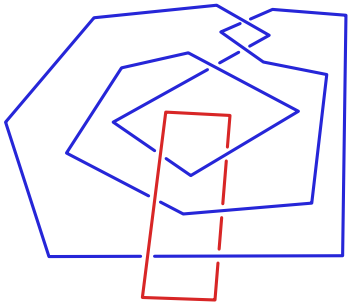
Additional documentation for "Exotic Mazur manifolds and knot trace invariants"

Kyle Hayden
Thomas E. Mark
Lisa Piccirillo

This is a guide to the computer-assisted proofs in [3]. All files referenced below are located on the site [2]. The calculations described rely on the software SnapPy [1], the knot Floer homology calculator [7], and the Khovanov homology calculator (aka SKnotJob) [6]. The SnapPy calculations can be verified by running SnapPy inside Sage [8]; see the official SnapPy documentation regarding verified computations.

A. Proof of Theorem 2.7

The claim to verify is that $V_0(P)$ is hyperbolic, where $V_0(P)$ is the 3-manifold given by zero-surgery on the Mazur pattern P in the solid torus V. For later use, we will also record SnapPy's volume estimate for $V_0(P)$. The exterior of the pattern P in the solid torus V is given by the exterior of the following 2-component link:



This diagram is saved as the SnapPy projection file `MazurPattern.lnk`. (Using SnapPy's link editor, we extracted a Dowker-Thistlethwaite code for this link, DT:[(-8,-14,18),(10,-6,-16,-4,-12,2)].) The manifold $V_0(P)$ is obtained by performing zero-surgery on the blue/second component. To verify that $V_0(P)$ is hyperbolic and obtain an estimate of its volume, we ran the commands in the file `VOP-input.py`. (These commands may be entered manually or by opening the file `VOP-input.py` in SnapPy.) The activity log is reproduced below; see the file `VOP-output.py` for a copy saved by SnapPy.

```
In:  P=Manifold('DT:[(-8,-14,18),(10,-6,-16,-4,-12,2)]')
In:  P.solution_type()
```

```
Out:  'all tetrahedra positively oriented'
In:  P.dehn_fill((0,1),1)
In:  V0P=P.filled_triangulation()
In:  V0P.solution_type()
Out:  'all tetrahedra positively oriented'
In:  V0P.volume()
Out:  3.66386237671
```

B. Proof of Theorem 1.1

We follow the notation in [3]. The link $L_1 = K_1 \cup C$ is saved as a SnapPy projection file `TrefPhone.lnk`, from which we extracted a corresponding DT code. To verify that $S^3_0(C) \setminus K_1$ is hyperbolic with $\mathrm{vol}(S^3_0(C) \setminus K_1) \approx 9$, we ran the commands in the file `L1-input.py`. A copy of the output is saved as `L1-output.py`.

The link $K \cup C \cup \alpha$ is saved as a SnapPy projection file `Lm.lnk`. As above, we used this to extract a corresponding DT code. To verify that $S^3_0(C) \setminus (K \cup \alpha)$ is hyperbolic with $\mathrm{vol}(S^3_0(C) \setminus (K \cup \alpha)) \approx 11$, we ran the commands in the file `Lm-input.py`. A copy of the output is saved as `Lm-output.py`.

C. Proof of Theorem 1.5

As above, we follow the notation in [3]. (In our files, we denote K by `K1` and K' by `K2`.) Diagrams of K and K' are saved as SnapPy projection files `K1.lnk` and `K2.lnk`. In addition, we saved a slightly simplified diagram for K as `K1simplified.lnk`; the two diagrams are related by an obvious Reidemeister I move, and this diagram simplification is required for use with the knot Floer homology calculator [7]. From these diagrams, we used SnapPy's link editor to extract Planar Diagram codes for K and K'. These PD codes are saved as `K1(PDcode).txt` and `K2(PDcode).txt`, formatted for use with [7]. We then used these files to complete the desired calculations, following the documentation for [7]; the resulting activity logs are copied into the summary files `K1summary.txt` and `K2summary.txt`.

Next we calculated the s-invariants of K and K' (with $\mathbb{F}_2$-coefficients). We used SnapPy's link editor to extract DT codes for K and K'. To comply with the formatting conventions of [6], we replaced commas with spaces in the DT codes, obtaining

- for K, the DT code $\mathcal{C}$ = -22 -68 -66 -24 80 52 -78 76 -28 34 -62 -56 -72 -18 44 -74 -16 -42 6 2 64 26 32 -14 12 -10 -50 8 38 -4 40 -20 -70 -58 -60 36 -30 -46 48 54
- for K', the DT code $\mathcal{C}'$ = -34 26 -40 -14 72 -106 98 64 -54 52 -50 48 -68 -102 42 -28 104 70 -24 66 100 30 -4 38 -82 84 -86 88 6 -76 110 -90 -56 46 80 -62 -96 12 32 -2 36 22 -20 18 -16 8 -74 108 -92 -58 44 78 60 94 -10

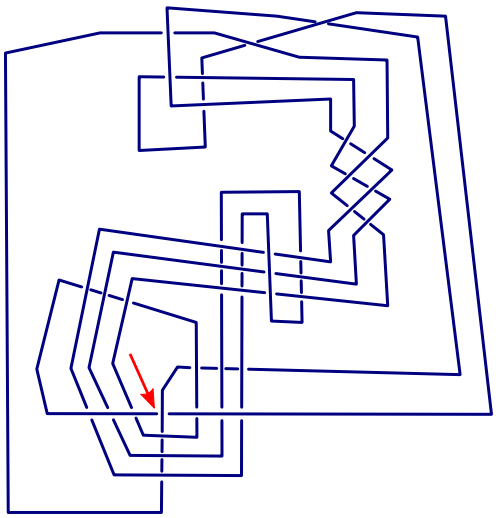
Before proceeding, we point out an ambiguity in the resulting calculation: a Dowker-Thistlethwaite code only determines a knot up to mirroring. Therefore we only use [6] to determine $|s(K)|$ and $|s(K')|$ (taking advantage of the fact that the s-invariant is negated by mirroring a knot). Below, we make a separate argument to show that $s(K), s(K') \geq 0$. For the computations of τ, this step is not strictly necessary, since [7] uses Planar Diagram codes (which are more robust) that we extracted directly from SnapPy's link editor. Still, we offer an additional argument that $\tau(K), \tau(K') \geq 0$ for the reader's peace of mind.

We entered the above DT codes $\mathcal{C}$ and $\mathcal{C}'$ into [6], which returned $s = 0$ for $\mathcal{C}$ and $s = -2$ for $\mathcal{C}'$. Therefore $s(K) = 0$ and $|s(K')| = 2$. Below we argue that $s(K), s(K') \geq 0$, letting us conclude that $s(K) = 0$ and $s(K') = 2$.

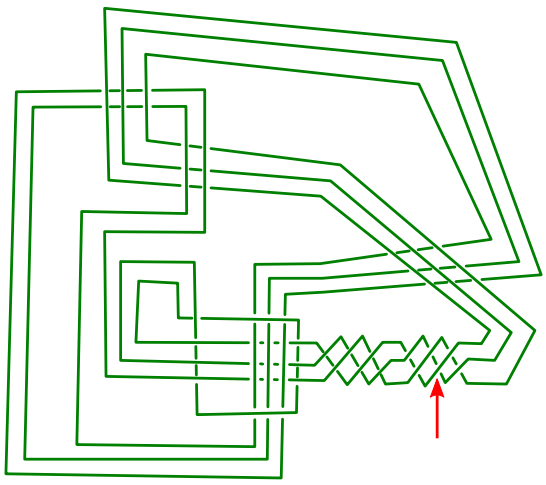
Verifying the signs of s and τ

We first point out that K has an unknotting crossing that is indicated in the diagram below. This crossing is positive, as can easily be seen by opening the file `K1.lnk` in SnapPy's link editor. By [5, Corollary 4.3] and [4, Corollary 3], we obtain $s(K) \geq 0$ and $\tau(K) \geq 0$.

For K', we take a similar approach. The crossing indicated in the diagram below is also seen to be positive; let J denote the knot obtained by changing this crossing. A diagram of J is saved as the SnapPy projection file `J.lnk`. As above, we used this file to extract DT and PD codes for J. Running these through [6] and [7] yielded $s(J) = 0$ and $\tau(J) = 0$. Since J was obtained from K' by changing a positive crossing to a negative crossing, we obtain $s(K') \geq 0$ and $\tau(K') \geq 0$.



K



K'

References

[1] **M Culler**, **N M Dunfield**, **M Goerner**, **J R Weeks**, *SnapPy, a computer program for studying the geometry and topology of* 3*-manifolds*, Available at http://snappy.computop.org

[2] **K Hayden**, **T E Mark**, **L Piccirillo**, Data and code to accompany *Exotic Mazur manifolds and knot trace invariants*, https://doi.org/10.7910/DVN/XFY7DL

[3] **K Hayden**, **T E Mark**, **L Piccirillo**, *Exotic Mazur manifolds and knot trace invariants*, Preprint (August 2018)

[4] **C Livingston**, *Computations of the Ozsváth-Szabó knot concordance invariant*, Geom. Topol. 8 (2004) 735–742

[5] **J Rasmussen**, *Khovanov homology and the slice genus*, Inventiones mathematicae 182 (2010) 419–447

[6] **D Schütz**, *SKnotJob*, Available at http://www.maths.dur.ac.uk/~dma0ds/knotjob.html (2018)

[7] **Z Szabó**, *Knot Floer homology calculator*, Available at https://web.math.princeton.edu/~szabo/HFKcalc.html (2019)

[8] **The Sage Developers**, *SageMath, the Sage mathematics software system*, Available at https://www.sagemath.org (2019)

Columbia University, New York, NY 10027

University of Virginia, Charlottesville, VA 22904

Brandeis University, Waltham, MA 02453

hayden@math.columbia.edu, tmark@virginia.edu, lpiccirillo@math.utexas.edu