithm has a running time that grows like the number of integer partitions of n. The connection to partitions comes from the fact that any surface's active component can be thought of as a partition of edges into cycles. Hardy and Ramanujan [3] show that the number of partitions, $p(n)$, is asymptotic to

$$\frac{1}{4\sqrt{3}n} e^{\pi\sqrt{\frac{2}{3}n}}.$$

Compare this to the number of surfaces to consider with a brute force approach, $(3n-1)!!$. This is asymptotic to

$$\sqrt{2}\left(\frac{3n}{e}\right)^{3n/2}.$$

Note that the former approximation is exponential in $\sqrt{n}$, while the latter is exponential in $n \log n$. As further evidence for the algorithm's improvement over a brute force approach, consider the values of these approximations for the first few values of n:

n	Approximation for $p(n)$	Approximation for $(3n-1)!!$
2	2.7	15.2
4	6.1	10467.3
6	12.9	3.4×10^{7}
8	25.5	3.1×10^{11}
10	48.1	6.2×10^{15}
12	86.9	2.2×10^{20}
14	151.9	1.3×10^{25}
16	257.8	1.2×10^{30}

Table 5.1: Comparison of approximations: improved algorithm vs. brute force.

On a personal computer, a C++ implementation of the algorithm was used to calculate the exact distribution of h for values of n up to 50, with the largest case taking about 30 minutes to solve. The output consists of frequency tables such as the following:

h	n = 4	n = 8	n = 12	n = 16
2	4536	89671548960	47613095537007369600	208492455487140641410513920000
4	5184	167751131625	111859431114774198240	568343345836882239259796160000
6		54201344400	53363490454533032235	337745361781304748044070897600
8		4448999520	8246689198672949640	70933949447208151335966759825
10		158941440	539931283054122960	6695489931194134994751828000
12		2177280	20005927297171200	345798071730794314463832000
14			446296688006400	11522850716579469185318400
16			5633633617920	265078151618980232332800
18			31039303680	4244306327710357708800
20				45503421164122521600
22				294595272799027200
24				871583647334400

Table 5.2: Sample frequency tables for $n = 4, 8, 12, 16$.

The full tables can be found at `http://math.hmc.edu/seniorthesis/archive/2008/kfleming/`, or by searching online for "boundary_cycle_distributions.txt".

5.2 Examination of the Frequency Tables

5.2.1 The Frequencies mod 10

Examining the tables produced brings about a conjecture:

Conjecture 5.1 *For even $n \geq 6$, the number of surfaces constructed from n triangles and having exactly h boundary cycles is congruent to 5 (mod 10) if $h = n/2$ and is congruent to 0 (mod 10) otherwise.*

One approach to proving this conjecture would be to examine the factors that the weights are multiplied by at each step in the surface construction process described in chapter 4. Specifically, for $n \geq 6$, there will be some point in the process where exactly five triangles remain outside of the active component. One of the edges in these five triangles will eventually have to be identified with an edge in the active component, and this step introduces a factor of 5 to the weight of the child surface. From empirical examination of the data, it appears that the only surface with $n/2$ boundary cycles that has a weight congruent to 5 (mod 10) is the surface consisting of $n/2$ torii. It remains to be shown that the construction of this surface does not introduce a factor of 2 to the weight, but the construction of all other surfaces does. The factor of 2 seems likely to come from identifying two edges in the same cycle, since there are almost always two equivalent ways to do this.

5.2.2 Calculating Frequencies

Pippenger and Schleich [5] obtained an exact formula for the probability that a surface constructed from n triangles, with $n/2$ odd, contains exactly one boundary cycle. The case of having one boundary cycle corresponds to the first lines of the frequency tables for values of n for which $n/2$ is odd.

There is a straightforward, if lengthy, way to calculate the value of any line a given distance from the bottom of any table. This method, however, relies on constants found in a different set of tables produced by a slightly modified version of the aforementioned C++ implementation. The tables in question list the number of ways to start with n triangles and identify edges to obtain a surface with *one* connected component, with a given Euler characteristic, denoted χ. Here are the tables for the first few values of n:

χ	n = 2	n = 4	n = 6	n = 8	n = 10
-4					357485480352000
-2			3061800	89414357760	2834113460935680
0	3	4536	19362240	164367221760	2332019568291840
2	12	5184	9797760	45148078080	392212641300480

Table 5.3: Number of ways to create one component from n triangles.

In other words, there are 5184 ways to identify the edges of four triangles to create a sphere (Euler characteristic 2), and so on.

As an example of the method to calculate these lines, we will look at the cases $h = 3n/2$ and $h = 3n/2 - 2$ (the last two lines of any given frequency table).

For the case $h = 3n/2$: the Euler characteristic of a surface with $3n/2$ boundary cycles will be $3n/2 - n/2 = n$. Noting that at least two triangles are needed to create any closed component, we see that there are a maximum of $n/2$ closed components in any surface created from n triangles. The maximum Euler characteristic of any closed component is 2, corresponding to a sphere. Thus we see that any surface with total Euler characteristic n must have $n/2$ spheres and nothing else. Every sphere must be made from two triangles, and there are no triangles left over to make any components. From our second tables we note that there are 12 ways to create a sphere from two triangles. Thus the process for creating a surface with $3n/2$ boundary cycles can be thought of as repeating these two steps until all triangles have been used:

1. Pick two triangles from the remaining unused triangles.

2. Create a sphere from the two chosen triangles.

The number of ways to do the first step is a binomial coefficient, choosing two from $n - k$, where k is the number of spheres already created. The number of ways to do the second step, as we have seen, is 12. Finally, forming the spheres in any order results in the exact same surface (all edges are identified in the same way), so we must divide by a factor of $(n/2)!$. Thus the total number of ways to create a surface with $3n/2$ boundary cycles is

$$\begin{aligned}
&12\binom{n}{2} \times 12\binom{n-2}{2} \times \times 12\binom{n-4}{2} \times \cdots \times 12\binom{n-(n-2)}{2}\frac{1}{(n/2)!}\\
&= 12^{n/2}\binom{n}{2}\binom{n-2}{2}\binom{n-4}{2}\cdots\binom{n-(n-2)}{2} \times \frac{1}{(n/2)!}\\
&= 12^{n/2}\frac{n!}{2!(n-2)!}\frac{(n-2)!}{2!(n-4)!}\frac{(n-4)!}{2!(n-6)!}\cdots\frac{(n-(n-2))!}{2!(n-n)!}\frac{1}{(n/2)!}\\
&= 12^{n/2}n!\frac{1}{2^{n/2}}\frac{1}{(n/2)!}\\
&= 12^{n/2}(n-1)!!.
\end{aligned}$$

The case for $h = 3n/2 - 2$ is similar, but slightly more complicated. A similar examination of the necessary Euler characteristic, $n - 2$, reveals that any such surface must consist of either one torus made of two triangles and $n/2-1$ spheres made of two triangles each, or one sphere made of four triangles and $n/2-2$ spheres made of two triangles each. Using the same process of choosing triangles, forming them into the necessary shapes, and then permuting like shapes, we have that the number of ways to create a surface with $3n/2-2$ boundary cycles is

$$\begin{aligned}
&12^{n/2-1} \times 3 \times \binom{n}{2}\binom{n-2}{2} \times \cdots \times \binom{n-(n-2)}{2}\frac{1}{(n/2-1)!} \\
&\quad + 12^{n/2-2} \times 5184 \times \binom{n}{4}\binom{n-4}{2}\binom{n-6}{2} \times \cdots \times \binom{n-(n-2)}{2}\frac{1}{(n/2-2)!} \\
&= 12^{n/2-1} \times 3 \times \frac{n!}{2(n-2)!!} + 12^{n/2-2} \times 5184 \times \frac{n!}{4!(n-4)!!}.
\end{aligned}$$

Note that the constants 3, 12, and 5184 were obtained from the second set of tables, discussed above.

In general, this method could be applied to calculate the value of any line of the original tables a given distance from the bottom; the calculations for smaller values of h would simply be more lengthy as there are more possible surfaces.

5.3 Spherical Numbers

Finally, we turn our attention to a problem introduced by the method of calculating frequencies. As we have seen, our process for such calculations involves knowing exactly how many ways there are to turn some number of triangles into one connected component. Without formulae for these numbers, however, the method is not much help. How, then, can we calculate these numbers? To reduce the scope of this new topic, we will focus solely on the problem of identifying the edges in n triangles to obtain one closed component with Euler characteristic 2–that is, a sphere. For brevity, let us say that the *spherical number* s_n is the number of ways of identifying the edges in n triangles to obtain one sphere. From the previous table, we see that the first few spherical numbers, s_2, s_4, and s_6, are 12, 5184, and 9797760. The numbers up to s_{50} can be found at `http://math.hmc.edu/seniorthesis/archive/2008/kfleming/`, or by searching online for "spherical_numbers.txt".

Unfortunately, this problem seems to be quite difficult, and an exact formula for s_n has proven elusive. What follows is a description of some of the properties of s_n and some attempted methods of obtaining a formula that may eventually lead to the solution.

From empirical examination of s_2 through s_{50}, s_n seems to factor into small primes. Furthermore, it seems to factor into most of the primes up to a certain limit, with the limit depending on the parity of $n/2$. Specifically, for $n/2$ even, s_n factors into primes no larger than n, and for $n/2$ odd, s_n factors into primes no larger than $3n/2$. Additionally, the factorization of s_n for $n/2$ even contains many more 2s than for $n/2$ odd. These observations strongly suggest that a double factorial of $3n/2$ may be involved. It is not clear, however, what such a double factorial might represent, and a double factorial of an even number (as would be the case for $n/2$ even) has yet to be seen in our investigation of random triangulated surfaces. Even if not $(3n/2)!!$ exactly, the factorization of s_n strongly suggests the presence of one or more factorials, along with some relatively large but undetermined number of 2s and 3s.

Note that a surface made from n triangles can be drawn as in the following figure (for this example, $n = 4$), where identified edges are either drawn over each other or connected with a line. We can view the drawing as consisting of one $n + 2$-sided polygon with some number of *internal pairings* (the pairings represented by drawing two edges on top of each other), with the sides of the polygon connected with *external pairings* (the pairings represented by drawing lines between two edges). The surface represented by a such a drawing will be a sphere if and only if the external pairing lines do not cross; any crossings will introduce handles to the surface. For this example, internal pairings are indicated in blue, and external pairings are indicated by dashed lines.

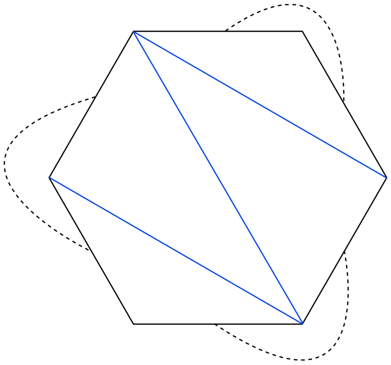

Figure 5.1: Representation of a sphere from 4 triangles.

Therefore, one way to create a sphere from n triangles is to first use all the triangles to make one $n+2$-sided polygon, and then identifying the sides of the polygon in a valid (non-crossing) way. Note that the number of ways to do the second step is the Catalan number C_n. This approach is more complicated than it might appear, however, as the distinction between internal and external pairings is a rather artificial one. We must make sure that we avoid overcounting; for example, we might count two surfaces as distinct when they have the same set of edge identifications, but with some reclassified from internal to external, and vice-versa. There does not seem to be a simple way to avoid this problem.

Another approach is to try to find a recurrence for s_n, by taking a completed sphere with $n-2$ triangles, adding on two triangles to make a bigger polygon, and filling in the required external pairings. This is greatly complicated, however, by the fact that adding on triangles in different places on the polygon, depending on the already-existing identifications, will result in different numbers of ways to complete the external pairings–sometimes none at all.

Alternatively, we can represent spheres from n triangles as planar cubic connected graphs on n vertices, with loops and multiple edges allowed. As discussed in the introduction, vertices represent triangles, and an edge between two vertices represents an identification of edges between triangles (a loop represents pairing two edges on the same triangle). The five possible graphs for $n = 4$ are shown below.

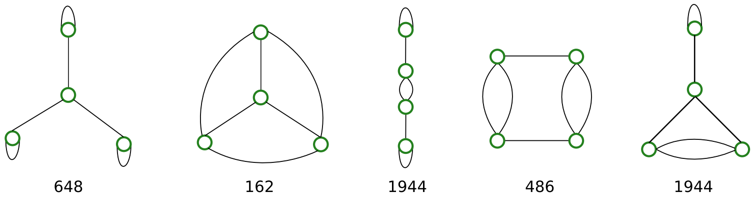

Figure 5.2: All five cubic planar connected graphs on 4 vertices.

After identifying the possible cubic planar connected graphs on n vertices, we can then combinatorically determine how many surfaces are represented by each graph. For the four-vertex graphs shown above, the graphs represent 648, 162, 1944, 486, and 1944 surfaces (read left to right), for a total of 5184 surfaces. Therefore we have broken down all 5184 possible spheres into five different classes. As an example of how to calculate these numbers, take the leftmost graph. Assign a triangle to each vertex; there are 4!

ways to do this. For each of the triangles with a loop, choose which edge will be identified with an edge in the center triangle. There are 3^3 ways to do this. Finally, for the center triangle, choose which edges will be identified with which outer triangles. There are 3! ways to do this. Finally, we must divide by 3! since permuting the three outer triangles can lead to the same set of edge identifications. Thus there are a total of

$$\frac{4! \times 3^3 \times 3!}{3!} = 4! \times 3^3 = 648$$

spheres represented by this graph. Performing similar calculations on the other four graphs, we arrive at the numbers presented above. There may be some way to more easily calculate these numbers, perhaps based on graph invariants.

Bibliography

[1] Arthur Benjamin, Gregory Preston, and Jennifer Quinn. A Stirling encounter with harmonic numbers. *Mathematics Magazine*, 75:95–103, 2002.

[2] Alex Gamburd. Poisson-Dirichlet distribution for random Belyi surfaces. *The Annals of Probability*, 34:1827–1848, 2006.

[3] Godfrey Hardy and Srinivasa Ramanujan. Asymptotic formulae in combinatorial analysis. *Proc. London Math Soc.*, 17:75–115, 1918.

[4] John Harer and Don Zagier. The Euler characteristic of the moduli space of curves. *Inventiones Mathematicae*, 85:457–485, 1986.

[5] Nicholas Pippenger and Kristin Schleich. Topological characteristics of random triangulated surfaces. *Random Structures and Algorithms*, 28:247–288, 2006.