Computing a Link Diagram from Its Exterior

Nathan M. Dunfield
Dept. of Math., University of Illinois at Urbana-Champaign, IL, USA

Malik Obeidin
Google, Inc., Mountain View, CA, USA

Cameron Gates Rudd
Dept. of Math., University of Illinois at Urbana-Champaign, IL, USA

Abstract

A knot is a circle piecewise-linearly embedded into the 3-sphere. The topology of a knot is intimately related to that of its exterior, which is the complement of an open regular neighborhood of the knot. Knots are typically encoded by planar diagrams, whereas their exteriors, which are compact 3-manifolds with torus boundary, are encoded by triangulations. Here, we give the first practical algorithm for finding a diagram of a knot given a triangulation of its exterior. Our method applies to links as well as knots, and allows us to recover links with hundreds of crossings. We use it to find the first diagrams known for 23 principal congruence arithmetic link exteriors; the largest has over 2,500 crossings. Other applications include finding pairs of knots with the same 0-surgery, which relates to questions about slice knots and the smooth 4D Poincaré conjecture.

2012 ACM Subject Classification Mathematics of computing → Geometric topology

Keywords and phrases computational topology, low-dimensional topology, knot, knot exterior, knot diagram, link, link exterior, link diagram

Digital Object Identifier 10.4230/LIPIcs.SoCG.2022.37

Related Version *Full Version*: `https://arxiv.org/abs/2112.03251v2`

Supplementary Material *Software (Source Code and Data)*: `https://doi.org/10.7910/DVN/BT1M8R`

Funding *Nathan M. Dunfield*: Partially supported by US National Science Foundation grants DMS-1510204 and DMS-1811156 and by a Simons Fellowship.
Malik Obeidin: Partially supported by US National Science Foundation grants DMS-1510204 and DMS-181115.
Cameron Gates Rudd: Partially supported by US National Science Foundation grant DMS-1811156.

Acknowledgements We thank Matthias Goerner and Henry Segerman for helpful correspondence, and thank the referees for their detailed comments which helped improve this paper.

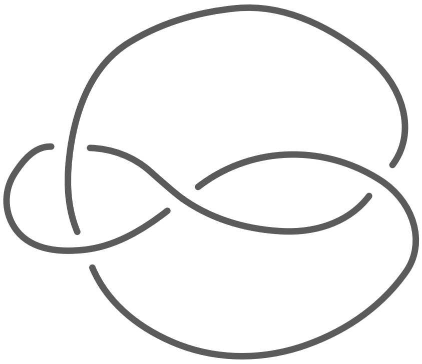


Figure 1 A planar diagram for a knot is a 4-valent graph with a planar embedding where every vertex represents a *crossing*, a place where one part of the knot crosses in front of the other in 3D.

38th International Symposium on Computational Geometry (SoCG 2022).
Editors: Xavier Goaoc and Michael Kerber; Article No. 37; pp. 37:1–37:24
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

A knot is a piecewise-linear (PL) embedding of a circle S^1 into the 3-sphere S^3. The study of knots goes back to the 19th century, and today is a central focus of low-dimensional topology, with applications to chemistry [23], biology [24], engineering [40], and theoretical computer science [16]. Two knots are topologically equivalent when they are isotopic, that is, when one can be continuously deformed to the other without passing through itself. Computationally, knots are typically encoded as planar diagrams (Figure 1); there are more than 350 million distinct knots with diagrams of at most 19 crossings as enumerated by [11].

The topology of knots is intimately related to that of their exteriors, where the *exterior* of a knot K is the compact 3-manifold with torus boundary $E(K) := S^3 \setminus N(K)$ where $N(K)$ is an open tubular neighborhood of K. Indeed, the exterior $E(K)$ determines the knot K [26]. Many algorithms for knots work via their exteriors, starting with Haken's foundational method for deciding when a knot is equivalent to a round circle [27]. Consequently, the problem of going from a diagram D of K to a triangulation of $E(K)$ is well-studied [28, §7]; for ideal triangulations (see Section 2.1 below), one needs only four tetrahedra per crossing of D [51, §3]. Here, we study the inverse problem:

▶ **Find Diagram.** *Input a triangulation $\mathcal{T}$ of a knot exterior $E(K)$, output a diagram of K.*

If the input triangulation $\mathcal{T}$ is guaranteed to be that of a knot exterior (in fact, this is decidable by Algorithm S of [33]), then a useless algorithm to find D is just this: start generating all knot diagrams, triangulate each exterior, and then do Pachner moves (see Section 2.4) on these triangulations. Since any two triangulations of a compact 3-manifold are connected by a sequence of such moves, one eventually stumbles across $\mathcal{T}$, thus finding a diagram for the underlying knot. We do not explore the computational complexity of FIND DIAGRAM here (though it is at least exponential space by Theorem E.1 in Appendix E of the full version [18]), but rather give the first algorithm that is highly effective in practice. We work more generally with links, where a *link* is a disjoint union of knots. While a link exterior does not uniquely determine a link [3, Figure 9.28], this indeterminacy is removed by specifying meridional curves for the link; hence we require such curves as part of the input in Section 1.2. Figures 2 and 3 show diagrams that were found by our method; these are the first known diagrams of these particular link exteriors, see Section 9.1.

1.1 Prior work

The case when the interior of $E(K)$ has a complete hyperbolic structure, in short is *hyperbolic*, is in practice generic for prime knots; for example, 99.999% of the knots in [11] are hyperbolic. The homeomorphism problem for such 3-manifolds can be quickly solved in practice using hyperbolic geometry, even for triangulations with 1,000 tetrahedra [53]. This allows a table lookup method for FIND DIAGRAM when K is small enough; one uses hyperbolic and homological invariants to form a hash of $E(K)$, queries a database of knots to get a handful of possible K_i, and then checks if any $E(K_i)$ is homeomorphic to $E(K)$. This technique is used by the `identify` method of [15], but is hopeless for something like Figure 2, as the number of links of that size exceeds the number of atoms in the visible universe [47].

A related approach was used in [14, 5] to find knot diagrams for all 1,267 knots where $E(K)$ is hyperbolic and can be triangulated with at most 9 ideal tetrahedra [8, 17]. While knots with few crossing have simple exteriors, the converse is not the case, and the simplest known diagrams for about 25% of these knots have 100–300 crossings. However, these knots either fall into very special families which can be tabulated to a large number of crossings, or

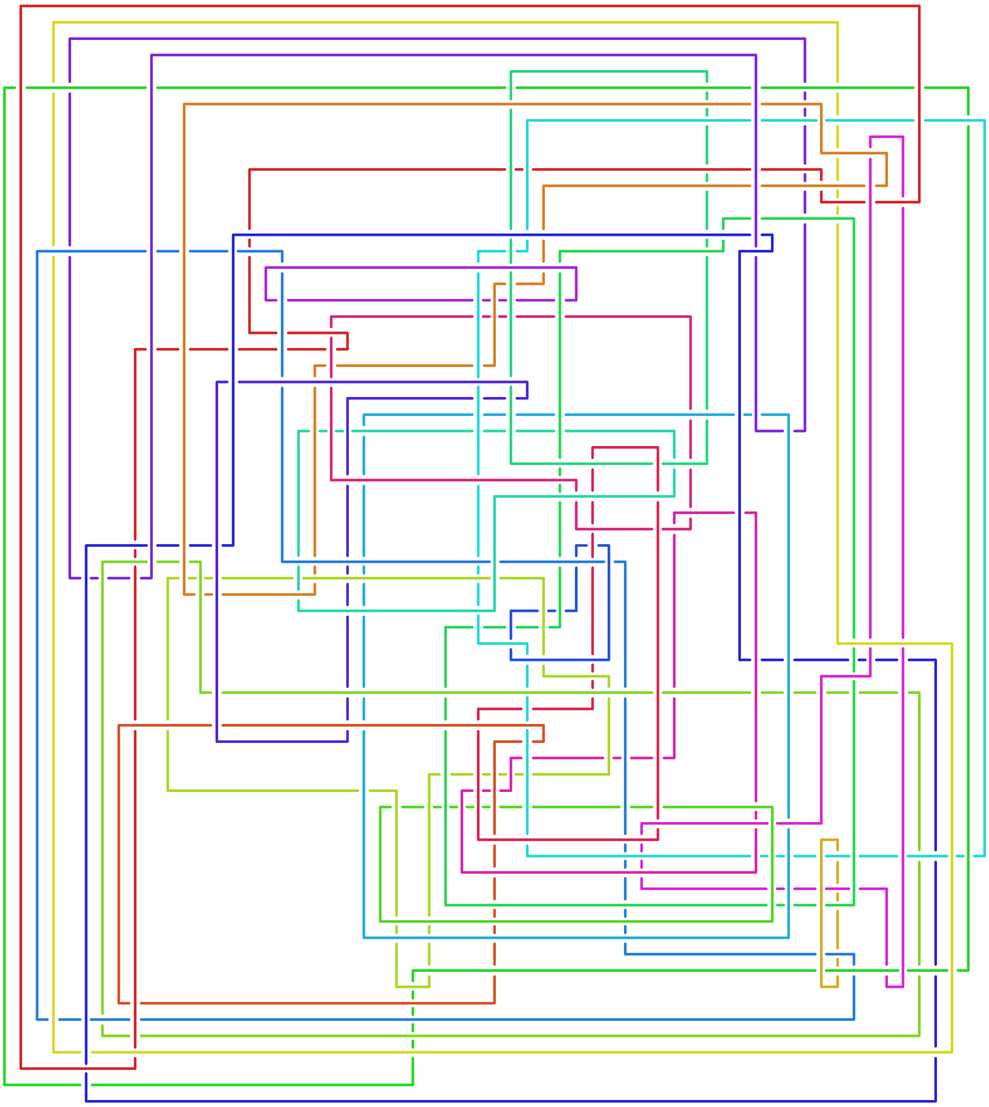


Figure 2 The first known diagram of a link whose exterior is $\mathring{M} = \mathbb{H}^3/\Gamma(I)$ where $\Gamma(I)$ is the principal congruence subgroup of $\mathrm{PSL}_2\mathbb{Z}[\frac{1+\sqrt{15}i}{2}]$ of level $I = \left\langle 6, \frac{-3+\sqrt{15}i}{2} \right\rangle$ from [6]; it has 24 components and 294 crossings. The input ideal triangulation $\mathring{\mathcal{T}}$ for $\mathring{M}$ had 249 tetrahedra. Since the hyperbolic volume of $\mathring{M} \approx 225.98$, any diagram must have at least 66 crossings by [1, Theorem 5.1].

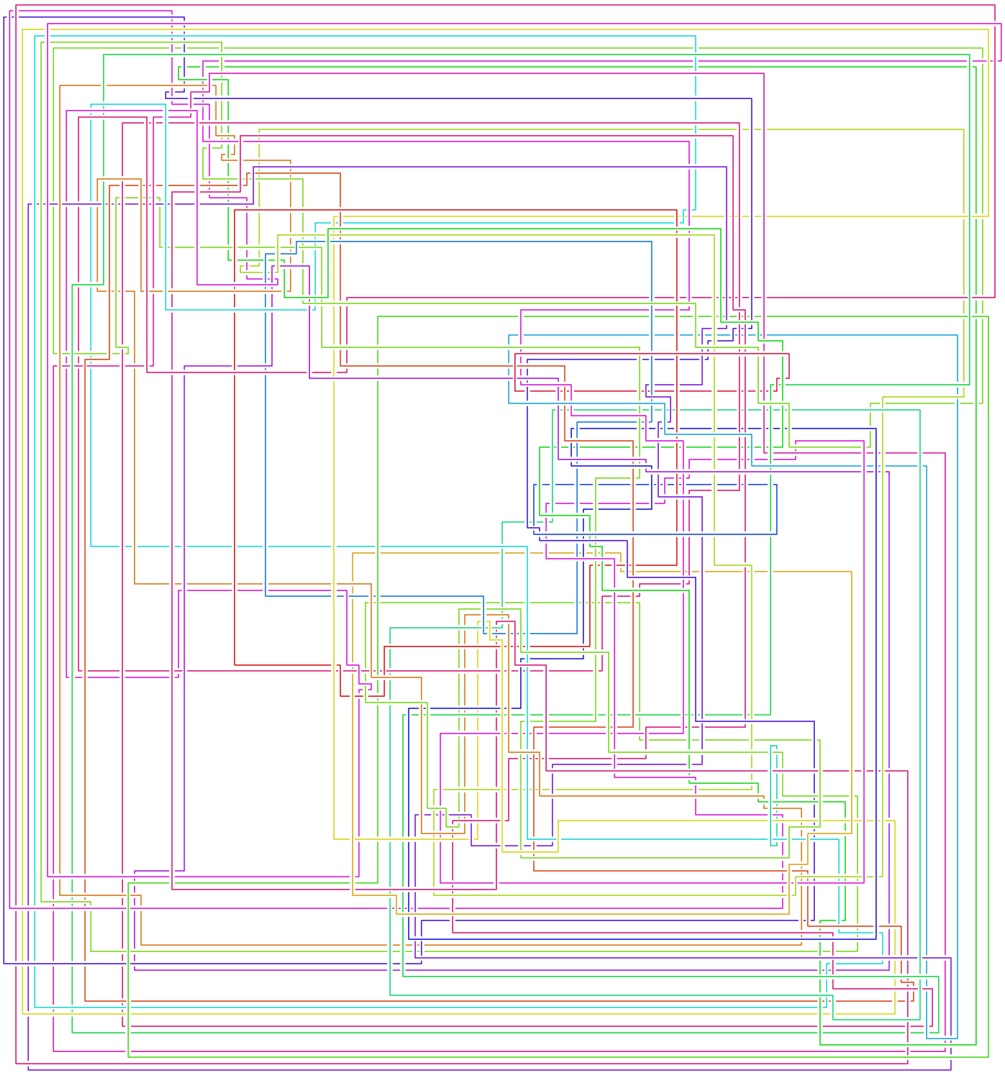


Figure 3 The first known diagram of a link whose exterior is $\mathring{M} = \mathbb{H}^3/\Gamma(I)$ where $\Gamma(I)$ is the principal congruence subgroup of $\mathrm{PSL}_2\mathbb{Z}[\frac{1+\sqrt{15}i}{2}]$ of level $I = \left\langle 5, \frac{5+\sqrt{15}i}{2} \right\rangle$ from [6]; it has 24 components and 1,092 crossings. The input ideal triangulation $\mathring{\mathcal{T}}$ for $\mathring{M}$ had 211 tetrahedra. Since the hyperbolic volume of $\mathring{M} \approx 188.32$, any diagram must have at least 56 crossings [1, Theorem 5.1].

one can drill out additional curves to get a link exterior that appears in an existing table and has special properties allowing the recovery of a diagram of the knot itself. There are other ad hoc methods in the literature, see e.g. [7] and references therein, but this paper is the first to give a generically applicable method for Find Diagram.

1.2 Outline of algorithm

As Figures 2 and 3 show, our method can solve Find Diagram in some cases where any diagram for the link has 55 or more crossings. We also easily recover everything in Section 1.1, and more applications are given in Sections 8 and 9. Experimental mean running time was $O(1.07^n)$, see Figure 14. With the definitions of Section 2, the input for our algorithm is:

▶ Input.

a. An ideal triangulation $\mathring{\mathcal{T}}$ of a compact 3-manifold $\mathring{M}$ with toroidal boundary, with an essential simple closed curve α_i for each boundary component of $\mathring{M}$.

b. A sequence (P_i) of Pachner moves transforming the layered filling triangulation $\mathcal{T}$ of the manifold $M = \mathring{M}(\alpha_1, \ldots, \alpha_k)$ into a specific 2-tetrahedra *base triangulation* $\mathcal{T}_0$ of S^3.

You might object that (b) is effectively cheating, since no polynomial-time algorithm for finding (P_i) is known, or indeed for deciding if M is S^3. Using the estimates in [37], one can perform a naive search to find some (P_i), but the complexity of this is super-exponential. However, recognizing S^3 by finding such moves is easy in practice, see Section 7, with the length of (P_i) linear in the size of $\mathcal{T}$ as per Figure 16. The output of the algorithm is a knot diagram D, encoded as a planar graph with over/under crossing data for the vertices.

The main data structure is a triangulation $\mathcal{T}$ of S^3 with a PL link L that is disjoint from the 1-skeleton. The link L is encoded as a sequence of line segments, each contained in a single tetrahedron of $\mathcal{T}$, with endpoints recorded in barycentric coordinates. An initial pair $(\mathcal{T}, L)$ in (b) is constructed from input (a) as described in Appendix A of the full version [18]. The algorithm proceeds by performing the Pachner moves P_i from (b), keeping track of the PL arcs encoding the link L throughout using the techniques of Section 3. The result is the base triangulation enriched with PL arcs representing the link L. As detailed in Section 5, this triangulation of S^3 can be cut open along faces and embedded in $\mathbb{R}^3$, giving an embedding of the cut-open link into $\mathbb{R}^3$ as a collection of PL arcs with endpoints on the boundary of these tetrahedra. As in Figure 11, these PL arcs are then tied up using the face identifications to obtain a collection of closed PL curves that represent L. An initial link diagram D is obtained by projecting this PL link onto a plane and recording crossing information. We then apply generic simplification methods to D and output the result.

This outline turns out to be deceptively simple. Some key difficulties are:

1. Understanding what $2 \to 3$ and $3 \to 2$ Pachner moves do to the link L is fairly straightforward as these correspond to changing the triangulation of a convex polyhedron in $\mathbb{R}^3$. However, while these two moves theoretically suffice for (b), in practice one wants to use $2 \to 0$ moves as well, see Section 7, and these are much harder to deal with, as Figure 6 shows. We thus expand each $2 \to 0$ move into a (sometimes quite lengthy) sequence of $2 \to 3$ and $3 \to 2$ moves as discussed in Section 4. We give a simplified expansion for the trickiest part, the endpoint-through-endpoint move, using 6 of the basic $2 \to 3$ and $3 \to 2$ moves instead of 14.
2. The complexity of the link grows very rapidly as we do Pachner moves, resulting in enormously complicated initial diagrams. We greatly reduce this by elementary local simplifications to the link after each Pachner move, see Section 3.2.
3. Prior work on simplifying link diagrams was focused on those with 30 or fewer crossings, where random application of Reidemeister moves (plus flypes) are extremely effective. Here, we need to simplify diagrams with 10,000 or even 100,000 crossings down to something with less than 100, and such methods proved ineffective for this. Instead, we used the more global *strand pickup* method of Section 6.

2 Background

2.1 Triangulations

Let M be a compact orientable 3-manifold, possibly with boundary. A *triangulation* of M is a cell complex $\mathcal{T}$ made from finitely many tetrahedra by gluing some of their 2-dimensional faces in pairs via orientation-reversing affine maps so that the resulting space is homeomorphic to M. These triangulations are not necessarily simplicial complexes, but rather what are sometimes called semi-simplicial, pseudo-simplicial, or singular triangulations. Of particular importance are those with a single vertex, the *1-vertex triangulations.* We use $\mathcal{T}^i$ to denote the i-skeleton of $\mathcal{T}$. When M has nonempty boundary, an *ideal triangulation* of M is a cell complex $\mathcal{T}$ made out of finitely many tetrahedra by gluing *all* of their 2-dimensional faces in pairs as above so that $M \setminus \partial M$ is homeomorphic to $\mathcal{T} \setminus \mathcal{T}^0$. Put another way, the manifold M is what you get by gluing together *truncated* tetrahedra in the corresponding pattern. See [49] for background on ideal triangulations, which we use only for 3-manifolds whose boundary is a union of tori. We always include the modifier "ideal", so throughout "triangulation" means a non-ideal, also called "finite", triangulation.

2.2 Triangulations with PL curves

Consider a tetrahedron Δ in $\mathbb{R}^n$ as the convex hull of its vertices v_0, v_1, v_2, and v_3. We encode points in Δ using *barycentric coordinates*, that is, write $p \in \Delta$ as the unique convex combination $\sum_i x_i v_i$ and then represent p by the vector (x_0, x_1, x_2, x_3), where of necessity $\sum_i x_i = 1$. For a 3-manifold triangulation $\mathcal{T}$, we view each tetrahedron τ as having a fixed identification with the tetrahedron in $\mathbb{R}^4$ whose vertices are the standard basis vectors; we use this to encode points in τ by barycentric coordinates.

An oriented PL curve in $\mathcal{T}$ will be described by a sequence of such barycentric coordinates as follows. A *barycentric arc a* is an ordered pair of points (u, v) in a tetrahedron τ, representing the straight segment joining them. We write $a.\texttt{start} = u$ and $a.\texttt{end} = v$. A *barycentric curve C* is a sequence of barycentric arcs a_i such that $a_i.\texttt{end}$ and $a_{i+1}.\texttt{start}$ correspond to the same point in M under the face identifications of $\mathcal{T}$. For a barycentric curve, we define $a_i.\texttt{next} = a_{i+1}$ and $a_{i+1}.\texttt{past} = a_i$; these may not lie in the same tetrahedron. Suppose the barycentric curve C consists of N barycentric arcs. If $a_0.\texttt{start}$ and $a_N.\texttt{end}$ correspond to the same point in M, we have a *barycentric loop.* An embedded barycentric loop is a *barycentric knot.* A *barycentric link* is a finite disjoint union of such knots.

We always require that a barycentric curve C is in the following kind of general position with respect to $\mathcal{T}$. First, C is disjoint from $\mathcal{T}^1$. Second, any intersection of a constituent barycentric arc a with $\mathcal{T}^2$ is an endpoint of a. Finally, arcs do not bounce off faces of $\mathcal{T}^2$, so if an arc ends in a face, the next arc must be in the adjacent tetrahedron on the other side of that face. Throughout, we use only points whose barycentric coordinates are in $\mathbb{Q}$.

2.3 Dehn filling

Suppose $\mathring{M}$ is a compact 3-manifold whose boundary is a union of tori. Given an essential simple closed curve α_i on each boundary component T_i, the *Dehn filling* of $\mathring{M}$ along $\alpha = (\alpha_1, \ldots, \alpha_k)$ is the closed 3-manifold $\mathring{M}(\alpha)$ obtained from $\mathring{M}$ by gluing a solid torus $D^2 \times S^1$ to each T_i so that $\partial D^2 \times \{\text{point}\}$ is α_i. When $\mathring{M}$ is the exterior of a link L in S^3 and each α_i is a small meridional loop about the i-th component of L, then $\mathring{M}(\alpha)$ is just S^3. Given an ideal triangulation $\mathring{\mathcal{T}}$ of $\mathring{M}$ and Dehn filling curves α, we follow [52, 32, 33] to create a 1-vertex triangulation $\mathcal{T}$ of $\mathring{M}(\alpha)$ that we call the *layered filling triangulation*; see Appendix A of the full version [18]. A key point is that the link L consisting of the cores of the k added solid tori is a barycentric link in $\mathcal{T}$ made of just k barycentric arcs.

2.4 Pachner moves

A 3-manifold triangulation $\mathcal{T}$ can be modified by local *Pachner moves* (bistellar flips) to give a new triangulation of the same underlying manifold. Those we use are:

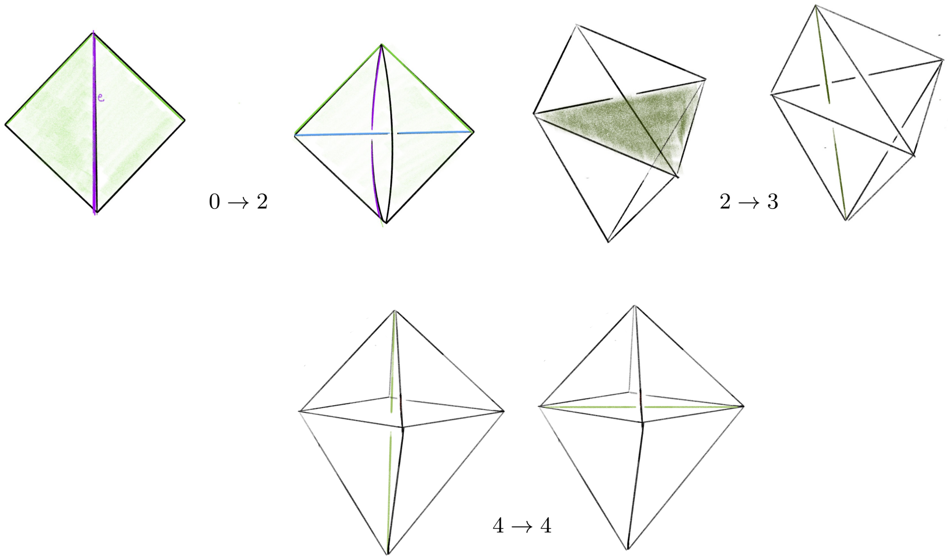


Figure 4 Pachner moves which preserve the number of vertices.

1. The 2 → 3 move and its inverse 3 → 2 move. These take a triangulation of a ball, possibly with boundary faces glued together, and retriangulate the interior without changing the boundary triangulation. Specifically, the 2 → 3 move takes a pair of distinct tetrahedra sharing a face and replaces them with three new tetrahedra around a new central edge. The 3 → 2 move reverses this, replacing three distinct tetrahedra around a valence-3 edge with two tetrahedra sharing a face.
2. The 4 → 4 move. The 4 → 4 move takes four tetrahedra around a central edge and replaces them with four new tetrahedra assembled around a new valence-4 edge.
3. The 2 → 0 move and its inverse 0 → 2 move. The 2 → 0 move takes a pair of tetrahedra sharing two faces to form a valence-2 edge and collapses them onto their common faces. The 0 → 2 move reverses this by puffing air into a pair of faces sharing an edge and adding two new tetrahedra. We call the complex created by the 0 → 2 move a *pillow*. The 0 → 2 move inflates a pillow and the 2 → 0 move collapses a pillow.

If $\mathcal{S}$ and $\mathcal{T}$ are two 1-vertex triangulations of the same closed 3-manifold M, then there is a sequence of Pachner moves that transforms $\mathcal{S}$ into $\mathcal{T}$; provided both $\mathcal{S}$ and $\mathcal{T}$ have at least two tetrahedra, one needs only use 2 → 3 and 3 → 2 moves by [36, Theorem 1.2.5] (see also [39, 42]). When M is S^3, any triangulation $\mathcal{T}$ with n tetrahedra is related to a standard triangulation by at most $12 \cdot 10^6 n^2 2^{2 \cdot 10^3 n^2}$ Pachner moves [37]. Experimentally, one needs many fewer moves [9]. In our data shown in Figure 16, the number is $O(n)$; this is essential for the utility of our algorithm for Find Diagram.

3 Modifying triangulations with arcs

Using part (a) of the input data, we first build the layered filled triangulation $\mathcal{T}$ of Section 2.3, which comes enriched with a barycentric link L. Part (b) of the input data is a sequence of Pachner moves (P_i) converting $\mathcal{T}$ to the base triangulation $\mathcal{T}_0$ of Section 5. The next step of our algorithm is to apply the moves (P_i) to $\mathcal{T}$, carrying the link L along as we go.

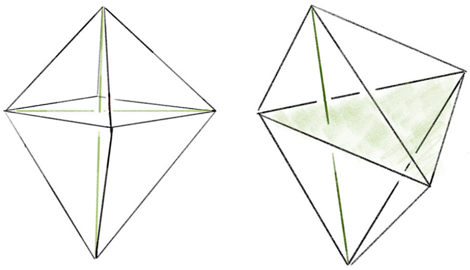


Figure 5 Two bipyramids with superimposed triangulations corresponding to before and after applying the 4 → 4 move and 2 → 3 or 3 → 2 moves.

3.1 Pachner moves with arcs

We call the 2 → 3, 3 → 2, and 4 → 4 moves the *simple Pachner moves*. Each simple Pachner move P takes a triangulated ball B in $\mathcal{T}$, possibly with boundary faces glued together, and re-triangulates B without changing the triangulation of ∂B to obtain $P\mathcal{T}$. The arcs of the link L contained in the ball are initially encoded using the barycentric coordinates of $\mathcal{T}$, and we need to re-express these arcs in the new barycentric coordinate system of $P\mathcal{T}$. We model each simple Pachner move as a pair of triangulations of concrete bipyramids in $\mathbb{R}^3$, as shown in Figure 5. We identify the tetrahedra in $\mathcal{T}$ and $P\mathcal{T}$ involved in P with tetrahedra in the corresponding bipyramid in $\mathbb{R}^3$. This identification allows us to map barycentric arcs from $\mathcal{T}$ into $\mathbb{R}^3$, and then to map these arcs in $\mathbb{R}^3$ into $P\mathcal{T}$. Appendix B of the full version [18] details how this is used to give a method `with_arcs`$[P]$ that applies a simple Pachner move P to $\mathcal{T}$ while transferring the barycentric arcs from $\mathcal{T}$ to $P\mathcal{T}$. This approach cannot work for the 2 → 0 move, as demonstrated by Figure 6. To implement `with_arcs`[2 → 0], we factor the 2 → 0 move into a sequence of 2 → 3 and 3 → 2 moves as described in Section 4.

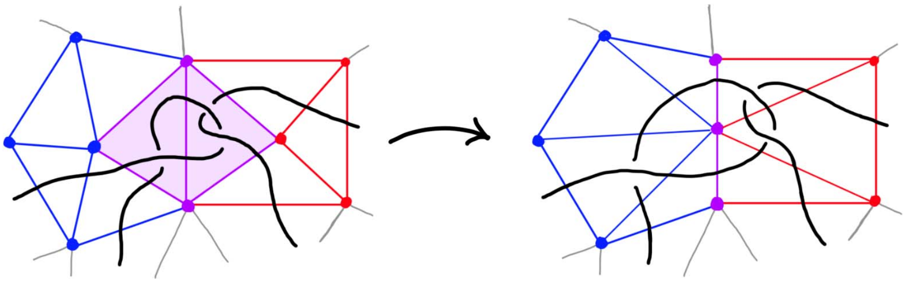


Figure 6 Cartoon showing the difficulty of doing a 2 → 0 move with arcs present. At left, the two tetrahedra in the pillow to be collapsed are shaded. Here, you should regard the vertical purple arc as the valence-2 edge, with the blue and red dots opposite being cross-sections of the two edges of the pillow that become identified in the collapse. The problem is that we have to push all the topology of the *link* out of the pillow before we collapse it, requiring us to move arcs into many of the tetrahedra adjacent to the pillow.

3.2 Simplifying arcs

Given the inputs (a) and (b) of Section 1.2, the machinery of Section 3.1 always produces the desired link L in the base triangulation $\mathcal{T}_0$. However, even in the smallest examples, applying the sequence of Pachner moves to $\mathcal{T}$ produces incredibly complicated configurations of arcs in $\mathcal{T}_0$ encoding L. This complexity makes necessary computational geometry tasks

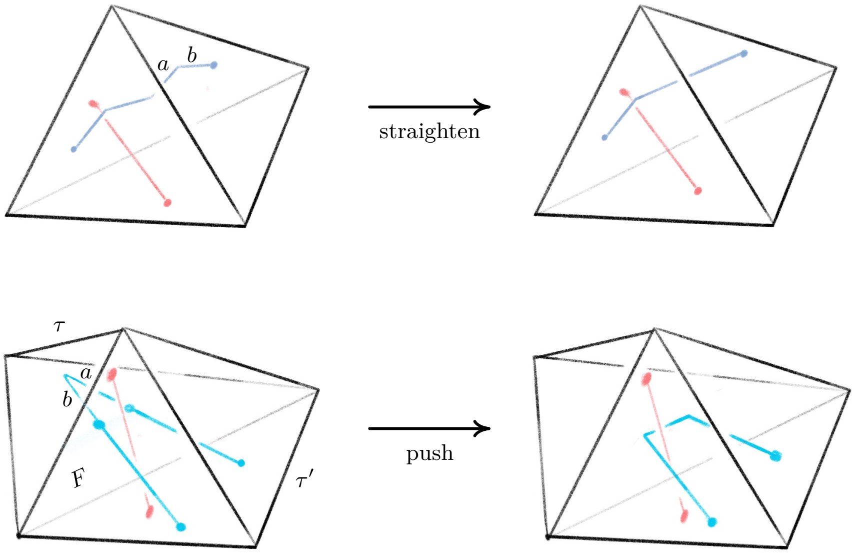


Figure 7 The *straighten* move removes unnecessary bends in the link, and the *push* move reduces unnecessary intersections with the 2-skeleton.

prohibitively expensive. Fortunately, much of this complexity is not topologically essential, and the number of arcs can be decreased dramatically by the basic simplifications we now describe. Without these, applying our full algorithm to an ideal triangulation $\mathring{\mathcal{T}}$ with just two tetrahedra resulted in 838 arcs and an initial link diagram with 5,130 crossings; with the simplifications, we get 19 arcs and 35 crossings. A 3-tetrahedra ideal triangulation resulted in 129,265 arcs compared to 27 with simplifications, and something with 10 tetrahedra would be impossible without them. Our two kinds of simplification moves are shown in Figure 7.

The first is `straighten`, which takes as input a tetrahedron τ with barycentric arcs. It then checks for each arc a in τ if the pair of arcs a and $b = a$.`next` can be replaced with a single arc that runs from a.`start` and b.`end`. The check is that no other arc in τ has an interior intersection with the triangle spanned by a and b. The other move is `push`, which removes unnecessary intersections with $\mathcal{T}^2$. When a starts on the same face F that $b = a$.`next` ends on, it checks whether any other arc intersects the triangle a and b span. If there are none, the move replaces a and b with an arc in the tetrahedron τ' glued to τ along F. This often produces a bend that can then be removed by a straighten move.

3.3 Computational geometry issues

Our algorithm requires many geometric computations with barycentric arcs, e.g. to test for one of our simplifying moves and to ensure we do not violate the general position requirement of Section 2.2. Difficult and subtle issues can arise here, and much work has been done to ameliorate them; see [44] for a survey. We took the approach of having all coordinates in $\mathbb{Q}$ so that so we can do these computations exactly. This entails a stiff speed penalty and leads to points represented by rational numbers with overwhelmingly large denominators. We handle such denominators by rounding coordinates so that the denominator is less than 2^{32}. One can certify at each step by simple local tests that this rounding does not change the isotopy type of the link. However, when the input manifold is hyperbolic, we instead certify correctness of the output diagram after the fact by checking that its exterior is homeomorphic to the manifold in part (a) of the input; this is considerably faster than checking at each step.

3.4 Putting the pieces together

Let $\mathcal{T}$ be a layered filling triangulation with arcs encoding the core curves of the filling and let (P_i) be Pachner moves reducing $\mathcal{T}$ to $\mathcal{T}_0$. Our process for producing a barycentric link in $\mathcal{T}_0$ that is isotopic to the initial L is:

Algorithm 1 `with_arcs[apply_Pachner_moves]`$\big(\mathcal{T}, (P_i)\big)$.

```
Start with $\mathcal{T}' := \mathcal{T}$ and loop over the $P_1, P_2, \ldots P_n$ as follows:
1. Apply with_arcs[$P_i$] to $\mathcal{T}'$ to get $P_i\mathcal{T}'$ with arcs representing $L$. Set $\mathcal{T}' := P_i\mathcal{T}'$.
2. Loop over the tetrahedra $\tau$ in $\mathcal{T}'$, applying push and straighten until the arcs stabilize.
```

4 Factoring the 2-to-0 move

As mentioned in Section 3.1, we factor each 2 → 0 move into a sequence of 2 → 3 and 3 → 2 moves so that we can carry along the barycentric link. This factorization is quite delicate in certain unavoidable corner cases; we outline our method in this section, but leave the details to Appendix C of the full version [18]. To begin to understand the 2 → 0 move, first consider its inverse 0 → 2 move shown in Figure 8. The possible 0 → 2 moves in Figure 8 correspond to a pillow splitting open the *book of tetrahedra* around the edge e. Following [46], we call this pillow a *bird beak* with upper and lower mandibles that pivot around the two outside edges of the beak (viewed from above, these are the purple and black vertices in the top right of Figure 8). On both sides of the bird beak are *half-books* of tetrahedra, together forming a *split-book*. When applying the inverse 2 → 0 move, the two half-books combine to form a book of tetrahedra assembled around the central edge.

The simplest 2 → 0 move is when there are two valence-2 edges that are opposite each other on a single tetrahedron, as shown in Figure 9; equivalently, one of the half-books has a single tetrahedron. This *base case* is handled by Matveev's V move, the composition of four 2 → 3 and 3 → 2 moves of [36, Figure 1.15]. To reduce other instances of the 2 → 0 move to the base case, we rotate a mandible of the bird beak, moving tetrahedra from one half-book to the other until one contains only a single tetrahedron. Because the tetrahedra in the split-book may repeat or be glued together in strange ways, this is rather delicate. When things are sufficiently embedded, Segerman [46] showed:

▶ **Proposition 4.1.** *Suppose e is a valence-2 edge where the half-books adjacent to the bird beak are embedded and contain m and n tetrahedra respectively. Then the $2 \to 0$ move can be implemented by $2 \cdot min(m, n) + 2$ basic $2 \to 3$ and $3 \to 2$ moves.*

Proof. We can rotate a mandible by one tetrahedron using the two basic moves of [46, Figure 11]. With $min(m, n) - 1$ such rotations we can reduce the smaller of the half-books to a single tetrahedron. As already noted, the base case can be done in four moves. ◀

▶ Remark 4.2. One cannot in general factor a 2 → 0 move into a sublinear number of 2 → 3 and 3 → 2 moves: the 2 → 0 move amalgamates two edges of valence $m + 1$ and $n + 1$ into a single edge of valence $m + n$, and each 2 → 3 or 3 → 2 move only changes valences by a total of 12 (counting with multiplicity).

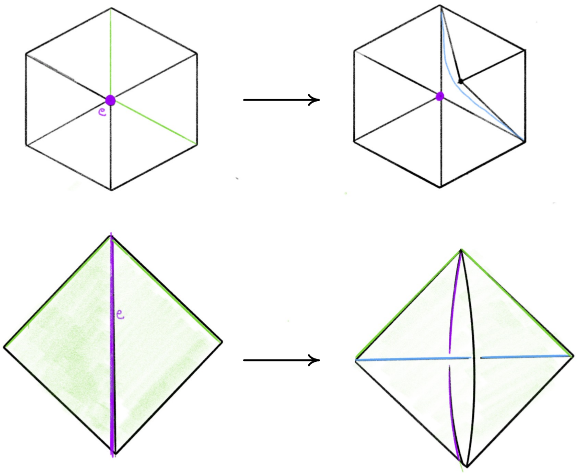
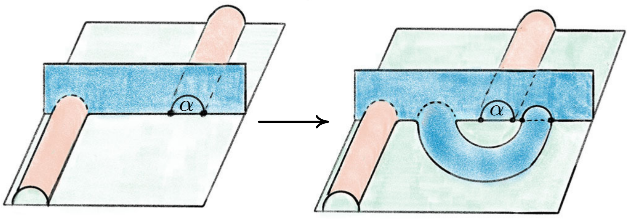


Figure 8 At top, a cross section of a $0 \to 2$ move; at bottom is a close-up of the inflation of the pillow. The move is performed on the pair of green faces meeting along the purple edge e at left. The resulting pillow is a *bird beak*, which splits open the book of tetrahedra about e. In the top right, the purple and black dots give edges that join together above and below the cross section.

4.1 Twisted beaks and endpoint-through-endpoint moves

The tricky case is when additional faces of the bird beak are glued to each other. There are two fundamentally different ways for this to happen, shown in Figures 28 and 29 of Appendix C of the full version [18]. The untwisting of these extremely confusing arrangements is done by the endpoint-through-endpoint move of Figure 10, which is in the dual language of special spines from Appendix C of [18]. Matveev's factorization of the endpoint-through-endpoint move is described in Figure 1.19 of [36]. We simplify this factorization from 14 moves to 6; the key is Proposition C.1 of Appendix C of [18] which shows that can be done with just two T moves, which is dual to two $2 \to 3$ Pachner moves. Proposition C.1 in Appendix C of the full version [18] was essential for determining the exact sequence of moves needed to factor the $2 \to 0$ move. Dual to the endpoint-through-endpoint move are a pair of *untwist the beak* moves, one for each of the situations in Figures 28 and 29, see Appendix C of [18]. We can thus factorize the $2 \to 0$ move as follows:



5 Building the initial diagram

The base triangulation $\mathcal{T}_0$ of S^3 has two tetrahedra and one vertex and is shown in Figure 11a; its isomorphism signature in the sense of [9, § 3.2], which completely determines the triangulation, is `cMcabbgdv`. We next give the method for obtaining a planar diagram D for a barycentric link L in $\mathcal{T}_0$. We first build a PL link in $\mathbb{R}^3$ representing L and then project it onto a plane to get D.

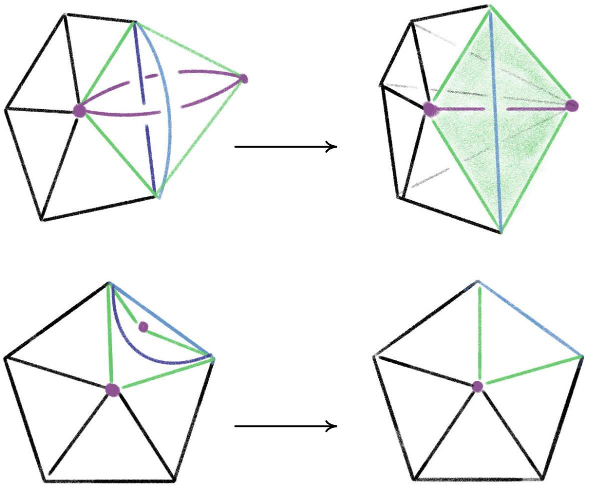


Figure 9 The base case of the 2 → 0 move at top with the cross section at bottom.

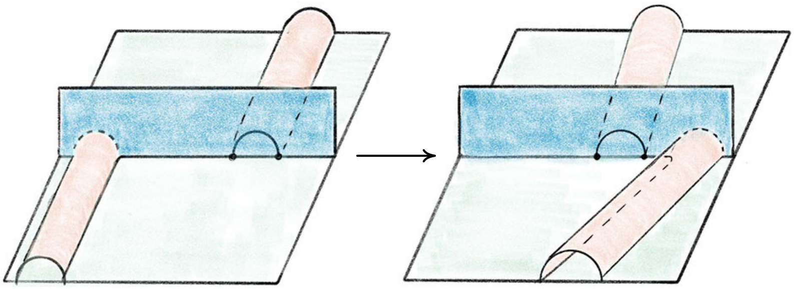
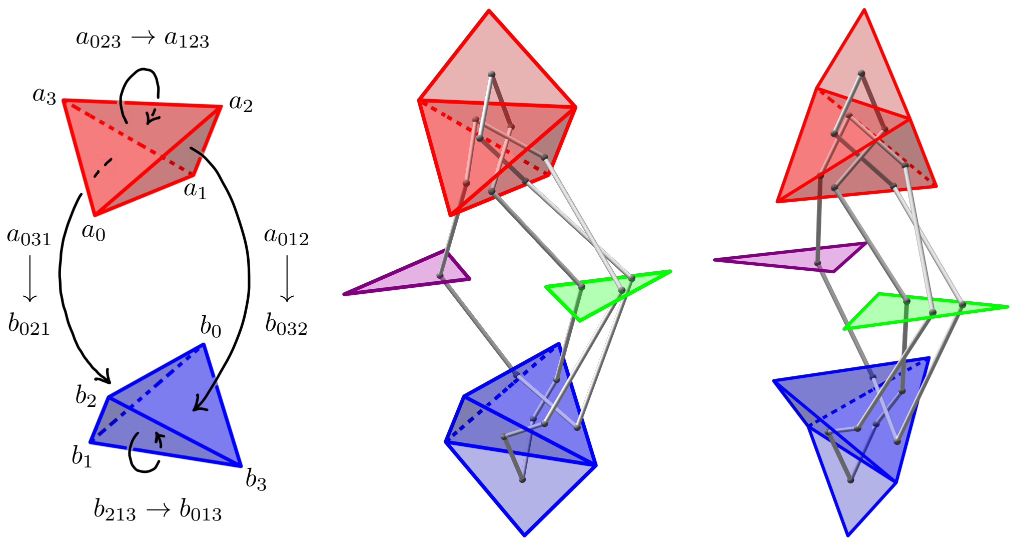


Figure 10 The *endpoint-through-endpoint* move in a special spine.

We cut open $\mathcal{T}_0$ along its faces and embed the resulting pair of tetrahedra in $\mathbb{R}^3$ as shown in Figure 11a. This cuts open the link L along its intersections with the faces of $\mathcal{T}_0$, resulting in a collection of curves in $\mathbb{R}^3$ inside the two tetrahedra. To reconnect these curves and recover L, we use *fins* and *lenses* as shown in Figure 11b to interpolate between pairs of faces that are identified in $\mathcal{T}_0$. There are two triangular fins, one attached vertically to each tetrahedron, with each fin corresponding to one of the two valence-1 edges of $\mathcal{T}_0$. The gluing of two faces incident to a valence-1 edge is realized by folding them onto the corresponding fin. Thus for each barycentric arc that ends in a face corresponding to a fin, we add the line segment joining this endpoint of the arc to the corresponding point in the fin.

Algorithm 2 `factor`[2 → 0].

1. If we are in the base case, do the sequence of moves in the triangulation dual to the factorization of the V move in Figure 1.15 of [36] and exit.
2. If we are in the twisted cases described by Figures 28 and 29 in Appendix C of the full version [18], do the appropriate untwist the beak move. Otherwise, rotate the mandible by one tetrahedron.
3. Go to to step 1.



(a) The base triangulation $\mathcal{T}_0$.

(b) Two views of the same link.

Figure 11 The base triangulation $\mathcal{T}_0$ in $\mathbb{R}^3$, with fins and lenses shown in the middle and at left.

The two triangular lenses lie between the two tetrahedra in a horizontal plane. There is an affine map taking the corresponding face in the top tetrahedron to its lens and a second affine map taking the lens to the corresponding face in the bottom tetrahedron, arranged so their composition is the face pairing in $\mathcal{T}_0$. For every arc in the top tetrahedron ending on a face corresponding to a lens, we add the line segment between the endpoint and its image under the affine map to the lens. For each such segment that terminates on a lens, we add the line segment from this endpoint to its image in the face of the bottom tetrahedron under the affine map. This results in a PL link in $\mathbb{R}^3 \subset S^3$ that must be isotopic to L: just imagine puffing out the two tetrahedra to fill all of S^3 following the guides given by the fins and lenses.

Given a collection of line segments in $\mathbb{R}^3$ corresponding to the link L, we can build a diagram for L by projecting the line segments onto a plane, computing the crossing information, and assembling this into a planar diagram. Our default choice is roughly to project onto the plane of the page in Figure 11b, with the (so far unused) fall-back of a small random matrix in $\mathrm{SL}_3\mathbb{Z}$ if a general-position failure occurs. The link diagrams resulting from this process have many more crossings than is necessary, and we deal with this in Section 6. Still, the specific configuration of fins, lenses, and projection were chosen to try minimize the number of crossings created at this stage; our initial approach used a more compact embedding where the tetrahedra shared a face, and this produced much larger diagrams.

6 Simplifying link diagrams

We now sketch how we simplified the initial link diagram constructed in Section 5, which sometimes had 10,000–100,000 crossings, to produce the final output of our algorithm for Find Diagram. Previous computational work focused on simplifying diagrams with 20 or fewer crossings [30, 11]. In that regime, random Reidemeister moves combined with flypes are extremely effective in reducing the number of crossings. However, these techniques alone proved inadequate for our much larger links. Instead, we used the more global *strand pickup* method of Figure 12. This technique was introduced by the third author and included in

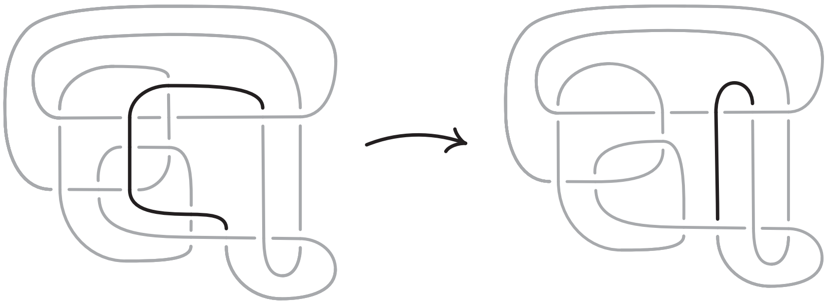


Figure 12 An example of the strand pickup method for diagram simplification. At left, an *overstrand*, which runs over each crossing it partipates in, is indicated by the darker line. At right is the result of isotoping the overstrand, fixing its endpoints, to get a diagram with fewer crossings. The best possible location for an overstrand can be found by solving a weighted shortest-path problem in the planar dual graph to the original diagram.

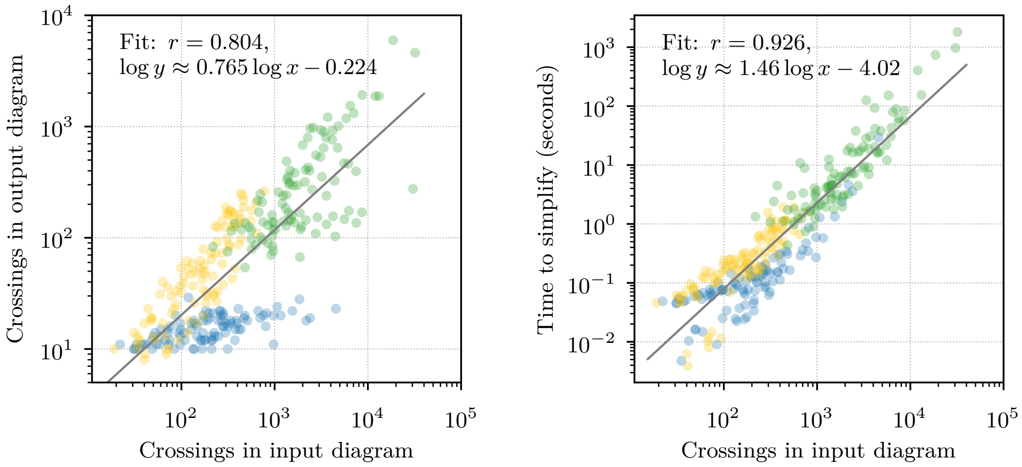


Figure 13 Simplifying 300 diagrams with between 19 and 32,095 crossings, drawn from Sections 8 and 9.3. The dramatic amount of simplification is shown at left, with an n-crossing knot turned into one with $O(n^{0.8})$ crossings. The running time at right is roughly $O(n^{1.5})$.

SnapPy [15] since version 2.3 (2015), but not previously documented in the literature. It has similarities with the arc representation/grid diagram approach of [20, 21, 22], but it works with arbitrary planar diagrams. When applying the pickup move, we start with the longest overstrands and work towards the shorter ones if no improvement is made. When a pickup move succeeds, we do more basic simplications before looking for another pickup move. We also do the same move on understrands, going back and forth between the two sides until the diagram stabilizes; for details, see [38]. The high amount of simplification and sub-quadratic running time are shown in Figure 13. As further evidence of its utility, we note that it strictly monotonically reduces the unknot diagrams D_{28}, D_{43}, and PZ_{78} in [12] to the trivial diagram; in constrast, these require adding at least three crossings if one uses only Reidemeister moves.

7 Finding certificates

Part (b) of the input to our algorithm is a certificate that the Dehn filling $M = \mathring{M}(\alpha)$ is S^3 in the form of Pachner moves simplifying a triangulation $\mathcal{T}$ of M to the base triangulation $\mathcal{T}_0$ of S^3. In practice, one starts with an ideal triangulation $\mathring{\mathcal{T}}$ and Dehn filling slopes α where it is unknown if $M(\alpha)$ is S^3. We therefore need a way of finding this sequence of Pachner moves when it exists. While deciding if a closed 3-manifold M is the S^3 is in **NP** by [31, 45] and additionally in **co-NP** assuming the Generalized Riemann Hypothesis [54, Theorem 11.2], no sub-exponential time algorithm is known. The current best algorithm for S^3 recognition is to heuristically simplify the input triangulation using Pachner moves and then apply the theory of almost normal surfaces, see Algorithm 3.2 of [10]. However, triangulations of S^3 that are truly hard to simplify using Pachner moves have not been encountered in practice, and it is open whether they exist at all [9]. Thus, when M is S^3, the initial stage of Algorithm 3.2 of [10] nearly always arrives at a 1-tetrahedron triangulation of S^3 and no normal surface theory is needed. The usefulness of our algorithm for Find Diagram relies on the fact that a heuristic search using Pachner moves gives a practical recognition algorithm for S^3.

▶ Remark 7.1. The effectiveness of our heuristic search procedure relies on the $2 \to 0$ move being atomic. Initially, we tried restricting our heuristic search to just the simple Pachner moves, but were typically unable to find a sequence that simplified the input triangulation of S^3 down to one with just a few tetrahedra. (To square this with [9], note from Figure 16 that our triangulations are much larger.) As is clear from Appendix C of the full version [18], factoring the $2 \to 0$ move as a sequence of $2 \to 3$ and $3 \to 2$ moves is complicated enough that one cannot expect to stumble upon these sequences when the triangulation is large and the search is restricted to simple Pachner moves.

Our simplification heuristic closely follows that of SnapPy [15], with some modifications that reduce the complexity of the final barycentric link in $\mathcal{T}_0$. These include:

1. Simplifying the layered filling triangulation $\mathcal{T}$ of Section 2.3 as much as possible without modifying the few tetrahedra containing the initial link.
2. Finding sequences of Pachner moves to $\mathcal{T}_0$ for several different layered filling triangulations, and then using the one requiring the fewest moves for the computations in Sections 3–6.
3. Ensuring the tail of the sequence of moves is a geodesic in the Pachner graph of [9].

The details are in Appendix D of the full version [18].

8 Implementation and initial experiments

We implemented our algorithm in Python, building on the pure-Python `t3mlite` library for 3-manifold triangulations that is part of SnapPy [15]. We also used SnapPy's C kernel to produce the layered filled triangulation $\mathcal{T}$ of Section 2.3 from the input ideal triangulation $\mathring{\mathcal{T}}$. The needed linear algebra over $\mathbb{Q}$ was handed by PARI [48]. Not including these libraries, our implementation consists of 1,800 lines of Python code. We had to put some effort into optimization to handle things as large as Figure 3, but more could be done. Our code and data is archived at [19] and the code will be incorporated into version 3.1 of SnapPy [15].

To validate our implementation, we applied it to two samples, one where the input is small and one where the best-possible output is small. The first, $\mathcal{CK}$, is the 1,267 hyperbolic knots whose exteriors have ideal triangulations with at most 9 tetrahedra [17, 5]. The second, $\mathcal{SK}$, consists of 1,000 knots whose minimal crossing number was between 10 and 19. There are 100 knots for each crossing number, which were selected at random from all the hyperbolic nonalternating knots with that crossing number [11]; the exception is that there are only 41 such 10-crossing knots, so 59 alternating 10-crossing knots were used as well. (Alternating knots have unusually close connections between their diagrams and exteriors, so were excluded as possibly being an easy case for Find Diagram.)

Our program found diagrams for all 2,267 of these exteriors. The running time was under 20 seconds for 96.7% of them, with a max of 2.5 minutes (CPUs were Intel Xeon E5-2690 v3 at 2.6GHz with 4G of memory per core, circa 2014); see Figure 14. The input ideal triangulations $\mathring{\mathcal{T}}$ had between 2 and 44 tetrahedra, and the resulting layered filling triangulation $\mathcal{T}$ had between 13 and 77 tetrahedra (mean of 31.5), typically 60% larger than $\mathring{\mathcal{T}}$; see Figure 15. The sequence of simple Pachner moves used to reduce $\mathcal{T}$ to $\mathcal{T}_0$ had length between 39 and 761 (mean of 241.0), see Figure 16; this was typically 7.5 times longer than the initial sequence of Pachner moves that included $2 \to 0$ moves (Figure 17). For the knots in $\mathcal{SK}$, we compare the size of the output diagram to the minimal crossing number in Figure 18; the output matched the crossing number for 42.1% of these exteriors, and it was within 3 for 87.8%. For $\mathcal{CK}$, the number of crossings in the output had max 303, mean 65.9, and median 40.

9 Applications

9.1 Congruence links

Powerful tools from number theory apply to the special class of arithmetic hyperbolic 3-manifolds. Thurston asked which link exteriors are in the subclass of principal congruence arithmetic manifolds; this was resolved in [6]: there are exactly 48 such exteriors. These 48 have hyperbolic volumes in [5.33348, 1365.37] and ideal triangulations with between 6 and 1,526 tetrahedra. Link diagrams for 15 of these 48 had previously been found by ad hoc methods [7]. Our program has found diagrams for 23 more, including Figures 2 and 3; collectively, we now have links for the 38 such exteriors of smallest volume, see Figure 20.

9.2 Dehn surgery descriptions

Every closed orientable 3-manifold is Dehn filling on some link exterior in S^3 [43, Chapter 9], and such *Dehn surgery descriptions* play a key role in both theory and practice. However, finding a Dehn surgery description from e.g. a triangulation can be extremely challenging. Thurston observed experimentally that, starting with a closed hyperbolic 3-manifold, one frequently arrives at a link exterior by repeatedly drilling out short closed geodesics, see page 516 of [2]. Combining this with our algorithm for Find Diagram gives an effective tool for finding Dehn surgery descriptions given a triangulation. We applied this to the Seifert–Weber dodecahedral space, which is an old example [50] still of much current interest [13, 34]. The resulting description in Figure 21 seems to be the first such published; a different description appeared subsequently in [4].

9.3 Knots with the same 0-surgery

The 0-surgery $Z(K)$ on a knot K is the unique Dehn filling N of $E(K)$ where $H_1(N;\mathbb{Q}) \neq 0$. Pairs of knots K and K' with $Z(K)$ homeomorphic to $Z(K')$ are of much interest in low-dimensional topology. Most strikingly, if such a pair K and K' exist with K slice (i.e. bounds a smooth D^2 in D^4) and the Rasmussen s-invariant of K' is nonzero, then the smooth 4-dimensional Poincaré conjecture is false. That is, there would exist a 4-manifold that is homeomorphic but not diffeomorphic to S^4. See [25, 35] for a general discussion, and also [41] for an important recent result using pairs with $Z(K) \cong Z(K')$. There are many techniques for constructing families of such pairs, which have been unified by the red-blue-green link

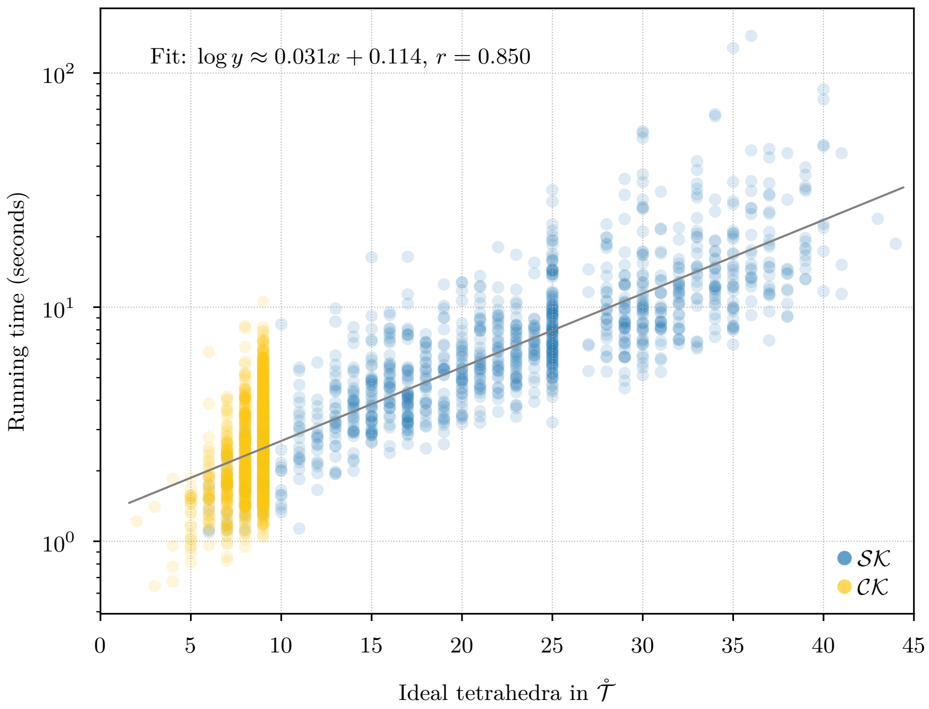


Figure 14 Mean running time for the 2,267 knot exteriors in $\mathcal{SK}$ and $\mathcal{CK}$ appears exponential with small base, roughly $O(1.07^n)$. Compare Figure 19 on the growth of the number of arcs in $\mathcal{T}_0$.

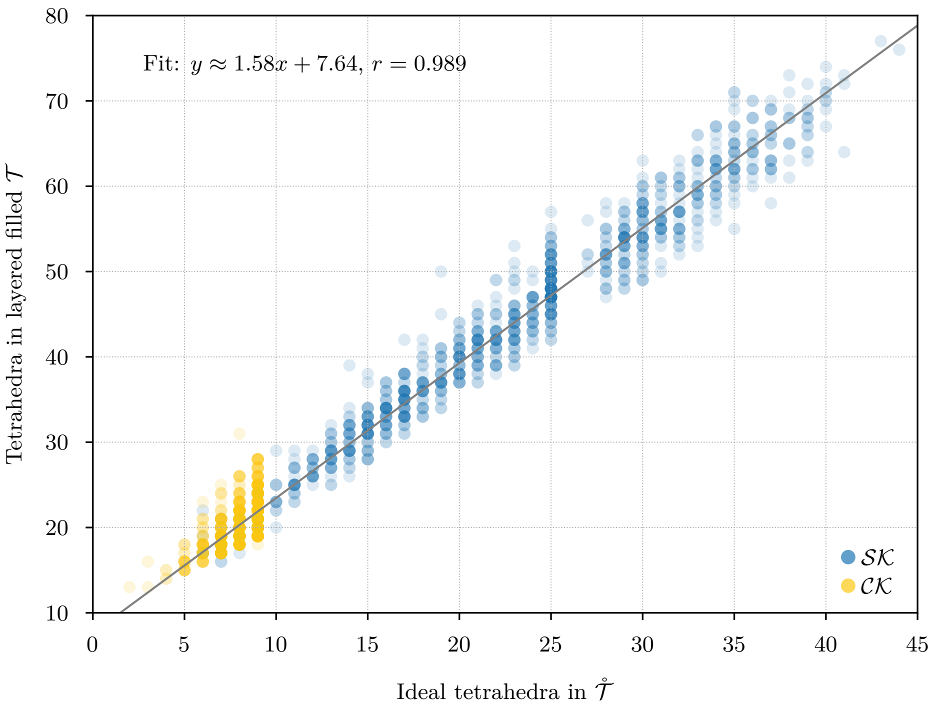


Figure 15 The number of tetrahedra in the layered filled $\mathcal{T}$ compared to the input ideal $\mathring{\mathcal{T}}$.

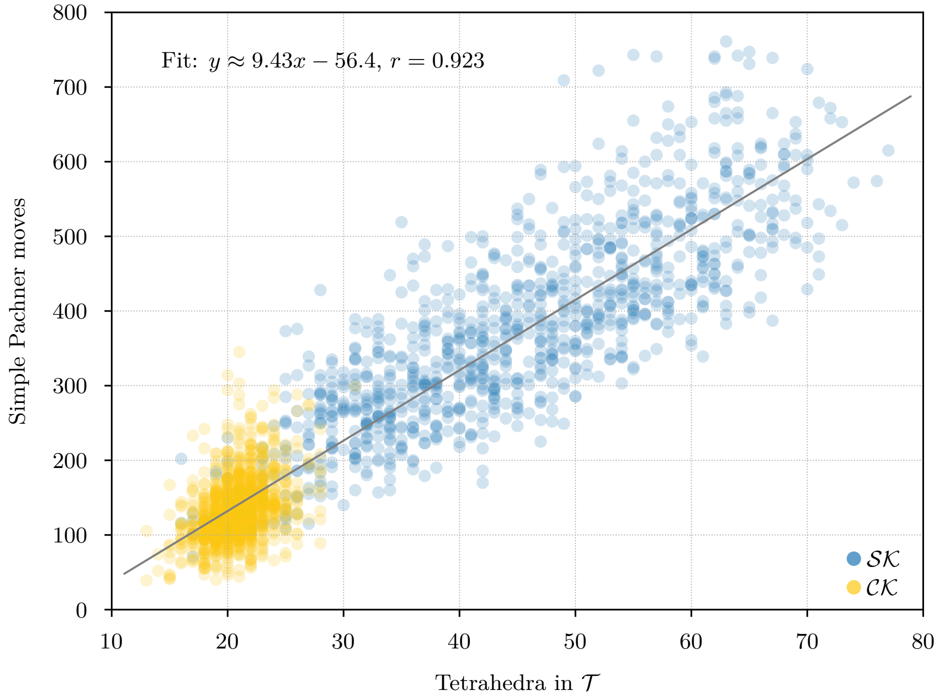


Figure 16 The number of *simple* Pachner moves used to transform the layered filled triangulation $\mathcal{T}$ into the base triangulation $\mathcal{T}_0$ is generically linear in the size of $\mathcal{T}$.

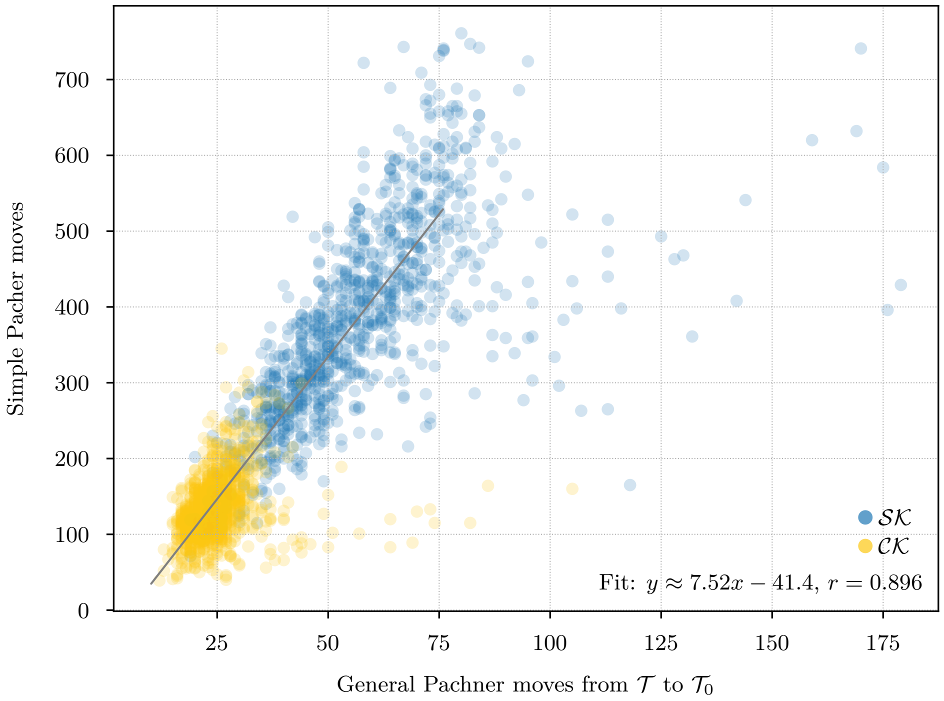
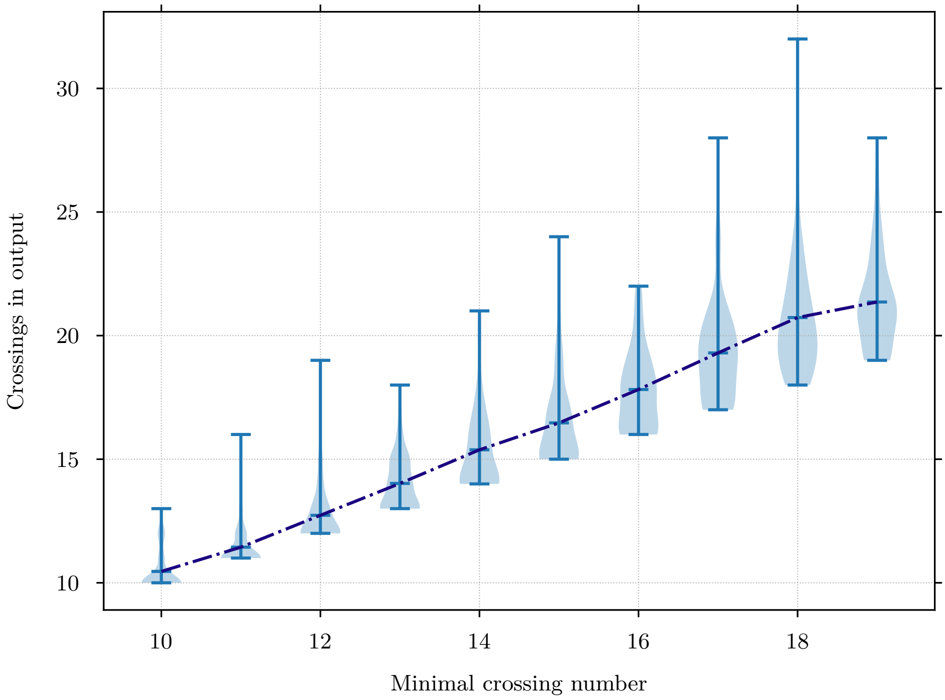
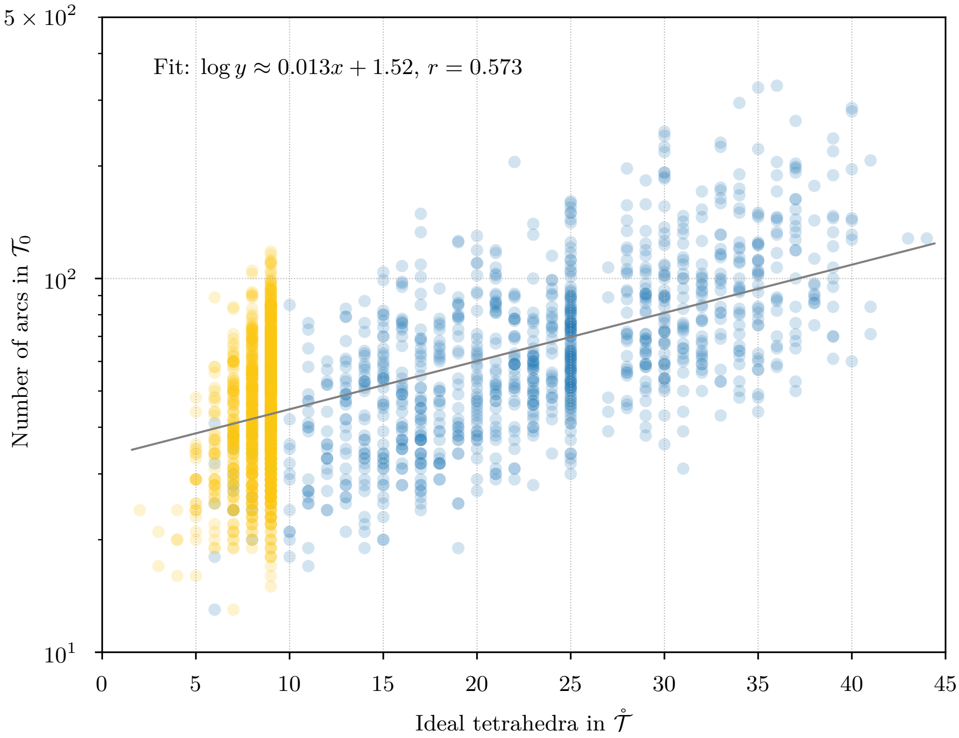


Figure 17 This plot shows the increase in the number of Pachner moves when we factor the $2 \to 0$ moves into simple Pachner moves. The regression line is based on points with $x < 75$.



🟨 **Figure 18** For the knots in $\mathcal{SK}$, grouped by minimum crossing number, the number of crossings in the diagram output by our program. The dotted line indicates the mean.



🟨 **Figure 19** The number of barycentric arcs when we arrive at $\mathcal{T}_0$ appears exponential in the size of the input $\mathring{\mathcal{T}}$, roughly $O(1.03^n)$.

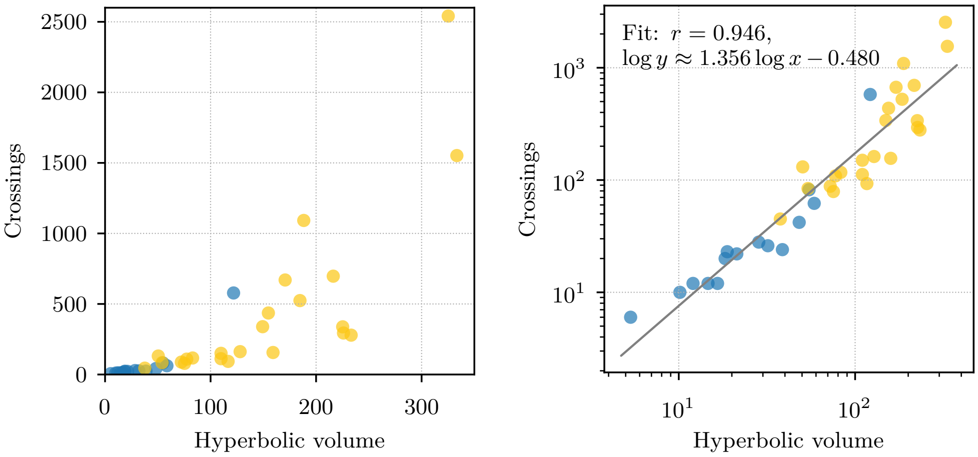


Figure 20 The 38 known link diagrams whose exteriors are principal congruence arithmetic; blue are the 15 from [7], yellow are new. The plots are the same save for the scales on the axes. The regression at right predicts that a link for the largest such exterior would have 9,000 crossings.

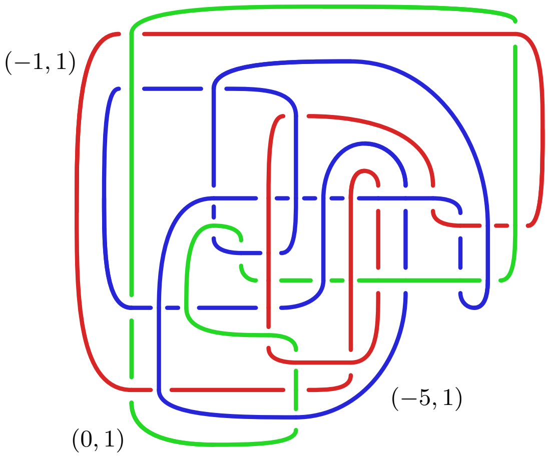


Figure 21 A Dehn surgery description of the Seifert–Weber dodecahedral space.

framework of [35]. However, given a particular K, a practical algorithm to search for K' with the same 0-surgery has been lacking. When $Z(K)$ is hyperbolic, we attack this as follows. First, find the short closed geodesics in $Z(K)$ using [29]. Then drill out each geodesic in turn, and test if the resulting manifold $\mathring{M}'$ has a Dehn filling which is S^3; if it does, use our algorithm for Find Diagram to $\mathring{M}'$ to get a diagram for K'.

Figure 22 shows the result of applying our algorithm to 100 pairs (K, γ) where K is a knot with at most 18 crossings and γ is a short closed geodesic in $Z(K)$ whose exterior is also that of a knot K' in S^3. In all cases, we were able to recover a diagram for K', and these were more challenging on average than the examples in Section 8.

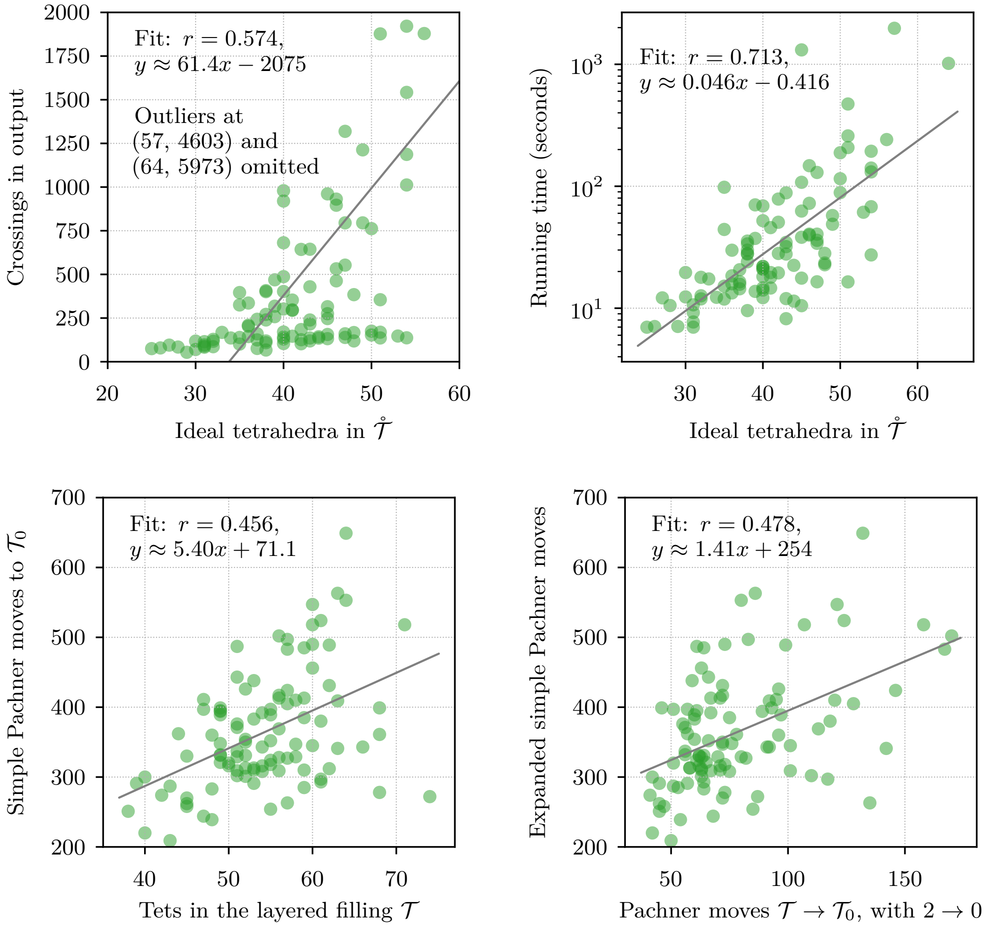


Figure 22 Data on the 100 knot exteriors from Section 9.3.

10 Future work

Having demonstrated the practicality of solving FIND DIAGRAM, we plan to refine our implementation and then incorporate it as a standard feature of SnapPy [15] so that it can be widely used. In particular, we aim to:

1. Explore whether the mean running time of $O(1.07^n)$ can be reduced. While Theorem E.1 in Appendix E of the full version [18] shows that the worst case running time must be at least exponential, it is not implausible that the mean running time is polynomial in the size of the *output*. The key issue is that the number of arcs in $\mathcal{T}_0$ is currently exponential in the size of both the input and the output, compare Figure 19.
2. To reduce the number of arcs, we could consider additional local PL simplification moves, or try the current moves in larger balls in $\mathcal{T}$ made up of several tetrahedra.
3. Explore whether modern methods in computational geometry can be used to speed up the work in Sections 3 and 5.

References

1 Colin Adams. Triple crossing number of knots and links. *J. Knot Theory Ramifications*, 22(2):1350006, 17, 2013. doi:10.1142/S0218216513500065.

2 Colin C. Adams. Isometric cusps in hyperbolic 3-manifolds. *Michigan Math. J.*, 46(3):515–531, 1999. doi:10.1307/mmj/1030132477.

3 Colin Conrad. Adams. *The knot book : an elementary introduction to the mathematical theory of knots*. W.H. Freeman, 1994.

4 Kenneth L. Baker. A sketchy surgery description of the seifert-weber dodecahedral space, 2021. URL: https://sketchesoftopology.wordpress.com/2021/12/09/a-sketchy-surgery.

5 Kenneth L. Baker and Marc Kegel. Census L-space knots are braid positive, except one that is not, in preparation.

6 M. D. Baker, M. Goerner, and A. W. Reid. All principal congruence link groups. *J. Algebra*, 528:497–504, 2019. doi:10.1016/j.jalgebra.2019.02.023.

7 Mark D. Baker, Matthias Goerner, and Alan W. Reid. All known principal congruence links. Preprint 2019, 9 pages. arXiv:1902.04426.

8 Benjamin A. Burton. The cusped hyperbolic census is complete. Preprint 2014, 32 pages. arXiv:1405.2695.

9 Benjamin A. Burton. The Pachner graph and the simplification of 3-sphere triangulations. In *Computational geometry (SCG'11)*, pages 153–162. ACM, New York, 2011. doi:10.1145/1998196.1998220.

10 Benjamin A. Burton. Computational topology with Regina: algorithms, heuristics and implementations. In *Geometry and topology down under*, volume 597 of *Contemp. Math.*, pages 195–224. Amer. Math. Soc., Providence, RI, 2013. doi:10.1090/conm/597/11877.

11 Benjamin A. Burton. The next 350 million knots. In *36th International Symposium on Computational Geometry*, volume 164 of *LIPIcs. Leibniz Int. Proc. Inform.*, pages Art. No. 25, 17. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2020. doi:10.4230/LIPIcs.SoCG.2020.25.

12 Benjamin A. Burton, Hsien-Chih Chang, Maarten Löffler, Arnaud de Mesmay, Clément Maria, Saul Schleimer, Eric Sedgwick, and Jonathan Spreer. Hard diagrams of the unknot. Preprint 2021, 26 pages. arXiv:2104.14076.

13 Benjamin A. Burton, J. Hyam Rubinstein, and Stephan Tillmann. The Weber-Seifert dodecahedral space is non-Haken. *Trans. Amer. Math. Soc.*, 364(2):911–932, 2012. doi:10.1090/S0002-9947-2011-05419-X.

14 Abhijit Champanerkar, Ilya Kofman, and Timothy Mullen. The 500 simplest hyperbolic knots. *J. Knot Theory Ramifications*, 23(12):1450055, 34, 2014. doi:10.1142/S0218216514500552.

15 Marc Culler, Nathan M. Dunfield, Matthias Goerner, and Jeffrey R. Weeks. SnapPy, a computer program for studying the geometry and topology of 3-manifolds, version 3.0.2, 2021. URL: https://snappy.computop.org.

16 Arnaud de Mesmay, Yo'av Rieck, Eric Sedgwick, and Martin Tancer. The unbearable hardness of unknotting. *Adv. Math.*, 381:Paper No. 107648, 36, 2021. doi:10.1016/j.aim.2021.107648.

17 Nathan M. Dunfield. A census of exceptional Dehn fillings. In *Characters in low-dimensional topology*, volume 760 of *Contemp. Math.*, pages 143–155. Amer. Math. Soc., [Providence], RI, 2020. doi:10.1090/conm/760/15289.

18 Nathan M. Dunfield, Malik Obeidin, and Cameron Gates Rudd. Computing a Link Diagram from its Exterior, 2021. Full version of this paper, 34 pages. arXiv:2112.03251v2.

19 Nathan M. Dunfield, Malik Obeidin, and Cameron Gates Rudd. Code and data for computing a link diagram from its exterior, 2022. doi:10.7910/DVN/BT1M8R.

20 I. A. Dynnikov. Three-page approach to knot theory. Coding and local motions. *Funktsional. Anal. i Prilozhen.*, 33(4):25–37, 96, 1999. doi:10.1007/BF02467109.

21 I. A. Dynnikov. Arc-presentations of links: monotonic simplification. *Fund. Math.*, 190:29–76, 2006. doi:10.4064/fm190-0-3.

22 Ivan Dynnikov and Vera Sokolova. Multiflypes of rectangular diagrams of links. *J. Knot Theory Ramifications*, 30(6):Paper No. 2150038, 15, 2021. doi:10.1142/S0218216521500383.

23 Erica Flapan. *When topology meets chemistry*. Outlooks. Cambridge University Press, Cambridge; Mathematical Association of America, Washington, DC, 2000. A topological look at molecular chirality. doi:10.1017/CBO9780511626272.

24 Erica Flapan, Adam He, and Helen Wong. Topological descriptions of protein folding. *Proc. Natl. Acad. Sci. USA*, 116(19):9360–9369, 2019. doi:10.1073/pnas.1808312116.

25 Michael Freedman, Robert Gompf, Scott Morrison, and Kevin Walker. Man and machine thinking about the smooth 4-dimensional Poincaré conjecture. *Quantum Topol.*, 1(2):171–208, 2010. doi:10.4171/QT/5.

26 C. McA. Gordon and J. Luecke. Knots are determined by their complements. *J. Amer. Math. Soc.*, 2(2):371–415, 1989. doi:10.2307/1990979.

27 Wolfgang Haken. Theorie der Normalflächen. *Acta Math.*, 105:245–375, 1961. doi:10.1007/BF02559591.

28 Joel Hass, Jeffrey C. Lagarias, and Nicholas Pippenger. The computational complexity of knot and link problems. *J. ACM*, 46(2):185–211, 1999. doi:10.1145/301970.301971.

29 Craig D. Hodgson and Jeffrey R. Weeks. Symmetries, isometries and length spectra of closed hyperbolic three-manifolds. *Experiment. Math.*, 3(4):261–274, 1994.

30 Jim Hoste, Morwen Thistlethwaite, and Jeff Weeks. The first 1,701,936 knots. *Math. Intelligencer*, 20(4):33–48, 1998. doi:10.1007/BF03025227.

31 S. V. Ivanov. The computational complexity of basic decision problems in 3-dimensional topology. *Geom. Dedicata*, 131:1–26, 2008. doi:10.1007/s10711-007-9210-4.

32 William Jaco and J. Hyam Rubinstein. Inflations of ideal triangulations. *Adv. Math.*, 267:176–224, 2014. doi:10.1016/j.aim.2014.09.001.

33 William Jaco and Eric Sedgwick. Decision problems in the space of Dehn fillings. *Topology*, 42(4):845–906, 2003. doi:10.1016/S0040-9383(02)00083-6.

34 Francesco Lin and Michael Lipnowski. Monopole Floer Homology, Eigenform Multiplicities and the Seifert-Weber Ddodecahedral Space. *Int. Math. Res. Notices*, to appear. doi:10.1093/imrn/rnaa310.

35 Ciprian Manolescu and Lisa Piccirillo. From zero surgeries to candidates for exotic definite four-manifolds. Preprint 2021, 30 pages. arXiv:2102.04391.

36 Sergei Matveev. *Algorithmic topology and classification of 3-manifolds*, volume 9 of *Algorithms and Computation in Mathematics*. Springer, Berlin, second edition, 2007.

37 Aleksandar Mijatović. Simplifying triangulations of S^3. *Pacific J. Math.*, 208(2):291–324, 2003. doi:10.2140/pjm.2003.208.291.

38 Malik Obeidin. Link simplification code for Spherogram. URL: https://github.com/3-manifolds/Spherogram/blob/master/spherogram_src/links/simplify.py.

39 Udo Pachner. P.L. homeomorphic manifolds are equivalent by elementary shellings. *European J. Combin.*, 12(2):129–145, 1991. doi:10.1016/S0195-6698(13)80080-7.

40 Satya R. T. Peddada, Nathan M. Dunfield, Lawrence E. Zeidner, Kai A. James, and James T. Allison. Systematic Enumeration and Identification of Unique Spatial Topologies of 3D Systems Using Spatial Graph Representations. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, volume 3A: 47th Design Automation Conference (DAC), 2021. doi:10.1115/DETC2021-66900.

41 Lisa Piccirillo. The Conway knot is not slice. *Ann. of Math. (2)*, 191(2):581–591, 2020. doi:10.4007/annals.2020.191.2.5.

42 Riccardo Piergallini. Standard moves for standard polyhedra and spines. *Rend. Circ. Mat. Palermo (2) Suppl.*, 18:391–414, 1988. Third National Conference on Topology (Italian) (Trieste, 1986).

43 Dale Rolfsen. *Knots and links*, volume 7 of *Mathematics Lecture Series*. Publish or Perish, Inc., Houston, TX, 1990. Corrected reprint of the 1976 original.

44 Stefan Schirra. Robustness and precision issues in geometric computation. In *Handbook of computational geometry*, pages 597–632. North-Holland, Amsterdam, 2000. `doi:10.1016/B978-044482537-7/50015-2`.

45 Saul Schleimer. Sphere recognition lies in NP. In *Low-dimensional and symplectic topology*, volume 82 of *Proc. Sympos. Pure Math.*, pages 183–213. Amer. Math. Soc., Providence, RI, 2011. `doi:10.1090/pspum/082/2768660`.

46 Henry Segerman. Connectivity of triangulations without degree one edges under 2-3 and 3-2 moves. *Proc. Amer. Math. Soc.*, 145(12):5391–5404, 2017. `doi:10.1090/proc/13485`.

47 Carl Sundberg and Morwen Thistlethwaite. The rate of growth of the number of prime alternating links and tangles. *Pacific J. Math.*, 182(2):329–358, 1998. `doi:10.2140/pjm.1998.182.329`.

48 The PARI Group, Univ. Bordeaux. *PARI/GP version 2.11.4*, 2020. URL: `http://pari.math.u-bordeaux.fr`.

49 Stephan Tillmann. Normal surfaces in topologically finite 3-manifolds. *Enseign. Math. (2)*, 54(3-4):329–380, 2008. `arXiv:math/0406271`.

50 C. Weber and H. Seifert. Die beiden Dodekaederräume. *Math. Z.*, 37(1):237–253, 1933. `doi:10.1007/BF01474572`.

51 Jeff Weeks. Computation of hyperbolic structures in knot theory. In *Handbook of knot theory*, pages 461–480. Elsevier B. V., Amsterdam, 2005. `doi:10.1016/B978-044451452-3/50011-3`.

52 Jeffery R. Weeks. Source code file `close_cusp.c` for SnapPea, version 2.5, circa 1995. URL: `https://github.com/3-manifolds/SnapPy/blob/master/kernel/kernel_code/`.

53 Jeffrey R. Weeks. Convex hulls and isometries of cusped hyperbolic 3-manifolds. *Topology Appl.*, 52(2):127–149, 1993. `doi:10.1016/0166-8641(93)90032-9`.

54 Raphael Zentner. Integer homology 3-spheres admit irreducible representations in $\mathrm{SL}(2,\mathbb{C})$. *Duke Math. J.*, 167(9):1643–1712, 2018. `doi:10.1215/00127094-2018-0004`.