Yarn-Ball Knots

Dror Bar-Natan, http://drorbn.net/fi20

The Fields Institute, November 2020

Agenda. Modest, off-topic, light conversation.

Abstract. Let there be scones! Our view of knot theory is biased in favour of pancakes. Technically, if K is a 3D knot that fits in volume V (assuming fixed-width yarn), then its projection to 2D will have about $V^{4/3}$ crossings. You'd expect genuinely 3D quantities associated with K to be computable straight from a 3D presentation of K. Yet we can hardly ever circumvent this $V^{4/3} \gg V$ "projection fee". Exceptions probably include the hyperbolic volume and certainly include finite type invariants (as we shall prove). But knot polynomials and knot homologies seem to always pay the fee.

If you can, please turn your video on! (And mic, whenever needed).

Thanks for inviting me to the Fields Institute! As most of you have never been there, here's a picture of the lecture room:

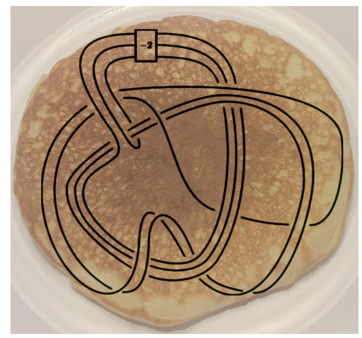
A recurring question in knot theory is "do we have a 3D understanding of our invariant?"

- See Witten and the Jones polynomial.
- See Khovanov homology.

I'll talk about my perspective on the matter...

We often think of knots as planar diagrams. 3-dimensionally, they are embedded in "pancakes".
This matters when

- We make statements about "random knots".
- We figure out computational complexity.



Knot by Lisa Piccirillo, pancake by DBN

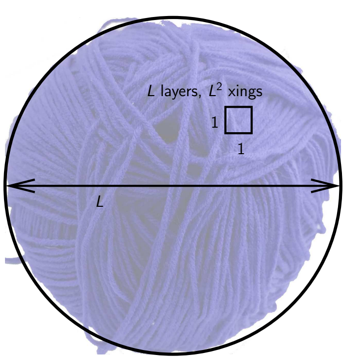
Yarn ball courtesy of Heather Young

'Connector' by Alexandra Griess and Jorel Heid (Hamburg, Germany). Image from www.waterfrontbia.com/ice-breakers-2019-presented-by-ports/.

$V \sim L^3$

$n = \text{xing number} \sim L^2 L^2 = L^4 = V^{4/3}$

("$\sim$" means "equal up to constant terms and log terms")

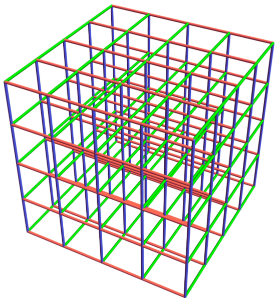


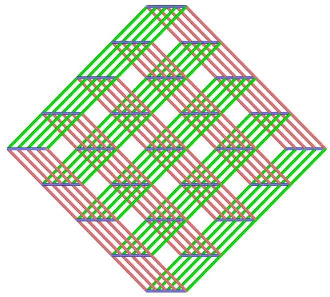
Conversation Starter 1. A knot invariant ζ is said to be Computationally 3D, or C3D, if

$$C_\zeta(3D, V) \ll C_\zeta(2D, V^{4/3}).$$

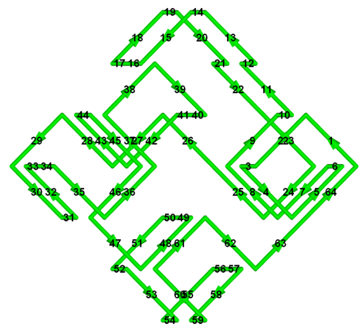
This isn't a rigorous definition! It is time- and naïveté-dependent! But there's room for less-stringent rigour in mathematics, and on a philosophical level, our definition means something.

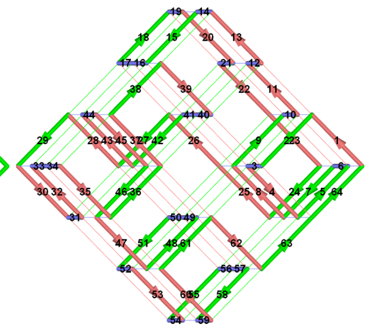
Theorem 1. Let lk denote the linking number of a 2-component link. Then $C_{lk}(2D, n) = n$ while $C_{lk}(3D, V) = V$, so lk is C3D!
Proof. WLOG, we are looking at a link in a grid, which we project as on the right:





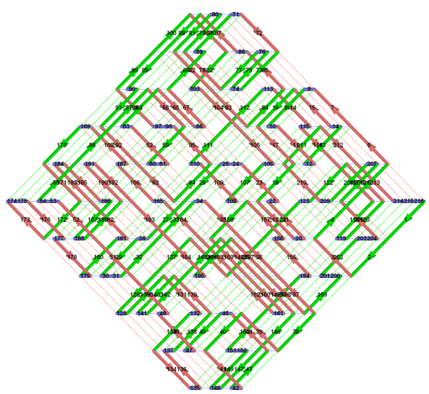
Here's what it look like, in the case of a knot:





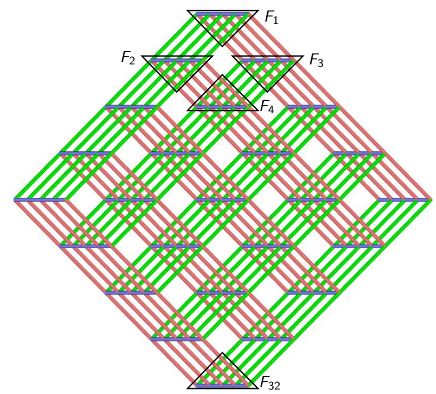
And here's a bigger knot.

This may look like a lot of information, but if V is big, it's less than the information in a planar diagram, and it is easily computable.



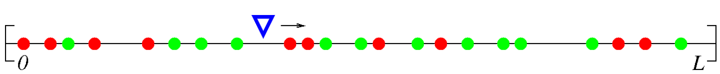
There are $2L^2$ triangular "crossings fields" F_k in such a projection.

WLOG, in each F_k all over strands and all under strands are oriented in the same way and all green edges belong to one component and all red edges to the other.



So $2L^2$ times we have to solve the problem "given two sets R and G of integers in $[0, L]$, how many pairs $\{(r, g) \in R \times G \colon r < g\}$ are there?". This takes time $\sim L$ (see below), so the overall computation takes time $\sim L^3$.

Below. Start with $rb = cf = 0$ ("reds before" and "cases found") and slide $\triangledown$ from left to right, incrementing rb by one each time you cross a $\bullet$ and incrementing cf by rb each time you cross a $\bullet$:



Great Embarrassment 1. I don't know if any of the Alexander, Jones, HOMFLY-PT, and Kauffman polynomials is C3D. I don't know if any Reshetikhin-Turaev invariant is C3D. I don't know if any knot homology is C3D.

Or maybe it's a cause for optimism — there's still something very basic we don't know about (say) the Jones polynomial. Can we understand it well enough 3-dimensionally to compute it well? If not, why not?

Conversation Starter 2. Similarly, if η is a stingy quantity (a quantity we expect to be small for small knots), we will say that η has Savings in 3D, or "has S3D" if $M_\eta(3D, V) \ll M_\eta(2D, V^{4/3})$.

Example (with van der Veen). It is probably true that the hyperbolic volume has *S3D*.

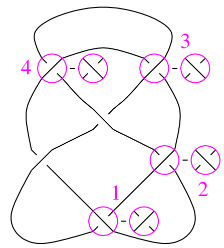
Great Embarrassment 2. I don't know if the genus of a knot has S3D! In other words, even if a knot is given in a 3-dimensional, the best way I know to find a Seifert surface for it is to first project it to 2D, at a great cost.

Theorem 2. If ζ is a finite type invariant of type d then $C_\zeta(3D, V)$ is at most $\sim V^d$.

It is known that $C_\zeta(2D, n)$ is at most $\sim n^d$ (e.g., my *Polynomial Invariants are Polynomial*), and one may expect that for most ζ, $C_\zeta(2D, n)$ is no better than $\sim n^d$ (exceptions: high coefficients of the Alexander polynomial and other poly-time knot polynomials).

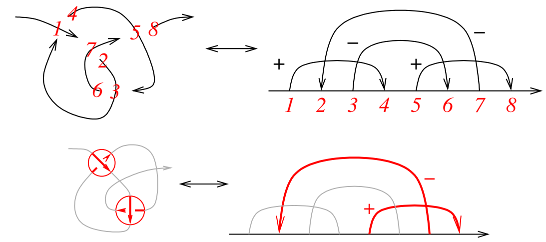
As $n^d \sim V^{\frac{4}{3}d} \gg V^d$, Theorem 2 says "most finite type invariants are C3D; the ones in doubt are the lucky few that can be computed unusually quickly".

A knot invariant is "type $d=3$" if it vanishes on all $(d+1=4)$-cubes like



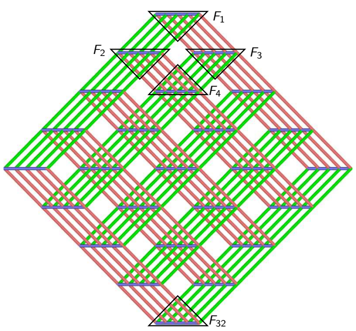
All pre-categorification knot polynomials are power series whose coefficients are finite type invariants. (This is sometimes helpful for the computation of finite type invariants, but rarely helpful for the computation of knot polynomials).

Gauss diagrams and sub-Gauss-diagrams:

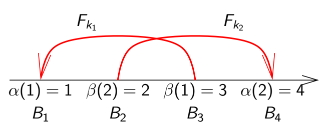


Let $\varphi_d\colon \{\text{knot diagrams}\} \to \langle\text{Gauss diagrams}\rangle$ map every knot diagram to the sum of all the sub-diagrams of its Gauss diagram which have at most d arrows.

Under-Explained Theorem (Goussarov-Polyak-Viro). A knot invariant ζ is of type d iff there is a linear functional ω on $\langle\text{Gauss diagrams}\rangle$ such that $\zeta = \omega\circ\varphi_d$.



Strategy. Count instances of  by counting instances that fall into specific crossing fields as follows:



here each B_i is the set of green/red strands within the relevant crossing field, having some specified orientations. There are two functions, $t\colon B_i\to\mathbb{Z}$ and $z\colon B_i\to[0..L]$ defined on each B_i.

Just picking the crossing fields costs $\sim L^{2d}$. The rest is handled by the following:

Counting Problem. Given $\alpha(j),\beta(j)\in[1..2d]$ for $j\in[1..d]$, given $2d$ "buckets" — sets B_i with $i\in[1..2d]$ — and given functions $t\colon (B:=\cup B_i)\to\mathbb{Z}$ and $z\colon B\to[0..L]$ such that $z|_{B_i}$ is injective, compute $|A|$, where

$$A=\left\{b=(b_i)_{i=1}^{2d}\in\prod B_i\colon \begin{array}{c} t(b_1)<t(b_2)<\ldots<t(b_{2d}) \\ \forall j\in[1..d],\ z(b_{\alpha(j)})<z(b_{\beta(j)}) \end{array}\right\}.$$

Proposition (Itai Bar-Natan). This computation can be carried out in time $\sim L^d$.

And hence the full computation of φ_d takes time $\sim L^{2d}L^d=V^d$, as claimed.

Lemma (same thing, minus the z data and conditions). Given $2d$ "buckets" — sets B_i with $i\in[1..2d]$ — and given $t\colon (B:=\cup B_i)\to\mathbb{Z}$. Assuming $|B_i|\sim L$, the quantity

$$N=\left|\left\{b=(b_i)_{i=1}^{2d}\in\prod B_i\colon t(b_1)<t(b_2)<\ldots<t(b_{2d})\right\}\right|$$

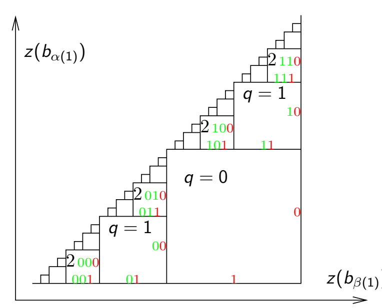
can be computed in time $\sim L^2$ (in fact, $\sim L$, but we don't need that).
Proof of Lemma. For $\iota\in[1..2d]$ and $\tau\in t(B)$ let

$$N_{\iota,\tau}=\left|\left\{b=(b_i)_{i=1}^{\iota}\in\prod B_i\colon t(b_1)<t(b_2)<\ldots<t(b_\iota)=\tau\right\}\right|.$$

Then each $N_{\iota,\tau}$ is computable from the $N_{\iota-1,\tau}$ in time $\sim L$, and there are $\sim L$ such computations to carry out. This done, $N=\sum_{\tau\in t(B)}N_{2d,\tau}$.

$\square$

Re-inserting z — the idea.



Proof of Bar-Natan's Proposition. WLOG $L+1=2^p$ for some $p\in\mathbb{N}$. Let $S_q=\{0,1\}^q$ and let $S=\bigcup_{q=0}^{p-1}S_q$ be the set of binary sequences of any length $q\in[0..p-1]$. Let $\sigma=(\sigma_j)_{j=1}^d$ a d-tuple of such binary sequences, where the length of σ_j is $|\sigma_j|=q_j\in[0..p-1]$. Then

$$A=\bigcup_{q\in[0..p-1]^d}A_q \qquad\text{where}\qquad A_q=\bigcup_{\sigma\colon \forall j\,|\sigma_j|=q_j}A_\sigma \qquad\text{and}$$

$$A_\sigma=\left\{b=(b_i)_{i=1}^{2d}\in\prod B_i\colon \begin{array}{c} t(b_1)<t(b_2)<\ldots<t(b_{2d}) \\ \forall j\in[1..d]\ \begin{array}{c} z(b_{\alpha(j)})=\sigma_j0* \\ z(b_{\beta(j)})=\sigma_j1* \end{array}\ \text{(in binary)} \end{array}\right\}.$$

By the lemma, $|A_\sigma|$ can be computed in time $\sim\left(\frac{L}{2^{\min|\sigma_j|}}\right)^2$. There are $2^{\sum q_j}$ A_σ's in A_q. so A_q can be computed in time $\sim 2^{\sum q_j}\left(\frac{L}{2^{\min q_j}}\right)^2$. The number of choices for q is ~ 1, so the only term that matters is the worst-case term, which is when for all j, $q_j=p-1$. In this case the computation time is $\sim\left(2^{p-1}\right)^d\left(\frac{L}{2^{p-1}}\right)^2\sim L^d\cdot 1^2=L^d$.

$\square$

If time — a word about braids.

Thank You!