Implementing Fast Khovanov Homology in C

Eric Min, Arjun Pandey, Kaya Young

MIT PRIMES-USA

November 2, 2026

Knot Theory

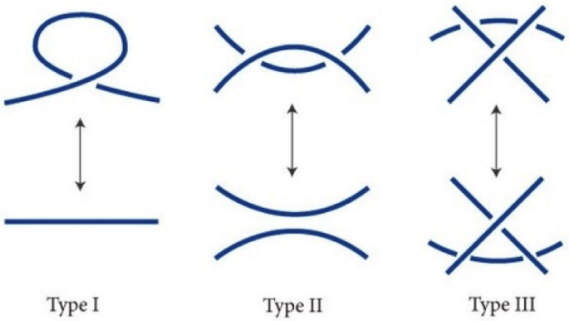
The main problem Knot Theory is concerned with is the distinguishing of various knots. A knot is a closed, continuous loop in three dimensions.

To differentiate knots, we use **invariants**. A knot invariant is a property that does not change under **reidemeister moves**.

Our goal is to study a very strong invariant, and to improve upon the implementations of known algorithms to attempt to speed up computation.

Reidemeister Moves

There are three Reidemeister Moves:



Reidemeister's Theorem states that a pair of knots/links are ambient isotopic iff one can equal the other through planar isotopies and Reidemeister moves.

Knot Projections

To draw knots, we essentially crush it into 2 dimensions. We can illustrate the overcrossings and undercrossings at intersections. This is a knot diagram.

Knot diagrams have strands. Strands are continuous segments that run from one undercrossing to the next.

p-colorability

To show invariance, one simply has to show invariance under the three Reidemeister moves. One such invariant is tricolorability. While weak, it is significantly easier to compute than other invariants.

Tricolorability is a binary invariant. We have 3 colors available, and the function returns true if there exists a coloring of strands such that at every crossing, the 3 strands are different colored, and at least 2 colors are used.

This can be expressed as a system of linear equations mod 3, one for every crossing. This can be generalized to p-colorability.

The Jones Polynomial

Another well-known invariant is the Jones Polynomial. The Jones Polynomial has four components: The Bracket Polynomial, a normalization term, the writhe, and a substitution.

The Bracket Polynomial satisfies the three following relations:

(1) $\langle \bigcirc \rangle = 1$

(2) $\langle L \cup \bigcirc \rangle = (-A^2 - A^{-2})\langle L \rangle$ for any link L

(3) $\langle \times \rangle = A \langle \asymp \rangle + A^{-1} \langle \rangle \langle \rangle$

Khovanov Homology

Ordinary Khovanov homology categorifies the Jones Polynomial, meaning it is strictly stronger. It is computed through assigning a chain complex to any given knot.

Each chain group in the complex is a direct sum of knot resolutions that with the same number of 1-resolutions. The differentials connecting the chain groups represent the relations between objects of one dimension and another.

Our goal is to implement this algorithm in detail in C. Existing implementations include Bar Natan's FastKh and Jeremy Green's JavaKh.

m and Δ maps

We assign each circle the module V with basis vectors v_- and v_+. Note that every state comprises of only circles after completely smoothing every crossing. Thus, a resolution with c circles contributes $V^{\otimes c}$

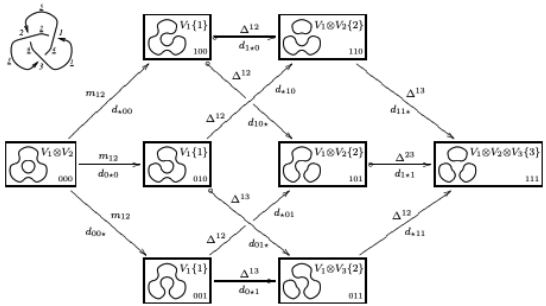
The maps between states are the merge and split maps, which satisfy:

$m(v_+ \otimes v_+) = v_+$
$m(v_+ \otimes v_-) = v_-$
$m(v_- \otimes v_+) = v_-$
$m(v_- \otimes v_-) = 0$
$\Delta(v_+) = v_+ \otimes v_- + v_- \otimes v_+$
$\Delta(v_-) = v_- \otimes v_-$

Two gradings

For a resolution s, let $r(s)$ be its 1 resolutions and $c(s)$ its circles. Let n_+ and n_- denote the number of positive and negative crossings, respectively. We have $i = r(s) - n_-$ and $k = r(s) + n_+ - 2n_-$.

Its contribution is $V^{\otimes c(s)}\{k\} \subset C^i$. Since $\deg(v_+)$ is 1 and $\deg(v_-)$ is -1, a tensor of circle labels has total quantum degree of the sums of the labels, added with k.



Fast Khovanov Homology

Computing Khovanov homology straight from the definition is expensive: a diagram with n crossings has a cube of 2^n resolutions.

Dror Bar-Natan's idea (*Fast Khovanov Homology Computations*, 2007) is **divide and conquer**. Cut the diagram into small pieces called **tangles**, compute an invariant of each piece, and glue the invariants together.

The catch is that gluing alone saves nothing: a tangle with n_1 crossings has 2^{n_1} objects, another with n_2 crossings has 2^{n_2}, and the glued complex has $2^{n_1} \cdot 2^{n_2} = 2^{n_1+n_2}$ objects. The savings come from **simplifying** each piece *before* gluing.

Khovanov Homology for Tangles

A **tangle** is a piece of a knot diagram inside a disk, with some strands running off the boundary. Bar-Natan (2005) extended Khovanov's construction to tangles: the cube of resolutions is the same, but its vertices are now **crossingless tangles** and its edges are **cobordisms**, surfaces between them such as the merge and split behind m and Δ.

Flattening the cube gives a chain complex $[\![T]\!]$ over a **category of cobordisms**:

- objects: crossingless tangles with the same boundary as T, with degree shifts $\{k\}$;
- morphisms: linear combinations of cobordisms, modulo local relations.

After degree shifts that depend only on the crossing signs, this becomes the Khovanov complex $\mathrm{Kh}(T)$, which is invariant up to homotopy. Gluing tangles side by side becomes a tensor product of complexes: $[\![D(T_1, T_2)]\!] = D([\![T_1]\!], [\![T_2]\!])$.

Cobordisms Modulo Local Relations

Cobordisms may carry **dots**, and we impose four local relations:

- a sphere is 0;
- a sphere with one dot is 1;
- a component with two dots is 0;
- **neck cutting:** cutting a tube and capping off both ends gives two terms, one with a dot on the first cap and one with a dot on the second.

These mirror the algebra of $V = \mathbb{Z}[X]/(X^2)$: $X^2 = 0$ and $\Delta(1) = 1 \otimes X + X \otimes 1$. They also force every closed surface to equal a plain number, which is why the final complex will only have scalars as morphisms.

Tool 1: Delooping

An object containing a closed loop can be replaced by two copies of itself with the loop removed, shifted in degree by $+1$ and -1:

$$\bigcirc \cong \varnothing\{+1\} \oplus \varnothing\{-1\}$$

The isomorphism is built from caps and cups, some of them dotted, and checking that the composites are identities uses all of the relations from the last slide. It is the cobordism version of $V \cong \mathbb{Z}\{+1\} \oplus \mathbb{Z}\{-1\}$, since 1 and X have degrees $+1$ and -1.

Delooping makes the complex bigger, but every object is now a loopless crossingless tangle, and there are only $C_k = \frac{1}{k+1}\binom{2k}{k}$ of those with $2k$ boundary points. Few possible objects means many repeats, so many morphisms are likely to be isomorphisms, which is exactly what the next tool needs.

Tool 2: Gaussian Elimination

If one entry $\varphi : b_1 \to b_2$ of a differential is an isomorphism, we can cancel b_1 and b_2 and correct the other maps:

$$C \xrightarrow{\binom{\alpha}{\beta}} b_1 \oplus D \xrightarrow{\begin{pmatrix} \varphi & \delta \\ \gamma & \epsilon \end{pmatrix}} b_2 \oplus E \xrightarrow{(\mu \quad \nu)} F$$

$$\simeq \quad C \xrightarrow{\beta} D \xrightarrow{\epsilon - \gamma\varphi^{-1}\delta} E \xrightarrow{\nu} F$$

Here $\simeq$ is homotopy equivalence, so the homology does not change. This works in any additive category, so it applies to complexes of cobordisms. In practice the isomorphisms that appear are identity cobordisms (up to a unit) between two copies of the same loopless tangle.

Example: The Figure-Eight Knot

The figure-eight knot has 4 crossings, so its cube has $2^4 = 16$ vertices. Bar-Natan cuts it into two 2-crossing tangles.

Each half starts with $2^2 = 4$ objects. After delooping and Gaussian elimination, each has just 3, so gluing them gives $3 \cdot 3 = 9$ objects instead of 16.

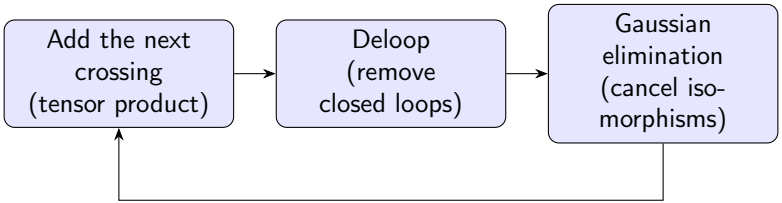
Delooping the glued complex and cancelling isomorphisms repeatedly leaves 6 objects, and over $\mathbb{Q}$ every differential is 0 (the underline marks homological degree 0):

$$\varnothing\{-5\} \xrightarrow{0} \varnothing\{-1\} \xrightarrow{0} \underline{\varnothing\{-1\} \oplus \varnothing\{1\}} \xrightarrow{0} \varnothing\{1\} \xrightarrow{0} \varnothing\{5\}$$

So over $\mathbb{Q}$, Kh of the figure-eight knot is 6-dimensional, with generators in (homological, quantum) bidegrees $(-2,-5)$, $(-1,-1)$, $(0,-1)$, $(0,1)$, $(1,1)$, $(2,5)$.

The Scanning Algorithm

Rather than cutting the knot in halves, **scan** across the diagram, adding one crossing at a time and simplifying after each step.



Repeat for every crossing

Once the last crossing is added, no boundary points remain: every object is a shifted empty diagram, and every morphism is a scalar. Over $\mathbb{Q}$, we continue cancelling until all differentials are zero, then read off the homology. Over $\mathbb{Z}$, only ± 1 can be cancelled, and remaining entries such as 2 can contribute to torsion.

Why It Is Fast

Bar-Natan's (non-rigorous) estimate: if the diagram is W strands wide at its widest cut, the complex for a partial diagram has size about C^W for some constant C.

- Gluing two halves means holding a double complex of size about C^{2W}.
- Scanning only tensors with a single crossing (size 2), so the biggest double complex has size about $2C^W$.

So the bottleneck is where the diagram is **widest**, not the total number of crossings n.

His unoptimized implementation, FastKh, computed the Khovanov homology of the 35-crossing torus knot $T_{7,6}$ in about a day, a task he estimated would take about 1000 years without tangles. Jeremy Green's JavaKh handles the 48-crossing torus knot $T_{8,7}$ in a few minutes.

References

- D. Bar-Natan, *Fast Khovanov homology computations*, J. Knot Theory Ramifications **16** (2007), no. 3, 243–255. arXiv:math/0606318.
- D. Bar-Natan, *Khovanov's homology for tangles and cobordisms*, Geom. Topol. **9** (2005), 1443–1499. arXiv:math/0410495.