Unknot recognition in quasi-polynomial time

Marc Lackenby

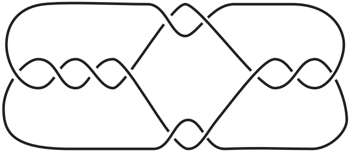
February 2021

Unknot recognition

Given a knot diagram, can we decide whether it represents the unknot?

Unknot recognition

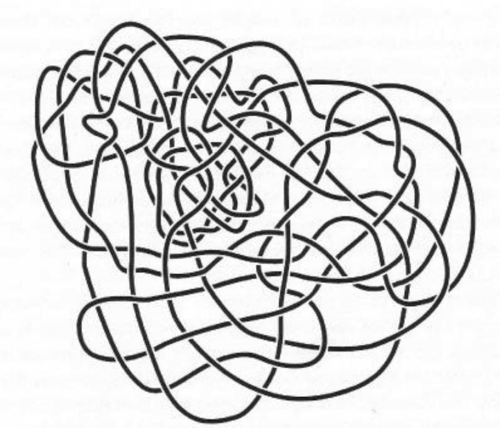
Given a knot diagram, can we decide whether it represents the unknot?



Goeritz's unknot

Unknot recognition

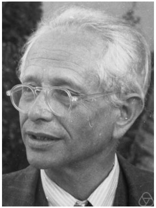
Given a knot diagram, can we decide whether it represents the unknot?



Haken's unknot

Early history

The problem was first formulated by Dehn in 1910.



Early history

The problem was first formulated by Dehn in 1910.

It was mentioned by Turing in his 1954 paper Solvable and Unsolvable Problems.

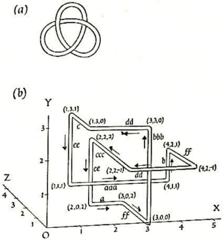
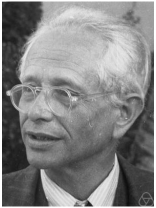
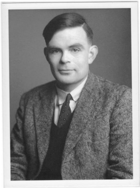


Fig. 1. (*a*) The trefoil knot (*b*) a possible representation of this knot as a number of segments joining points.





Early history

The problem was first formulated by Dehn in 1910.

It was mentioned by Turing in his 1954 paper Solvable and Unsolvable Problems.

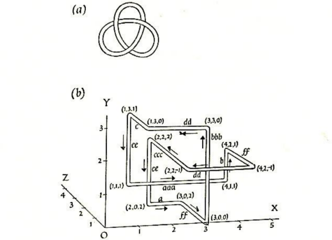
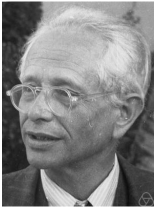
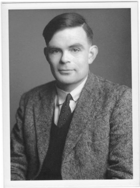


Fig. 1. (a) The trefoil knot (b) a possible representation of this knot as a number of segments joining points.





Theorem: [Haken, 1961] There is an algorithm to determine whether a given knot is the unknot.

Many other approaches

- Normal surfaces [Haken, Hass-Lagarias-Pippenger]
- Geometric structures [Thurston]
- Representations of π_1 [Kuperberg]
- Hierarchies [Agol, L]
- Khovanov homology [Kronheimer-Mrowka]
- Heegaard Floer homology [Ozsváth-Szabó, Sarkar-Wang, Manolescu-Ozsváth-Sarkar]
- Arc presentations [Dynnikov, L]
- Reidemeister moves [Hass-Lagarias, L]
- Pachner moves [Mijatovic]

A polynomial time solution?

Unsolved problem: Can we solve unknot recognition in polynomial time?

A polynomial time solution?

Unsolved problem: Can we solve unknot recognition in polynomial time?

[Thurston 2011] ‘A lot of people have thought about this question.’ ‘I think it’s entirely possible that there’s a polynomial-time combinatorial algorithm to unknot an unknottable curve, but this has been a very hard question to resolve.’

The complexity of unknot recognition

Theorem: [Hass-Lagarias-Pippenger 1999, L 2014] Unknot recognition lies in NP.

The complexity of unknot recognition

Theorem: [Hass-Lagarias-Pippenger 1999, L 2014] Unknot recognition lies in NP.

Theorem: [Kuperberg 2014, Agol 2002, L 2016] Unknot recognition lies in co-NP.

The complexity of unknot recognition

Theorem: [Hass-Lagarias-Pippenger 1999, L 2014] Unknot recognition lies in NP.

Theorem: [Kuperberg 2014, Agol 2002, L 2016] Unknot recognition lies in co-NP.

Main Theorem: [L 2021] There is an algorithm to determine whether a diagram with n crossings is the unknot that completes in time $n^{O(\log n)}$.

The complexity of unknot recognition

Theorem: [Hass-Lagarias-Pippenger 1999, L 2014] Unknot recognition lies in NP.

Theorem: [Kuperberg 2014, Agol 2002, L 2016] Unknot recognition lies in co-NP.

Main Theorem: [L 2021] There is an algorithm to determine whether a diagram with n crossings is the unknot that completes in time $n^{O(\log n)}$. 'quasi-polynomial time'

Hierarchies

Let M be a compact orientable 3-manifold, for example $S^3 - \mathrm{int}(N(K))$ for K a knot

Hierarchies

Let M be a compact orientable 3-manifold, for example $S^3 - \mathrm{int}(N(K))$ for K a knot

A hierarchy is a sequence of 3-manifolds $M = M_1, \ldots, M_{\ell+1}$ and orientable surfaces $S_1, \ldots, S_\ell$ such that each S_i is properly embedded in M_i and $M_{i+1} = M_i \backslash\backslash S_i$.

Hierarchies

Let M be a compact orientable 3-manifold, for example $S^3 - \mathrm{int}(N(K))$ for K a knot

A hierarchy is a sequence of 3-manifolds $M = M_1, \ldots, M_{\ell+1}$ and orientable surfaces $S_1, \ldots, S_\ell$ such that each S_i is properly embedded in M_i and $M_{i+1} = M_i \backslash\backslash\, S_i$.

We do not require the surfaces to be incompressible or for the final manifold $M_{\ell+1}$ to be balls (although we will be aiming to produce such hierarchies).

Boundary patterns

A boundary pattern for a 3-manifold M is subset P of ∂M that is a disjoint union of simple closed curves and trivalent graphs.

Boundary patterns

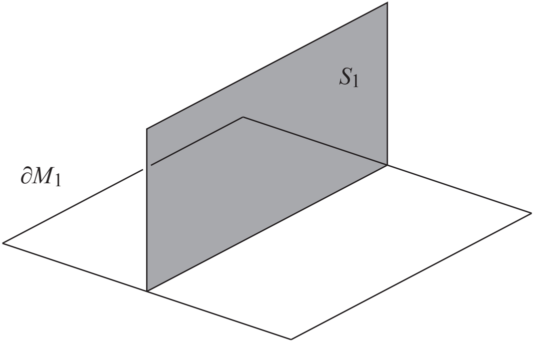
A boundary pattern for a 3-manifold M is subset P of ∂M that is a disjoint union of simple closed curves and trivalent graphs.

The manifolds in a hierarchy naturally inherit a boundary pattern.

Boundary patterns

A boundary pattern for a 3-manifold M is subset P of ∂M that is a disjoint union of simple closed curves and trivalent graphs.

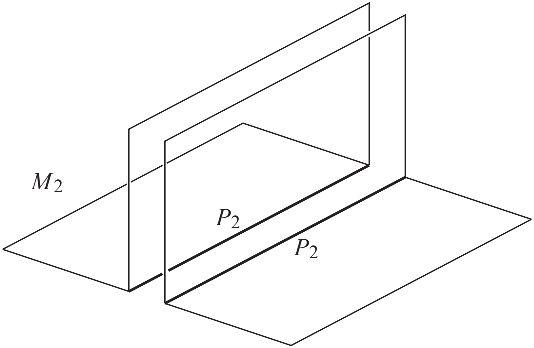
The manifolds in a hierarchy naturally inherit a boundary pattern.



Boundary patterns

A boundary pattern for a 3-manifold M is subset P of ∂M that is a disjoint union of simple closed curves and trivalent graphs.

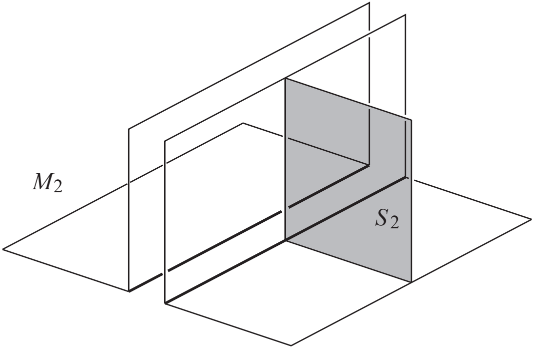
The manifolds in a hierarchy naturally inherit a boundary pattern.



Boundary patterns

A boundary pattern for a 3-manifold M is subset P of ∂M that is a disjoint union of simple closed curves and trivalent graphs.

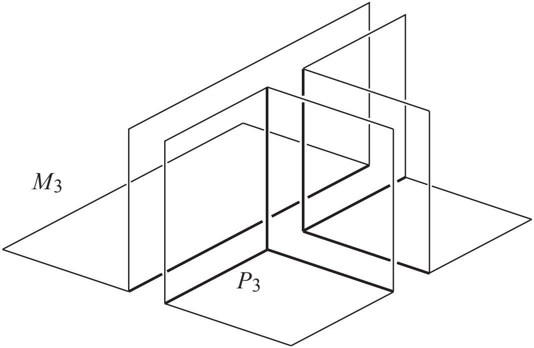
The manifolds in a hierarchy naturally inherit a boundary pattern.



Boundary patterns

A boundary pattern for a 3-manifold M is subset P of ∂M that is a disjoint union of simple closed curves and trivalent graphs.

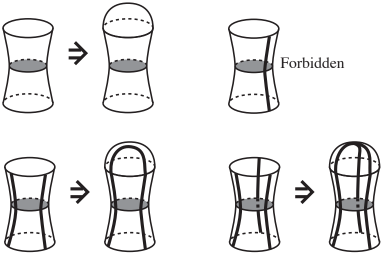
The manifolds in a hierarchy naturally inherit a boundary pattern.



Essential boundary patterns

A boundary pattern P is essential if, for any properly embedded disc D that intersects P at most three times, ∂D bounds a disc D' in ∂M that intersects P in one of the following:

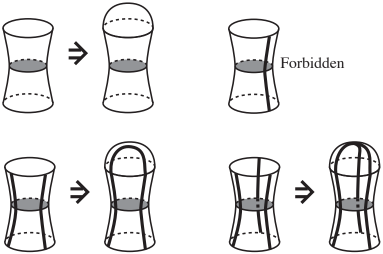
- the empty set,
- an arc,
- a tripod.



Essential boundary patterns

A boundary pattern P is essential if, for any properly embedded disc D that intersects P at most three times, ∂D bounds a disc D' in ∂M that intersects P in one of the following:

- the empty set,
- an arc,
- a tripod.



A disc D properly embedded in M that intersects P at most three times for which ∂D does not bound a disc D' in ∂M as above is a violating disc.

Essential hierarchies

A hierarchy $M = M_1, \ldots, M_{\ell+1}$ is essential if the final manifold $M_{\ell+1}$ inherits an essential boundary pattern.

Essential hierarchies

A hierarchy $M = M_1, \ldots, M_{\ell+1}$ is essential if the final manifold $M_{\ell+1}$ inherits an essential boundary pattern.

Theorem: [Waldhausen, Johansson] Let M be a compact orientable 3-manifold with non-empty boundary and empty boundary pattern. Then the following are equivalent:

Essential hierarchies

A hierarchy $M = M_1, \ldots, M_{\ell+1}$ is essential if the final manifold $M_{\ell+1}$ inherits an essential boundary pattern.

Theorem: [Waldhausen, Johansson] Let M be a compact orientable 3-manifold with non-empty boundary and empty boundary pattern. Then the following are equivalent:

- ∂M is incompressible and M is irreducible;

Essential hierarchies

A hierarchy $M = M_1, \ldots, M_{\ell+1}$ is essential if the final manifold $M_{\ell+1}$ inherits an essential boundary pattern.

Theorem: [Waldhausen, Johansson] Let M be a compact orientable 3-manifold with non-empty boundary and empty boundary pattern. Then the following are equivalent:

- ∂M is incompressible and M is irreducible;
- M has an essential hierarchy where the final manifold is a union of balls.

Essential hierarchies

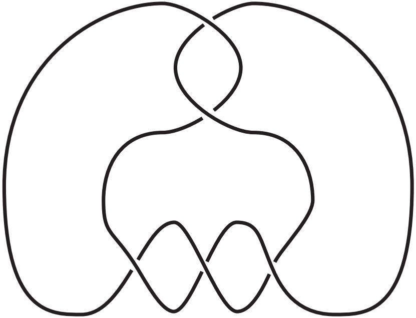
A hierarchy $M = M_1, \ldots, M_{\ell+1}$ is essential if the final manifold $M_{\ell+1}$ inherits an essential boundary pattern.

Theorem: [Waldhausen, Johansson] Let M be a compact orientable 3-manifold with non-empty boundary and empty boundary pattern. Then the following are equivalent:

- ∂M is incompressible and M is irreducible;
- M has an essential hierarchy where the final manifold is a union of balls.

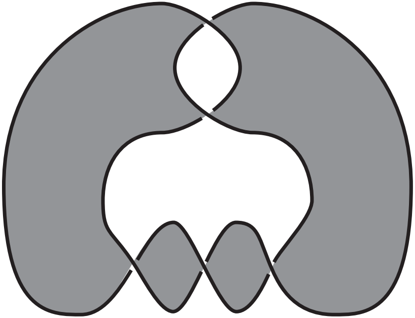
So, we will use essential hierarchies as a way of proving that a knot is non-trivial.

Example



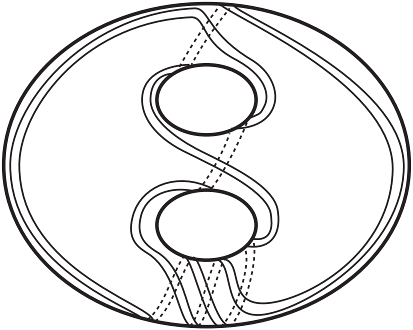
(i) The knot 5_2

Example



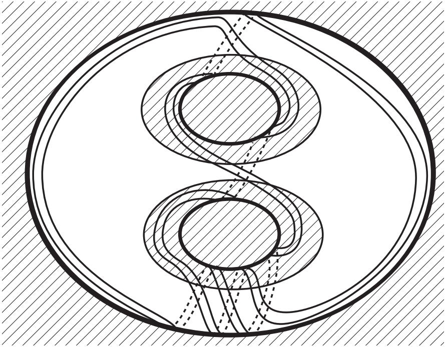
(ii) The first surface in the hierarchy

Example



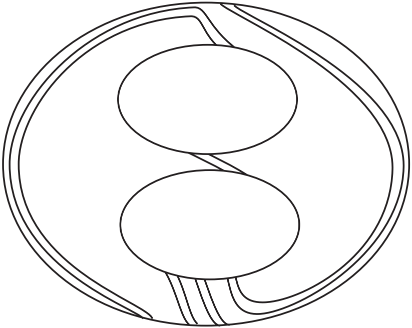
(iii) The exterior of this surface

Example



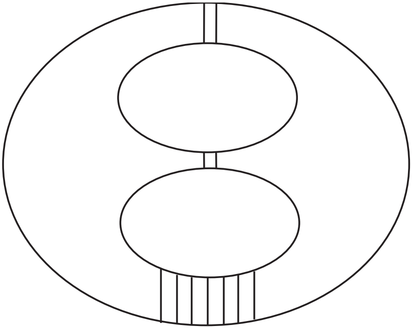
(iv) The second surface in the hierarchy

Example



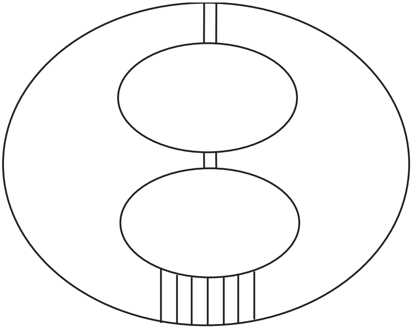
(v) The pattern of one of the balls

Example



(vi) A simplified copy of the pattern

Example



(vi) A simplified copy of the pattern

This is an essential boundary pattern, and hence the knot 5_2 is not the unknot

Simplifying inessential hierarchies

What if we have a hierarchy $M_1, \dots, M_{\ell+1}$ that is inessential?

Simplifying inessential hierarchies

What if we have a hierarchy $M_1, \ldots, M_{\ell+1}$ that is inessential?

Let D be a violating disc for $M_{\ell+1}$.

Simplifying inessential hierarchies

What if we have a hierarchy $M_1, \ldots, M_{\ell+1}$ that is inessential?

Let D be a violating disc for $M_{\ell+1}$.

Then D can be used to simplify the hierarchy.

Simplifying inessential hierarchies

What if we have a hierarchy $M_1, \dots, M_{\ell+1}$ that is inessential?

Let D be a violating disc for $M_{\ell+1}$.

Then D can be used to simplify the hierarchy.

Let S_j be the last surface that ∂D runs over. Then D can be used to compress or 'pattern-compress' S_j.

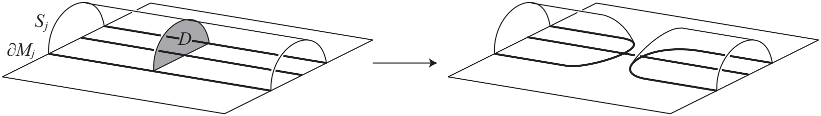
Simplifying inessential hierarchies

What if we have a hierarchy $M_1, \ldots, M_{\ell+1}$ that is inessential?

Let D be a violating disc for $M_{\ell+1}$.

Then D can be used to simplify the hierarchy.

Let S_j be the last surface that ∂D runs over. Then D can be used to compress or 'pattern-compress' S_j.



A new unknot recognition algorithm

Input: a diagram of a knot K

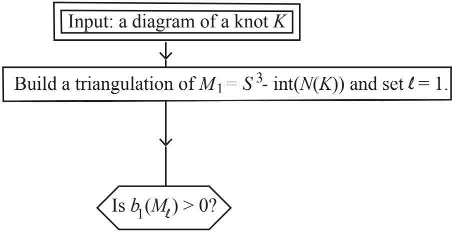
A new unknot recognition algorithm

Input: a diagram of a knot K

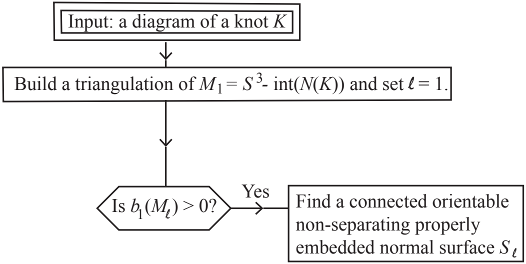
↓

Build a triangulation of $M_1 = S^3 - \text{int}(N(K))$ and set $\ell = 1$.

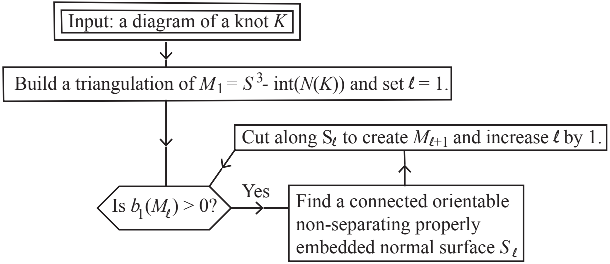
A new unknot recognition algorithm



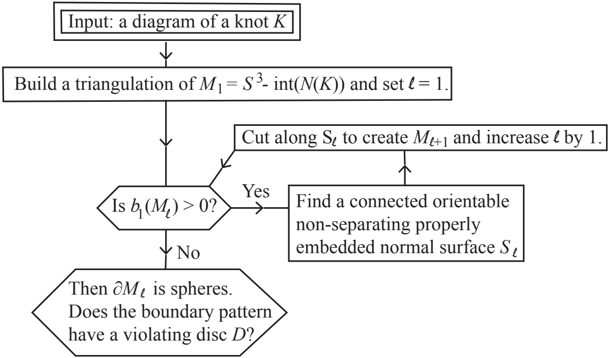
A new unknot recognition algorithm



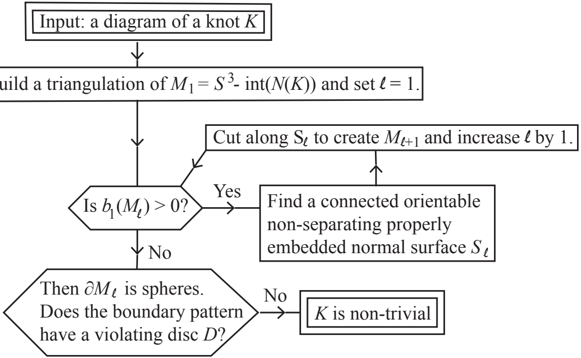
A new unknot recognition algorithm



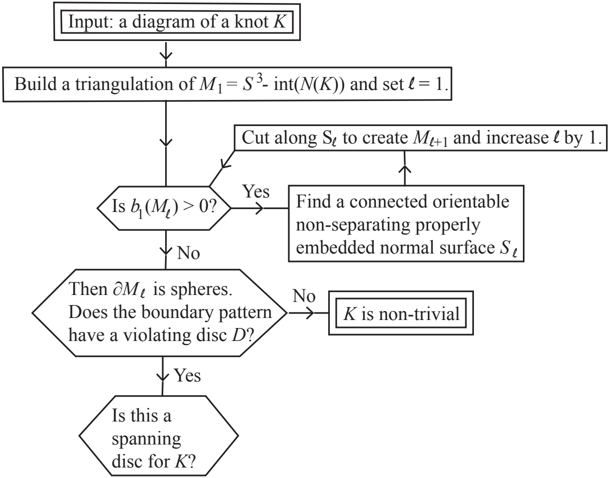
A new unknot recognition algorithm



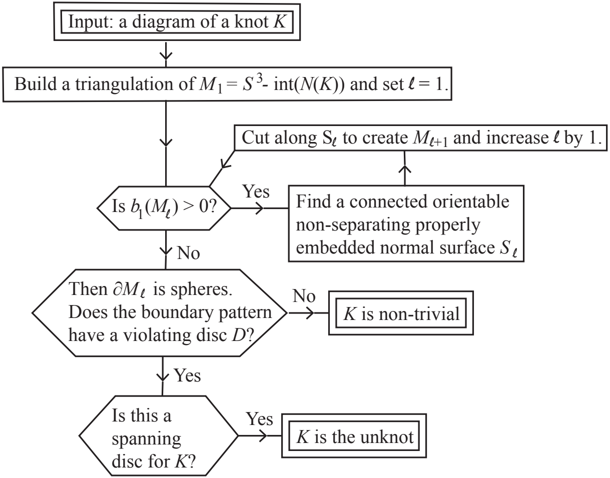
A new unknot recognition algorithm



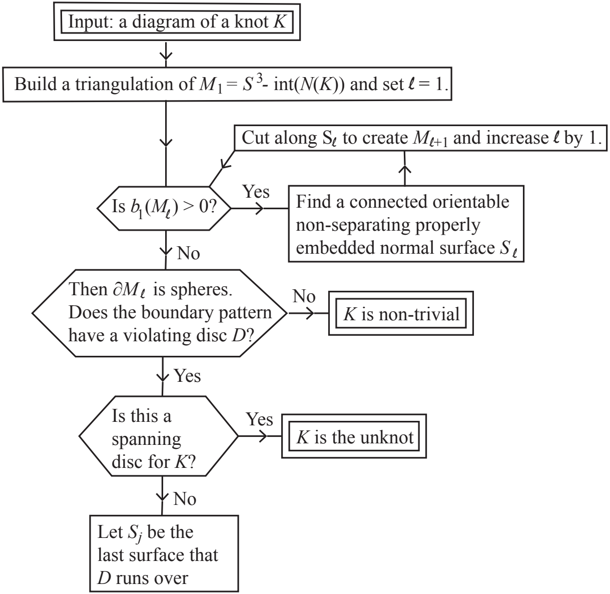
A new unknot recognition algorithm



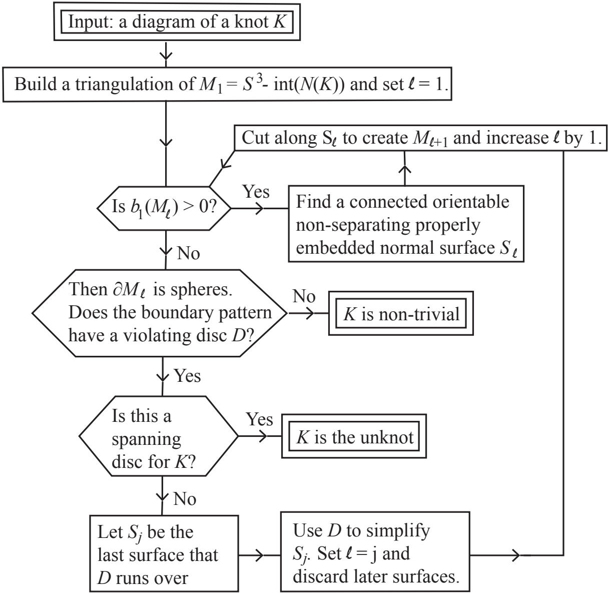
A new unknot recognition algorithm



A new unknot recognition algorithm



A new unknot recognition algorithm



Why does this terminate?

We cannot have an infinite sequence of decompositions along normal surfaces.

Why does this terminate?

We cannot have an infinite sequence of decompositions along normal surfaces.

So eventually we must end with a collection of 3-balls.

Why does this terminate?

We cannot have an infinite sequence of decompositions along normal surfaces.

So eventually we must end with a collection of 3-balls.

We can decide whether the pattern in the 3-balls is essential.

Why does this terminate?

We cannot have an infinite sequence of decompositions along normal surfaces.

So eventually we must end with a collection of 3-balls.

We can decide whether the pattern in the 3-balls is essential.

It it is, the knot is non-trivial.

Why does this terminate?

We cannot have an infinite sequence of decompositions along normal surfaces.

So eventually we must end with a collection of 3-balls.

We can decide whether the pattern in the 3-balls is essential.

It it is, the knot is non-trivial.

If there is a violating disc, the resulting compression or pattern compression reduces the complexity

$$(g(S_1), \ldots, g(S_\ell))$$

where we use lexicographical ordering.

Why does this terminate?

We cannot have an infinite sequence of decompositions along normal surfaces.

So eventually we must end with a collection of 3-balls.

We can decide whether the pattern in the 3-balls is essential.

It it is, the knot is non-trivial.

If there is a violating disc, the resulting compression or pattern compression reduces the complexity

$$(g(S_1), \dots, g(S_\ell))$$

where we use lexicographical ordering.

(Throughout the talk, I'll refer to the 'genus' $g(S_i)$ but I may mean some related notion, for example χ_- or a version that also counts intersections with the pattern.)

An estimate of running time

An estimate of running time

Suppose that we knew

- each surface S_i in the hierarchy had $g(S_i) \leq g$;
- the maximal length of the hierarchy was at most L.

An estimate of running time

Suppose that we knew

- each surface S_i in the hierarchy had $g(S_i) \leq g$;
- the maximal length of the hierarchy was at most L.

Then we could define the (g, L)-complexity of the hierarchy to be

$$\sum_{i=1}^{L} g(S_i) g^{L-i-1}.$$

An estimate of running time

Suppose that we knew

- each surface S_i in the hierarchy had $g(S_i) \leq g$;
- the maximal length of the hierarchy was at most L.

Then we could define the (g, L)-complexity of the hierarchy to be

$$\sum_{i=1}^{L} g(S_i) g^{L-i-1}.$$

(ie we would view $g(S_1), g(S_2), \ldots$ as a sequence of digits in base g).

An estimate of running time

Suppose that we knew

- each surface S_i in the hierarchy had $g(S_i) \leq g$;
- the maximal length of the hierarchy was at most L.

Then we could define the (g, L)-complexity of the hierarchy to be

$$\sum_{i=1}^{L} g(S_i) g^{L-i-1}.$$

(ie we would view $g(S_1), g(S_2), \ldots$ as a sequence of digits in base g).

Each time we simplify the hierarchy, its (g, L)-complexity decreases.

An estimate of running time

Suppose that we knew

- each surface S_i in the hierarchy had $g(S_i) \leq g$;
- the maximal length of the hierarchy was at most L.

Then we could define the (g, L)-complexity of the hierarchy to be

$$\sum_{i=1}^{L} g(S_i) g^{L-i-1}.$$

(ie we would view $g(S_1), g(S_2), \ldots$ as a sequence of digits in base g).

Each time we simplify the hierarchy, its (g, L)-complexity decreases.

So the 'running time' would be at most Lg^L.

Methods of speeding up the algorithm

The ‘running time’ is at most Lg^L.

Methods of speeding up the algorithm

The ‘running time’ is at most Lg^L.

Initial estimates only give $L \leq O(n)$ and $g \leq 2^{O(n)}$, where $n =$ the initial crossing number.

Methods of speeding up the algorithm

The 'running time' is at most Lg^L.

Initial estimates only give $L \leq O(n)$ and $g \leq 2^{O(n)}$, where $n =$ the initial crossing number.

We will give 4 methods of speeding up the algorithm:

Methods of speeding up the algorithm

The 'running time' is at most Lg^L.

Initial estimates only give $L \leq O(n)$ and $g \leq 2^{O(n)}$, where $n =$ the initial crossing number.

We will give 4 methods of speeding up the algorithm:

1. use surfaces with $g \leq O(n^2)$;

Methods of speeding up the algorithm

The ‘running time’ is at most Lg^L.

Initial estimates only give $L \leq O(n)$ and $g \leq 2^{O(n)}$, where $n =$ the initial crossing number.

We will give 4 methods of speeding up the algorithm:

1. use surfaces with $g \leq O(n^2)$;
2. encode hierarchies efficiently [Agol-Hass-Thurston];

Methods of speeding up the algorithm

The 'running time' is at most Lg^L.

Initial estimates only give $L \leq O(n)$ and $g \leq 2^{O(n)}$, where $n =$ the initial crossing number.

We will give 4 methods of speeding up the algorithm:

1. use surfaces with $g \leq O(n^2)$;
2. encode hierarchies efficiently [Agol-Hass-Thurston];
3. use 'multi-surfaces';

Methods of speeding up the algorithm

The 'running time' is at most Lg^L.

Initial estimates only give $L \leq O(n)$ and $g \leq 2^{O(n)}$, where $n =$ the initial crossing number.

We will give 4 methods of speeding up the algorithm:

1. use surfaces with $g \leq O(n^2)$;
2. encode hierarchies efficiently [Agol-Hass-Thurston];
3. use 'multi-surfaces';
4. use Heegaard splittings and 'Cheeger regions'.

Methods of speeding up the algorithm

The 'running time' is at most Lg^L.

Initial estimates only give $L \leq O(n)$ and $g \leq 2^{O(n)}$, where $n =$ the initial crossing number.

We will give 4 methods of speeding up the algorithm:

1. use surfaces with $g \leq O(n^2)$;
2. encode hierarchies efficiently [Agol-Hass-Thurston];
3. use 'multi-surfaces';
4. use Heegaard splittings and 'Cheeger regions'.

The effect of this is ensure that $L \leq O(\log n)$.

Methods of speeding up the algorithm

The ‘running time’ is at most Lg^L.

Initial estimates only give $L \leq O(n)$ and $g \leq 2^{O(n)}$, where $n =$ the initial crossing number.

We will give 4 methods of speeding up the algorithm:

1. use surfaces with $g \leq O(n^2)$;
2. encode hierarchies efficiently [Agol-Hass-Thurston];
3. use ‘multi-surfaces’;
4. use Heegaard splittings and ‘Cheeger regions’.

The effect of this is ensure that $L \leq O(\log n)$.

Hence, the running time is $n^{O(\log n)}$.

A quadratic bound on the genus of surfaces

A quadratic bound on the genus of surfaces

The first surface in the hierachy is a Seifert surface.

A quadratic bound on the genus of surfaces

The first surface in the hierachy is a Seifert surface.

Seifert's algorithm creates a surface S_1 with $g(S_1) \leq O(n)$.

A quadratic bound on the genus of surfaces

The first surface in the hierachy is a Seifert surface.

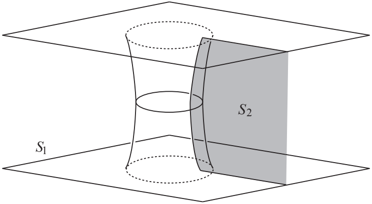
Seifert's algorithm creates a surface S_1 with $g(S_1) \leq O(n)$.

For the later surfaces in the hierarchy, we use a generalisation of Seifert's algorthm: we do not forget that our manifolds M_i lie in S^3.

What makes the algorithm inefficient?

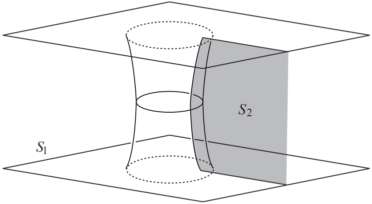
What makes the algorithm inefficient?

The problem is that if we compress a surface S_j, we have to discard the later surfaces.

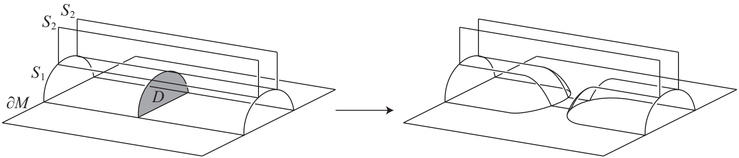


What makes the algorithm inefficient?

The problem is that if we compress a surface S_j, we have to discard the later surfaces.



This is not the case with boundary-compressions:



Multi-surfaces

Multi-surfaces

Instead of cutting along one surface at a time, we cut along a 'multi-surface'.

Multi-surfaces

Instead of cutting along one surface at a time, we cut along a ‘multi-surface’.

A multi-surface in a 3-manifold M is a collection of properly embedded oriented surfaces $S_1, \dots, S_k$ such that $[S_1], \dots, [S_k]$ represent linearly independent elements of $H_2(M, \partial M)$. Its rank is k.

Multi-surfaces

Instead of cutting along one surface at a time, we cut along a ‘multi-surface’.

A multi-surface in a 3-manifold M is a collection of properly embedded oriented surfaces $S_1, \ldots, S_k$ such that $[S_1], \ldots, [S_k]$ represent linearly independent elements of $H_2(M, \partial M)$. Its rank is k.

We can use a multi-surface $S_1, \ldots, S_k$ to create k steps in the hierarchy using the surfaces

$$S_1' = S_1,$$

$$S_2' = S_2 \backslash\backslash S_1,$$

$$S_3' = S_3 \backslash\backslash (S_1 \cup S_2) \ldots$$

Multi-surfaces

Instead of cutting along one surface at a time, we cut along a 'multi-surface'.

A multi-surface in a 3-manifold M is a collection of properly embedded oriented surfaces $S_1, \ldots, S_k$ such that $[S_1], \ldots, [S_k]$ represent linearly independent elements of $H_2(M, \partial M)$. Its rank is k.

We can use a multi-surface $S_1, \ldots, S_k$ to create k steps in the hierarchy using the surfaces

$$S'_1 = S_1,$$

$$S'_2 = S_2 \backslash\backslash S_1,$$

$$S'_3 = S_3 \backslash\backslash (S_1 \cup S_2) \ldots$$

Poincaré duality implies that a compact orientable 3-manifold M contains a multi-surface of rank at least $g(\partial M)$.

Compressing a multi-surface

Suppose that we boundary-compress some S'_j. Then we do not need to discard the rest of the hierarchy.

Compressing a multi-surface

Suppose that we boundary-compress some S_j'. Then we do not need to discard the rest of the hierarchy.

Suppose that we compress some S_j'. Then we can view this as a compression of S_j, and we simplify the multi-surface $S_1, \dots, S_k$.

Compressing a multi-surface

Suppose that we boundary-compress some S_j'. Then we do not need to discard the rest of the hierarchy.

Suppose that we compress some S_j'. Then we can view this as a compression of S_j, and we simplify the multi-surface $S_1, \ldots, S_k$.

So as far as our algorithm is concerned, a multi-surface behaves like a single surface.

Compressing a multi-surface

Suppose that we boundary-compress some S'_j. Then we do not need to discard the rest of the hierarchy.

Suppose that we compress some S'_j. Then we can view this as a compression of S_j, and we simplify the multi-surface $S_1, \dots, S_k$.

So as far as our algorithm is concerned, a multi-surface behaves like a single surface.

Our aim is to find a hierarchy of multi-surfaces of length $O(\log n)$.

Logarithmic length

We hope to find a hierarchy

$$M_1 \xrightarrow{\mathcal{S}_1} M_2 \xrightarrow{\mathcal{S}_2} \dots \xrightarrow{\mathcal{S}_\ell} M_{\ell+1}$$

where

- $\mathcal{S}_i$ is a multi-surface of rank $g(\partial M_i)$;
- $g(\partial M_i)$ grows exponentially as a function of i;

and so the hierarchy terminates after $O(\log n)$ step.

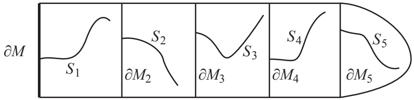
Long and thin manifolds

How might this fail?

Long and thin manifolds

How might this fail?

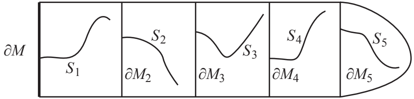
A problematic case is where $g(\partial M_i)$ is ‘small’ for each i:



Long and thin manifolds

How might this fail?

A problematic case is where $g(\partial M_i)$ is 'small' for each i:

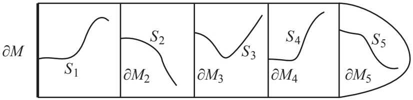


I examined such manifolds when I was investigating Cheeger constants some years ago.

Long and thin manifolds

How might this fail?

A problematic case is where $g(\partial M_i)$ is 'small' for each i:



I examined such manifolds when I was investigating Cheeger constants some years ago.

Recall that the Cheeger constant of a Riemannian n-manifold M is

$$\inf \left\{ \frac{\mathrm{Area}(\partial M')}{\min\{\mathrm{Vol}(M'), \mathrm{Vol}(M - M')\}} \right\}$$

as M' ranges over all n-dimensional submanifolds of M.

Measuring progress through the manifold

A hierarchy gives $M = M_1 \supset M_2 \supset \cdots \supset M_{\ell+1}$.

Measuring progress through the manifold

A hierarchy gives $M = M_1 \supset M_2 \supset \cdots \supset M_{\ell+1}$.

We measure the 'size' of M_i using Heegaard splittings.

Measuring progress through the manifold

A hierarchy gives $M = M_1 \supset M_2 \supset \cdots \supset M_{\ell+1}$.

We measure the 'size' of M_i using Heegaard splittings.

We work with a generalised Heegaard splitting on our manifold M arising from a Morse function h.

Measuring progress through the manifold

A hierarchy gives $M = M_1 \supset M_2 \supset \cdots \supset M_{\ell+1}$.

We measure the ‘size’ of M_i using Heegaard splittings.

We work with a generalised Heegaard splitting on our manifold M arising from a Morse function h.

Let’s consider the simple case where it is a Heegaard splitting with Heegaard surface H.

Measuring progress through the manifold

A hierarchy gives $M = M_1 \supset M_2 \supset \cdots \supset M_{\ell+1}$.

We measure the 'size' of M_i using Heegaard splittings.

We work with a generalised Heegaard splitting on our manifold M arising from a Morse function h.

Let's consider the simple case where it is a Heegaard splitting with Heegaard surface H.

The 'size' of M_i is $g(H \cap M_i)$.

Cheeger regions

Cheeger regions

We work with a generalised Heegaard splitting on our manifold M arising from a Morse function h.

Cheeger regions

We work with a generalised Heegaard splitting on our manifold M arising from a Morse function h.

At each stage of the hierarchy, we have a submanifold M_i of M.

Cheeger regions

We work with a generalised Heegaard splitting on our manifold M arising from a Morse function h.

At each stage of the hierarchy, we have a submanifold M_i of M.

We say that M_i is Cheeger region if

Cheeger regions

We work with a generalised Heegaard splitting on our manifold M arising from a Morse function h.

At each stage of the hierarchy, we have a submanifold M_i of M.

We say that M_i is Cheeger region if $h|M_i$ is a Heegaard Morse function with Heegaard surface H_i,

Cheeger regions

We work with a generalised Heegaard splitting on our manifold M arising from a Morse function h.

At each stage of the hierarchy, we have a submanifold M_i of M.

We say that M_i is Cheeger region if $h|M_i$ is a Heegaard Morse function with Heegaard surface H_i, and

$$g(\partial M_i) \leq (1/3)\min\{g(H_i), g(H) - g(H_i)\},$$

where H is the level of h containing H_i.

Cheeger regions

We work with a generalised Heegaard splitting on our manifold M arising from a Morse function h.

At each stage of the hierarchy, we have a submanifold M_i of M.

We say that M_i is Cheeger region if $h|M_i$ is a Heegaard Morse function with Heegaard surface H_i, and

$$g(\partial M_i) \leq (1/3) \min\{g(H_i), g(H) - g(H_i)\},$$

where H is the level of h containing H_i.

- If we come across a Cheeger region, we can simplify the initial generalised Heegaard splitting.

Cheeger regions

We work with a generalised Heegaard splitting on our manifold M arising from a Morse function h.

At each stage of the hierarchy, we have a submanifold M_i of M.

We say that M_i is Cheeger region if $h|M_i$ is a Heegaard Morse function with Heegaard surface H_i, and

$$g(\partial M_i) \leq (1/3) \min\{g(H_i), g(H) - g(H_i)\},$$

where H is the level of h containing H_i.

- If we come across a Cheeger region, we can simplify the initial generalised Heegaard splitting.
- If we never see a Cheeger region, the hierarchy completes in $O(\log n)$ steps.

The algorithm that runs in quasi-polynomial time