Knots, Links, and the Search for Exotic 4-spheres

Julius Zhang

Advisor: Ciprian Manolescu

August 2021

Abstract

This paper presents a method that would potentially discover an exotic 4-sphere by constructing RBG links.

Contents

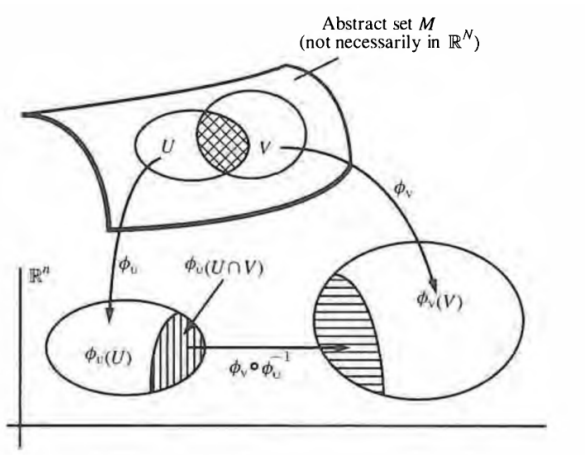
1 Introduction

The smooth Poincaré conjecture in dimesion four asks whether there exists a manifold that is homeomorphic to S^4 with its standard topology but not diffeomorphic to it. By Freedman's theorem [6], any homotopy 4-sphere is is necessarily diffeomorphic to S^4. Therefore, the smooth Poincaré conjecture in dimension 4 can be formulated as whether any homotopy 4-sphere is diffeomorphic to S^4. A counter example would be referred to as an exotic 4-sphere.

This paper presents a strategy that would potentially lead to the discovery of an exotic 4-sphere. We would present relevant backgrounds in topology and knot theory. Two computer programs used in our research are included in the appendix.

2 Topology background

Intuitively, a manifold is a topological space that looks locally like an Euclidean space (in local coordinate charts). A smooth manifold is a manifold where the transition functions between local charts are smooth. For a topological manifold, the transition functions are assumed to be only continuous.



3 The smooth Poincaré conjecture

By an exotic sphere $\mathbb{S}^n$, we mean a manifold homeomorphic to $\mathbb{S}^n$ with its usual topology but not diffeomorphic to it. The existence of an exotic sphere is unknown for $n = 4$. (This is known as 4D smooth Poincaré conjecture.) For $n = 1, 2, 3, 5, 6$, there is no such an exotic sphere. There exists exotic spheres for $n = 7$ and for some higher n. In fact, Milnor [5] constructed exotic 7-spheres as S^3-bundles over S^4, and he showed that there are at least seven different smooth structures on S^7.

4 Knot Theory

4.1 Basic definitions

We now define what we mean by a knot and a slice knot.

Definition 1 (Knot) *A knot is an embedding of* $\mathbb{S}^1$ *in* $\mathbb{S}^3$.

Definition 2 (Link) *A link is a finite union of disjoint knots.*

Definition 3 (Slice knot) *A slice knot is a knot that bounds a disk* $\mathbb{D}^2$ *in* $\mathbb{D}^4$, *where* $\mathbb{S}^3$ *is viewed as the boundary of* $\mathbb{D}^4$ *and* $\mathbb{D}^2$ *is smoothly embedded.*

Definition 4 (Ribbon knot) *A knot is a ribbon knot if it bounds a disk* $\mathbb{D}^2$ *and its intersections with itself are arcs lie in the interior of* $\mathbb{D}^2$.

By moving in the fourth dimension, we can remove the intersections. Therefore [6],

Theorem 1 *All the ribbon knots are slice.*

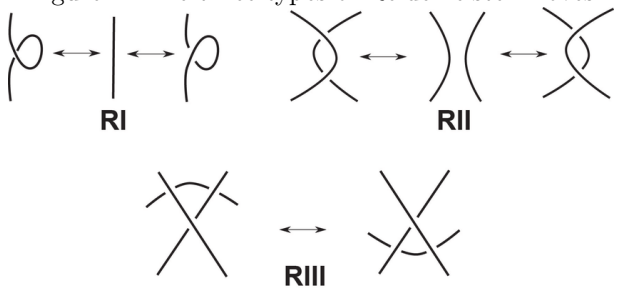
Ralph Fox conjectured the following [6]:

Conjecture 1 (The slice-ribbon conjecture) *Every slice knot is a ribbon knot.*

4.2 Reidemeister moves

We can define three types of moves called the Reidemeister moves [3] as follows,

Figure 1: The three types of Reidemeister moves

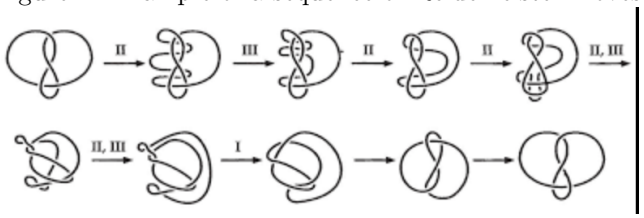


and the following theorem [3] shows that any two equivalent knot diagrams (by which it is meant that there exists an isotopy that maps one knot to the other) are related by a sequence of Reidemeister moves.

Theorem 2 *Any two equivalent (ambient isotopic) knot diagrams can be related via a sequence of Reidemeister moves.*

The figure below shows an example of how a knot can be isotopied into anther by Reidemeister moves.

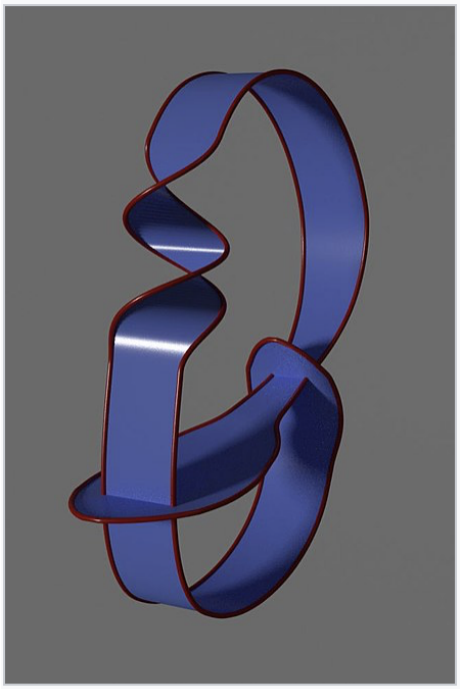
Figure 2: Example of a sequence of Reidemeister moves



4.3 Slice knots

The figure below shows an example of a slice knot, which is actually a ribbon knot as all the intersections are ribbon singularities.

Figure 3: Example of a slice knot



A 3-dimensional rendering of the ribbon knot 8_{20}, showing the ribbon property

Although the band intersects in the 3-dimensional space, we can remove the intersections in the extra fourth dimension, so the knot is slice.

In general, most knots are not slice. We can define and calculate knot variants to show a knot is not slice. One such invariant is the Alexander polynomial.

4.4 Alexander polynomial

One combinatorial way to define the Alexander polynomial is by the Skein relation. The Alexander polynomial satisfies the following properties

(I) *Invariance*: If L is equivalent to L', then $\nabla(L) = \nabla(L')$.

(II) *Normalization*: $\nabla(\bigcirc) = 1$ for the unknot $\bigcirc$.

(III) *Skein relation*:

$$\nabla\Big(\quad\Big) - \nabla\Big(\quad\Big) = x\nabla\Big(\quad\Big)$$

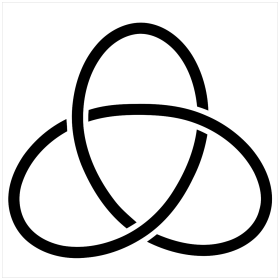
where in in (III), x stands for $t^{\frac{1}{2}} - t^{-\frac{1}{2}}$.

In fact, property (I) follows from properties (II) and (III) by showing that it is an invariant under Reidemeister moves due to the theorem that any two equivalent knot diagrams are related by a sequence of Reidemeister moves. Refer to Saveliev's book on 3-manifolds for a proof that the Alexasnder polynomial is well defined from (II) and (III).

The Alexander polynomial relates to sliceness via the Fox-Milnor condition [3].

Theorem 3 ((Fox-Milnor condition)) *The Alexander polynomial of a slice knot can be factored into $f(t)f(t^{-1})$, where $f(t)$ is some Laurent polynomial with integral coefficients.*

Using the Fox-Milnor condition, one can show that the trefoil knot is not slice, as its Alexander polynomial is $t - 1 + t^{-1}$.

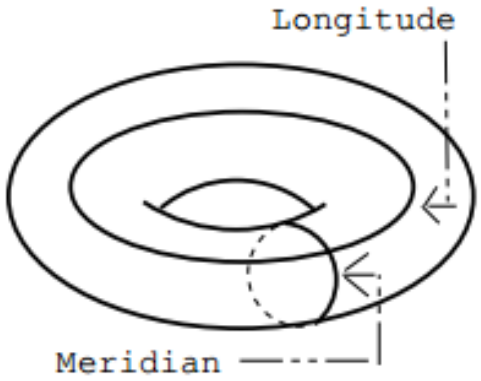


5 Surgery theory

5.1 2-torus

In order to define 0-surgery on a solid torus, we firstly need to define the meridian and longitude [4] of a torus.

Definition 5 *The meridian and longitude of a 2-torus are defined as a choice of loops such that the meridian bounds a disk in the solid torus and the longitude generates the fundamental group.*

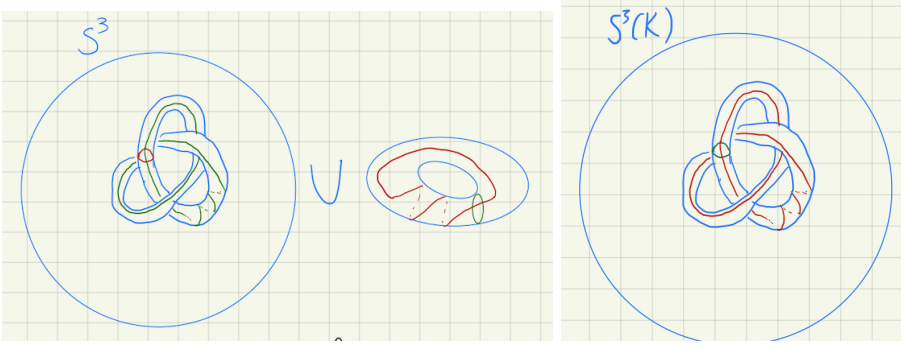


5.2 0-surgery

We have the following definition for 0-surgeries:

Definition 6 (0-surgery) *By a 0-surgery along a knot K in $\mathbb{S}^3$ we remove a tubular neighbourhood of K, which is diffeomorphic to a solid torus N, and paste N back by a diffeomorphism that maps the meridian to the longitude and the longitude to the meridian.*

Figure 4: (Left) Disjoint union of S^3 with a tubular neighbourhood of the knot removed and a solid torus. (right) 3-manifold after the 0-surgery

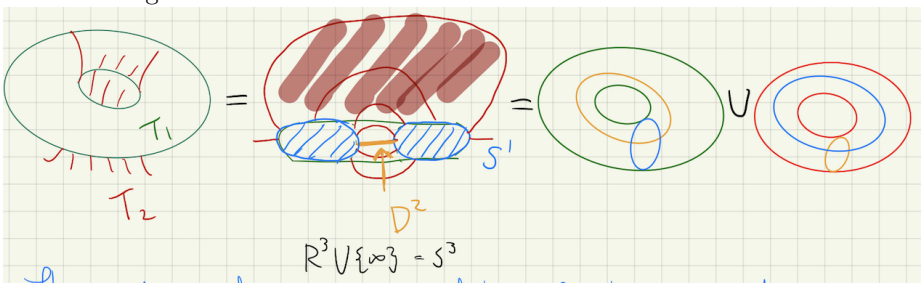


The figure above illustrates the process of doing the 0-surgery along a knot (the trefoil knot in this case). In the picture to the left, the red and blue loops indicate how the meridians and longitudes of the tubular neighbourhood of the knot and the solid torus are identified. We can see by comparing the colouring before and after the 0-surgery that the meridian and the longitude are switched.

5.2.1 Example: 0-surgery along the unknot

As an example, we show that doing 0-surgery along the unknot in S^3 gives $S^1 \times S^2$.

Figure 5: S^3 can be viewed as the union of two solid tori.

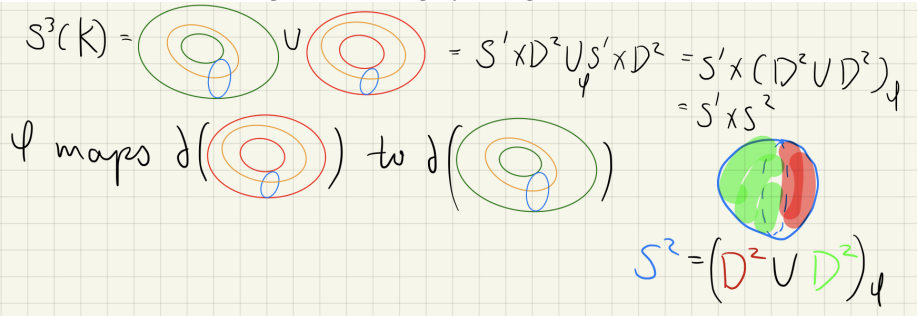


First of all, we show that S^3 is the union of two solid tori, where the meridian and the longitude of one torus get mapped to the longitude and the meridian, respectively, of the other torus.

One way to see this is by first noticing that $\mathbb{S}^3$ can be identified with $\mathbb{R}^3 \bigcup \{\infty\}$, the one-point compactification of $\mathbb{R}^3$ with its usual topology, via stereographic projection. Given a solid torus $\mathbb{T}$ in $\mathbb{R}^3 \bigcup \{\infty\}$, we show that the complement of the interior of $\mathbb{T}$ is $\mathbb{S}^1 \times \mathbb{D}^2$, a solid torus. To see this, consider the cross section of $\mathbb{T}$ with the centred vertical plane (parallel to the z-axis), which consists of two disks (shaded in blue). Then, we choose a point on the boundary of one blue circle. Then, consider the circle formed by the revolving the point chosen around the center of the torus. This circle bounds a disk $\mathbb{D}^2$ if the point chosen is not the furthest from the center of the solid torus, and we can make the disk such that it intersects with the vertical z-axis at exactly one point, and every point on the vertical z-axis intersects with such a disk, if the point chosen is not the furthest from the center of the torus. If the point chosen is the furthest from, the circle still bounds a disk with the point at infinity added. Therefore, the complement of the interior of $\mathbb{T}$ is again a solid torus.

We also notice that in our construction, the two solid tori are pasted by a diffeomorphism that maps the longitude and the meridian to the meridian and the longitude, respectively, of the other torus, because the the boundary of the disk in the complement of $\mathbb{T}$, as described above, is homotopic the longitude in $\mathbb{T}$.

Figure 6: 0-surgery along the unknot



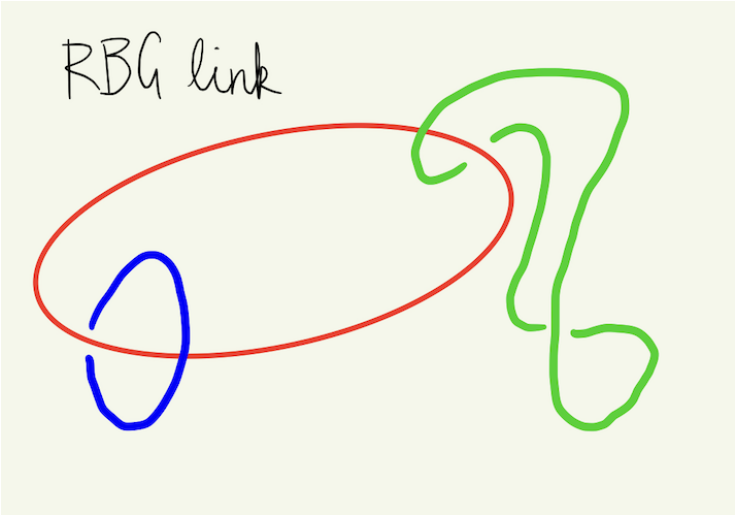
After doing the 0-surgery, we have two solid tori that are pasted together by a diffeomorphism that maps the longitude and meridian of one solid torus to the longitude and the meridian, respectively, of the other torus. Therefore, the resultant manifold is $\mathbb{S}^1 \times \mathbb{S}^2$.

5.3 RBG links

We define the RBG link [1] and explain a method that could potentially discover exotic 4-spheres.

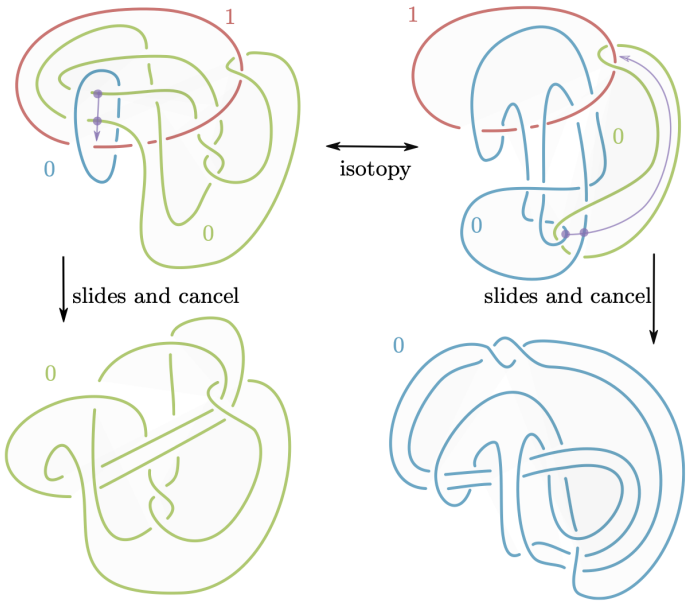
Definition 7 *An RBG link is a 3-component link where ignoring G makes B look like the meridian unknot around R, and ignoring B makes G look like the meridian unknot around R*

Figure 7: Example of an RBG link



The figure above shows a very simple example of an RBG link. The figure below [1], due to Ciprian Manolescu and Lisa Piccirillo, is more complicated as the areas bounded by the green and blue knots intersect. However, ignoring the green knot makes the blue one look like a meridian and the red one look like a longitude. Similarly, after an isotopy, ignoring the blue one make the green one look like the meridian and the red one look like a longitude.

Figure 8: "From zero surgeries to candidates for exotic definite four-manifolds" (C. Manolescu and L. Piccirillo), preprint (2021)



As shown in the figure above, we can construct knots K_B and K_G from the original RBG link by doing handle slides and cancellations [1].

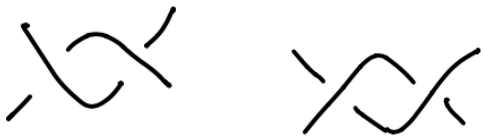
Our project is motivated by the following theorem [1], which can potentially lead to the discovery of an exotic 4-sphere.

Theorem 4 *From any RBG link we get a pair of knots (one green and one blue) with the same 0-surgery. If one of these knots is slice and the other is not slice, then we find an exotic smooth structure on* S^4

6 Constructing RBG links

6.1 Constructing links with the same 0-surgery

Figure 9: A pair of 2-twists with opposite crossing directions in a knot diagram

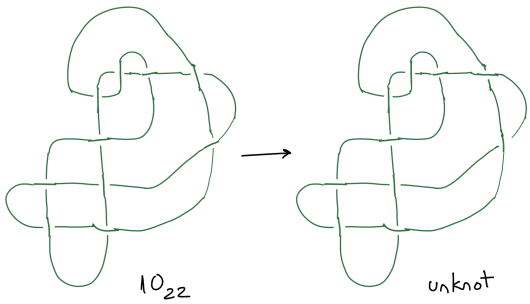


By detecting a 2-twist in a knot diagram and switching its overcrossing and undercrossing, if after the switch we get an unknot from a slice knot, then we construct an RBG diagram and then a knot K_B with the same 0-surgery as K_G. If K_B is not slice, then an exotic 4-sphere exists.

Going through a list of slice knots, our Python program identified the following slice knots with the previous property: K14n18348, K13n1019, K14a8282, K14n11711, K14n12149, K14a3615, K14n16279, K8a4, K14a19216, K14a6554, K14a19320, K14a8203, K12a1277, K13a2585, K13n659, K14n27023, K14n21983, K14n16582, K6a3, K14n9684, K10a117, K13a3662, K12n48, K13a1577, K12n501, K13n939, K12a3, K14n7469, K13a4187, K11a169, K10a112, K13n3246, K12a1029, K13a521, K11a103, K13a2115, K13a4094.

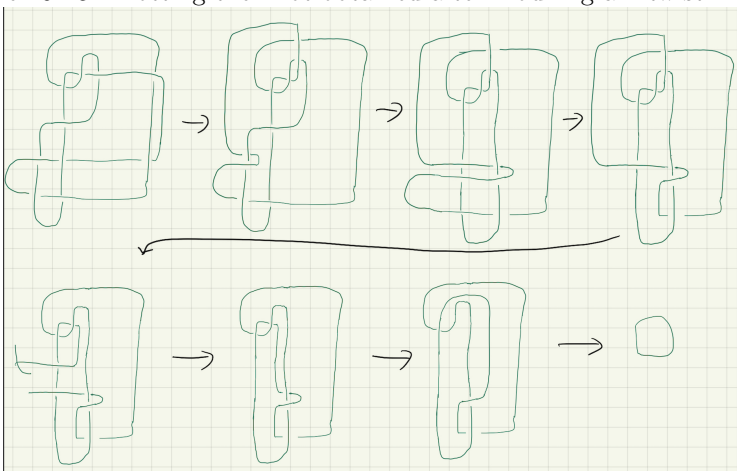
6.2 An RBG link constructed from 10_{22}

Using a Python program to go through a list of knots with unknown sliceness and identify those with the property on the previous slide, we identified that the knot 10_{22} is a 2-twist away from an unknot.



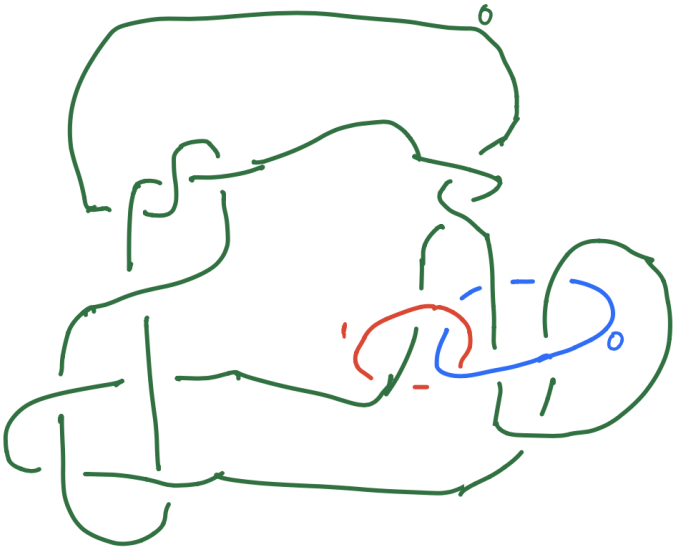
The figure below illustrates why the knot after changing a 2-twist is the unknot.

Figure 10: Unknotting the knot obtained after modifing a 2-twist in $10_2 2$



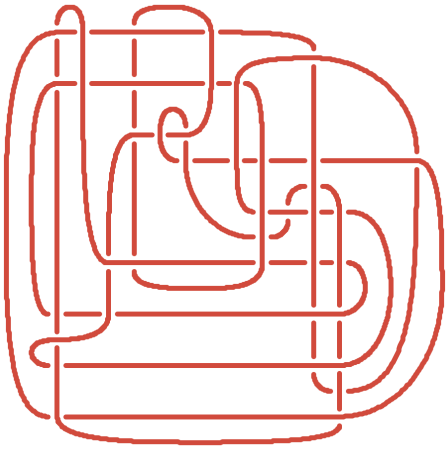
Finally, we construct an RBG link from $10_2 2$.

Figure 11: RBG link constructed from 10_{22}



6.3 K_B

Figure 12: K_B from the RBG link constructed from 10_{22}



This knot has the same 0-surgery as 10_{22}, and 10_{22} is known to be slice.

This is a potential counterexample to the smooth 4D Poincare conjecture: If one could show that this knot is not slice, then an exotic 4-sphere exists. (However, it's more likely that it is slice.)

We also have more examples like this. The difficulty is that we do not have algorithms to determine if knots are slice or not. Still, this is a possible approach towards the smooth 4D Poincare conjecture.

References

[1] Ciprian Manolescu, Lisa Piccirillo. *From zero surgeries to candidates for exotic definite four-manifolds*, preprint (2021), arXiv:2102.04391v2.

[2] Freedman, Michael Hartley. *The topology of four-dimensional manifolds.* J. Differential Geometry 17 (1982), no. 3, 357–453.

[3] Lickorish, W. (1997). *An Introduction to Knot Theory.*

[4] Saveliev, N. (2015). *Lectures on the Topology of 3-Manifolds - Toc.*

[5] Milnor, John W. (1956). *On manifolds homeomorphic to the 7-sphere.* Annals of Mathematics

[6] Fox, R. H. (1962).*Some problems in knot theory.* Topology of 3-manifolds and related topics

A Program: *slice_test.py*

```
import snappy
import csv
from ch2twist import func

# Populating slice.csv
createCSV = False
if createCSV:
    with open('slice_raw.csv', newline='') as csvfile:
        with open('slice.csv', 'w', newline='') as writefile:
            fieldnames = ['name', 'ribbon_bound']
            reader = csv.reader(csvfile, delimiter=',',
            ↪  quotechar='|')
            writer = csv.DictWriter(writefile,
            ↪  fieldnames=fieldnames)
            count = 0
            for row in reader:
                if count % 2 == 0:
                    print(row[0])
                    writer.writerow({'name': row[0],
                    ↪  'ribbon_bound': row[1]})
                count = count + 1

names = set()

# Reading from slice.csv
write = True
with open('slice.csv', newline='') as csvfile:
    with open('slice_results.csv', 'w', newline='') as
    ↪  writefile:
        fieldnames = ['name', 'original DT', 'modified DT',
        ↪  'sliceness', 'knot components after simplication']
        reader = csv.reader(csvfile, delimiter=',',
        ↪  quotechar='|')
        writer = csv.DictWriter(writefile,
        ↪  fieldnames=fieldnames)
        count = 0
        for row in reader:
            if count > 0:
                M = snappy.Manifold(row[0])
                DT = M.DT_code()
                out = func(DT[0], False)
                print(out)
                for each in out:
```

```
                    each = [tuple(i for i in each)]
                    L = snappy.Link("DT: {0}".format(each))
                    sliceness = L.simplify('global')
                    if len(L.DT_code()) == 0:
                        names.add(row[0])

                    writer.writerow({'name': row[0], 'original
                    ↪  DT': DT, 'modified DT': each,
                                    'sliceness': sliceness,
                                    ↪  'knot components
                                    ↪  after simplication':
                                    len(L.DT_code())})

                #print(M.DT_code()[0])
                #print(type(M.DT_code()[0]))
            count = count + 1

    print("Printing knots: \n")
    for each in names:
        print(each)
    print("Finished reading from slice.csv and displaying DT
    ↪  codes.")
```

B Program: *unknown_slice_test.py*

```
import snappy
import csv
from ch2twist import func

# Populating slice.csv
createCSV = False
if createCSV:
    with open('unknownslice_raw.csv', newline='') as csvfile:
        with open('unknownslice.csv', 'w', newline='') as
        ↪  writefile:
            fieldnames = ['name', 'num_cross', 'alt',
            ↪  'volume', 'det', 'alex', 'jones', 'genus3',
            ↪  'genus4', 'slice',
             'ribbon', 'sym_union', 'signature', 'hyp',
             ↪  'alex_test', 'sig_fn_0', 'ras_s_F2',
             ↪  'ras_s_Q',
             'ribbon_to', 'ribbon_bound',
             ↪  'ribbon_by_two_bands', 'L_space', 'fibered',
             ↪  'HFK', 'tau',
             'nu', 'epsilon', 'HFK_test', 'HKL_obs']
```

```
            reader = csv.reader(csvfile, delimiter=',',
            ↪  quotechar='|')
            writer = csv.writer(writefile)
            count = 0
            for row in reader:
                if count % 2 == 0:
                    writer.writerow(row)
                count = count + 1

# Reading from slice.csv
write = True
with open('unknownslice.csv', newline='') as csvfile:
    with open('unknown_slice_results.csv', 'w', newline='') as
    ↪  writefile:
        fieldnames = ['name', 'original DT', 'modified DT',
        ↪  'sliceness', 'knot components after simplication']
        reader = csv.reader(csvfile, delimiter=',',
        ↪  quotechar='|')
        writer = csv.DictWriter(writefile,
        ↪  fieldnames=fieldnames)
        count = 0
        for row in reader:
            if count > 0:
                M = snappy.Manifold(row[0])
                DT = M.DT_code()
                out = func(DT[0], False)
                print(out)
                for each in out:
                    each = [tuple(i for i in each)]
                    L = snappy.Link("DT: {0}".format(each))
                    sliceness = L.simplify('global')
                    writer.writerow({'name': row[0], 'original
                    ↪  DT': DT, 'modified DT': each,
                                     'sliceness': sliceness,
                                     ↪  'knot components
                                     ↪  after simplication':
                                     len(L.DT_code())})

                #print(M.DT_code()[0])
                #print(type(M.DT_code()[0]))
            count = count + 1

    print("Finished reading from slice.csv and displaying DT
    ↪  codes.")
```