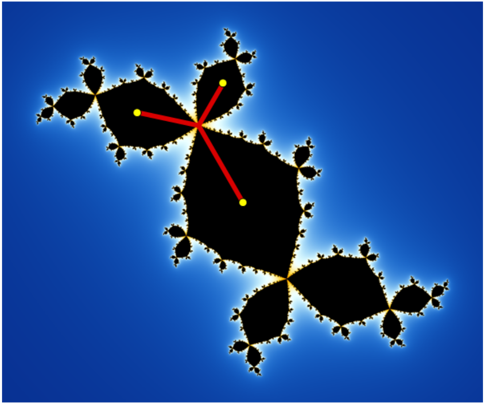
The Tree Lifting Algorithm



Jim Belk, University of St Andrews

Collaborators

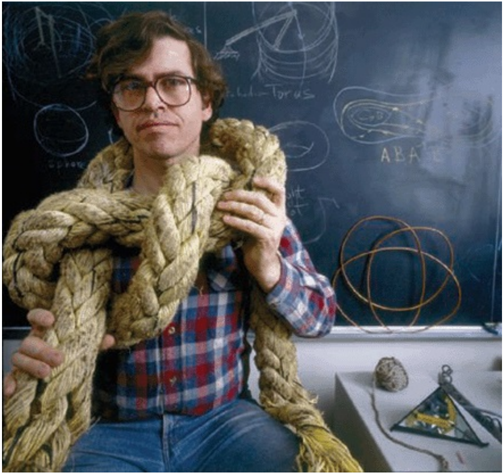
Justin Lanier,
Georgia Tech

Dan Margalit,
Georgia Tech

Becca Winarski,
U. Michigan

Topological Polynomials

In the 1980's, Bill Thurston began to study complex polynomials from a topological viewpoint.



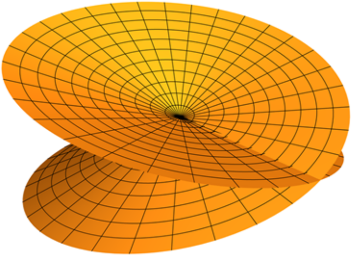
Topological Polynomials

In the 1980's, Bill Thurston began to study complex polynomials from a topological viewpoint.

A ***topological polynomial*** is any orientation-preserving branched cover

$$f : \mathbb{C} \to \mathbb{C}$$

with finitely many branch points.

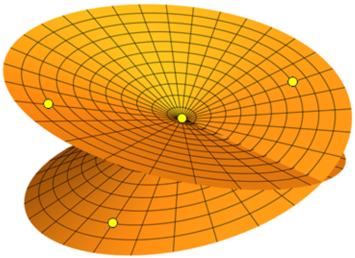


In analogy with polynomials, we refer to the branch points as ***critical points***, and their images as ***critical values***.

Marked Points

We can ***mark*** a topological polynomial by choosing a finite set $M \subset \mathbb{C}$, where

1. $f(M) \subset M$, and
2. M contains the critical values of f.



Marked Points

We can ***mark*** a topological polynomial by choosing a finite set $M \subset \mathbb{C}$, where

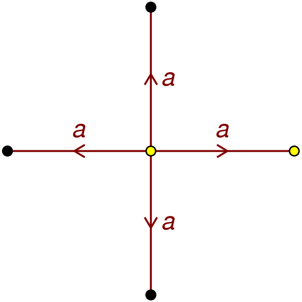
1. $f(M) \subset M$, and
2. M contains the critical values of f.

Basic Question: Which marked topological polynomials (f, M) are topologically equivalent to polynomials?

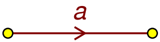
Alexander Method

We can specify (f, M) up to isotopy by drawing

1. Any tree T containing M, and
2. The mapping $f^{-1}(T) \to T$.



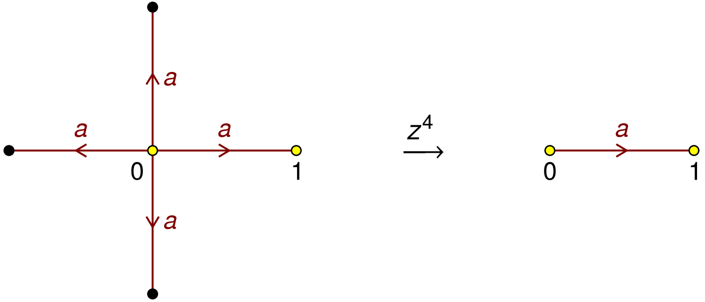
$\longrightarrow$



Alexander Method

We can specify (f, M) up to isotopy by drawing

1. Any tree T containing M, and
2. The mapping $f^{-1}(T) \to T$.



Alexander Method

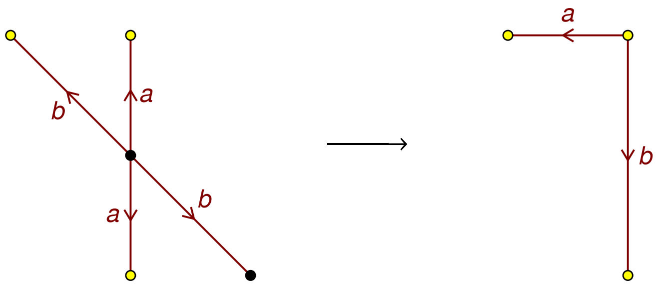
We can specify (f, M) up to isotopy by drawing

1. Any tree T containing M, and
2. The mapping $f^{-1}(T) \to T$.

Alexander Method

We can specify (f, M) up to isotopy by drawing

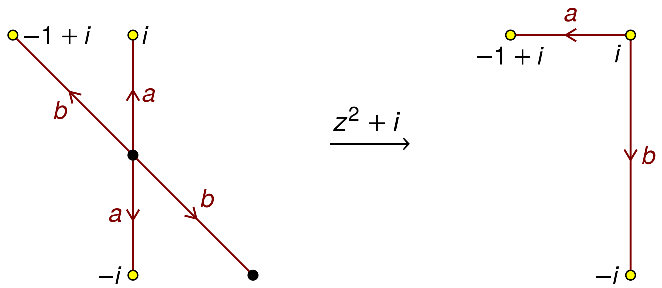
1. Any tree T containing M, and
2. The mapping $f^{-1}(T) \to T$.



Alexander Method

We can specify (f, M) up to isotopy by drawing

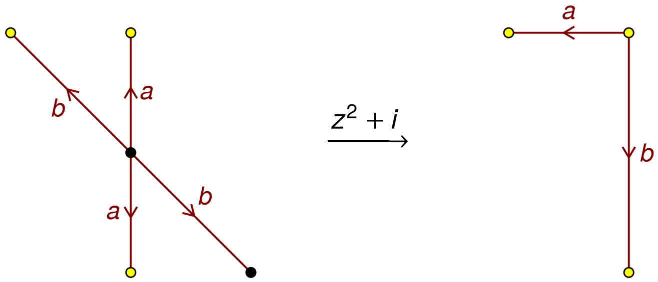
1. Any tree T containing M, and
2. The mapping $f^{-1}(T) \to T$.



Alexander Method

We can specify (f, M) up to isotopy by drawing

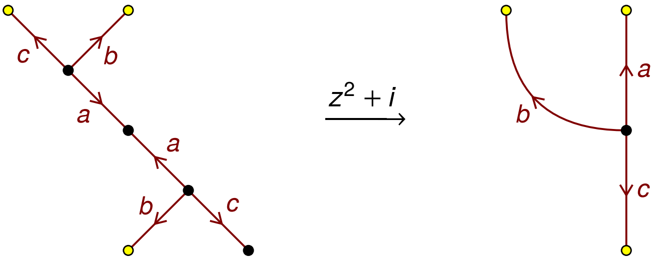
1. Any tree T containing M, and
2. The mapping $f^{-1}(T) \to T$.



Alexander Method

We can specify (f, M) up to isotopy by drawing

1. Any tree T containing M, and
2. The mapping $f^{-1}(T) \to T$.



Alexander Method

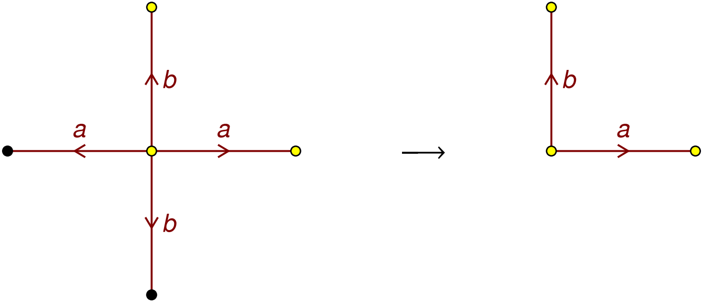
We can specify (f, M) up to isotopy by drawing

1. Any tree T containing M, and
2. The mapping $f^{-1}(T) \to T$.

Alexander Method

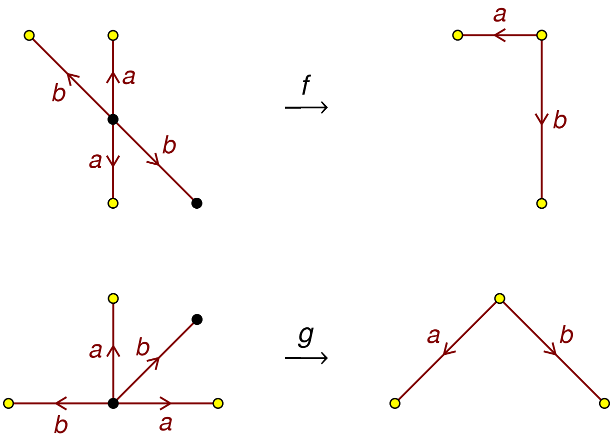
We can specify (f, M) up to isotopy by drawing

1. Any tree T containing M, and
2. The mapping $f^{-1}(T) \to T$.



Thurston Equivalence

Two marked topological polynomials are ***Thurston equivalent*** if there is a homeomorphism conjugating one to the other.



Thurston Equivalence

Two marked topological polynomials are ***Thurston equivalent*** if there is a homeomorphism conjugating one to the other.

$$\begin{tikzcd}
(\mathbb{C}, M) \arrow[r, "f"] \arrow[d, "h"'] & (\mathbb{C}, M) \arrow[d, "h"'] \\
(\mathbb{C}, N) \arrow[r, "g"] & (\mathbb{C}, N)
\end{tikzcd}$$

Thurston's Theorem

Theorem (W. Thurston, 1982)

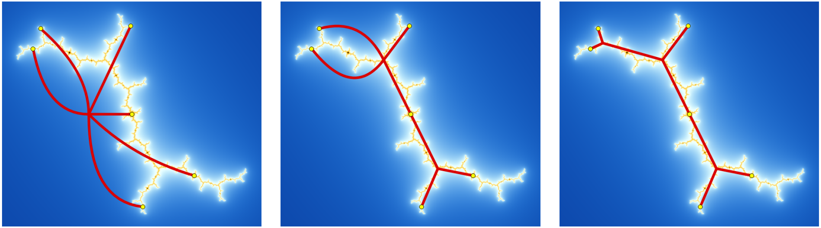
Let (f, M) be a marked topological polynomial. Then exactly one of the following holds:

1. (f, M) is Thurston equivalent to a polynomial, which is unique up to affine conjugacy.
2. (f, M) has a ***Thurston obstruction***.

This is an existence result only. It doesn't tell us how to find the polynomial (in case 1) or Thurston obstruction (in case 2).

Main Result

We have developed a simple geometric algorithm that solves these problems.



Given an (f, M), the algorithm produces either

1. The Hubbard tree for a polynomial equivalent to (f, M), or
2. The canonical Thurston obstruction for (f, M).

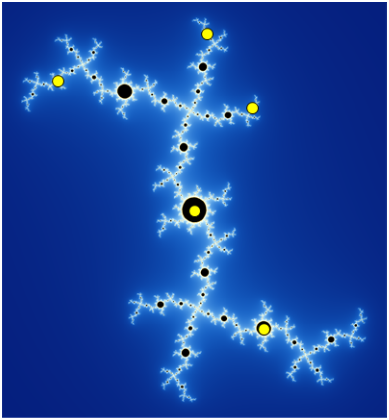
Lifting Trees

Goal: The Hubbard Tree

Every polynomial f (with marked set M) has a special tree called its ***Hubbard tree***.

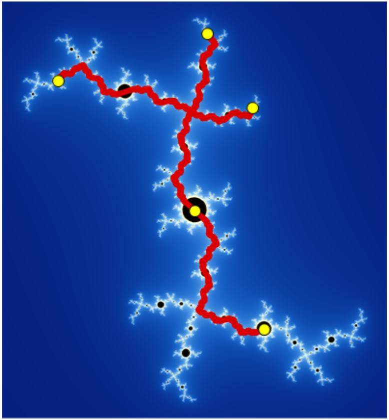
Goal: The Hubbard Tree

Every polynomial f (with marked set M) has a special tree called its ***Hubbard tree***.



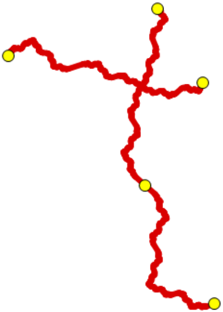
Goal: The Hubbard Tree

Every polynomial f (with marked set M) has a special tree called its ***Hubbard tree***.



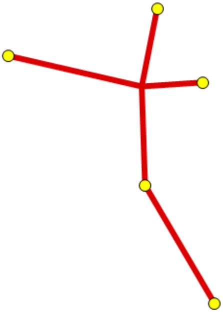
Goal: The Hubbard Tree

Every polynomial f (with marked set M) has a special tree called its ***Hubbard tree***.



Goal: The Hubbard Tree

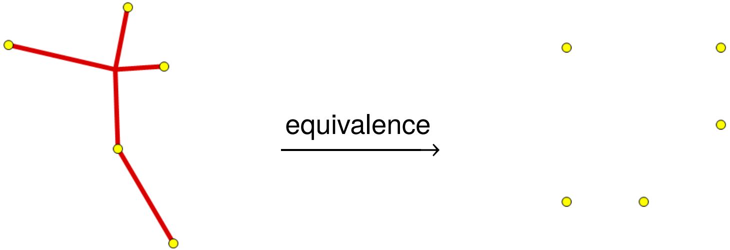
Every polynomial f (with marked set M) has a special tree called its ***Hubbard tree***.



Goal: The Hubbard Tree

Every polynomial f (with marked set M) has a special tree called its ***Hubbard tree***.

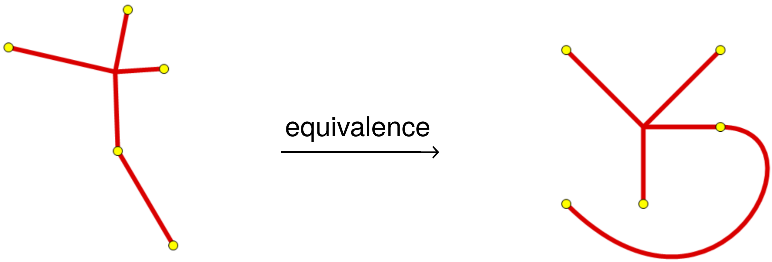
Any map that's Thurston equivalent to a polynomial has a ***topological Hubbard tree***.



Goal: The Hubbard Tree

Every polynomial f (with marked set M) has a special tree called its ***Hubbard tree***.

Any map that's Thurston equivalent to a polynomial has a ***topological Hubbard tree***.



Goal: The Hubbard Tree

Every polynomial f (with marked set M) has a special tree called its ***Hubbard tree***.

Any map that's Thurston equivalent to a polynomial has a ***topological Hubbard tree***.

Idea: Use topology to recover the topological Hubbard tree.

Goal: The Hubbard Tree

Every polynomial f (with marked set M) has a special tree called its ***Hubbard tree***.

Any map that's Thurston equivalent to a polynomial has a ***topological Hubbard tree***.

Idea: Use topology to recover the topological Hubbard tree.

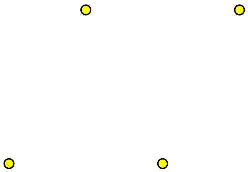
Note: Once the Hubbard tree is found, there are known algorithms (e.g. Hubbard–Schleicher) to recover the coefficients of f.

Trees in $(\mathbb{C}, M)$

We will consider trees in $(\mathbb{C}, M)$ satisfying the following conditions:

1. T contains M, and
2. Every leaf of T lies in M.

Isotopic trees are considered the same.

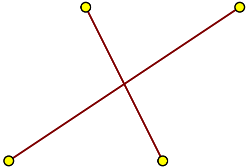


Trees in $(\mathbb{C}, M)$

We will consider trees in $(\mathbb{C}, M)$ satisfying the following conditions:

1. T contains M, and
2. Every leaf of T lies in M.

Isotopic trees are considered the same.

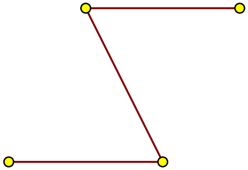


Trees in $(\mathbb{C}, M)$

We will consider trees in $(\mathbb{C}, M)$ satisfying the following conditions:

1. T contains M, and
2. Every leaf of T lies in M.

Isotopic trees are considered the same.

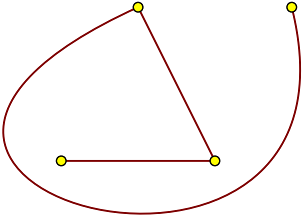


Trees in $(\mathbb{C}, M)$

We will consider trees in $(\mathbb{C}, M)$ satisfying the following conditions:

1. T contains M, and
2. Every leaf of T lies in M.

Isotopic trees are considered the same.

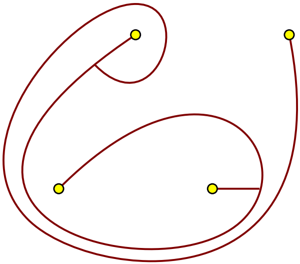


Trees in $(\mathbb{C}, M)$

We will consider trees in $(\mathbb{C}, M)$ satisfying the following conditions:

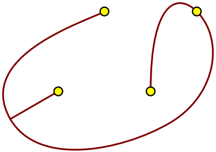
1. T contains M, and
2. Every leaf of T lies in M.

Isotopic trees are considered the same.

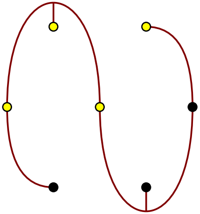


Lifting Trees

The preimage $f^{-1}(T)$ of a tree in $(\mathbb{C}, M)$ is not an allowed tree in $(\mathbb{C}, M)$.



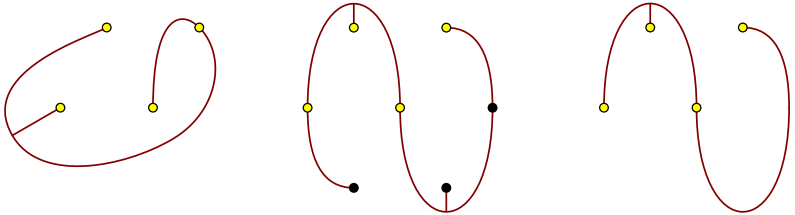
Tree T



preimage $f^{-1}(T)$

Lifting Trees

The preimage $f^{-1}(T)$ of a tree in $(\mathbb{C}, M)$ is not an allowed tree in $(\mathbb{C}, M)$.



Tree T | preimage $f^{-1}(T)$ | Lift $\lambda_f(T)$

The ***lift*** of T is the subtree of $f^{-1}(T)$ spanned by M.

Lifting Trees

Lifting under f defines a function

$$\lambda_f : \text{trees in } (\mathbb{C}, M) \longrightarrow \text{trees in } (\mathbb{C}, M)$$

Lifting Trees

Lifting under f defines a function

$$\lambda_f : \text{trees in } (\mathbb{C}, M) \longrightarrow \text{trees in } (\mathbb{C}, M)$$

Fact: The (topological) Hubbard tree is a fixed point for λ_f.

Lifting Trees

Lifting under f defines a function

$$\lambda_f : \text{trees in } (\mathbb{C}, M) \longrightarrow \text{trees in } (\mathbb{C}, M)$$

Fact: The (topological) Hubbard tree is a fixed point for λ_f.

This is because the Hubbard tree T satisfies

$$T \subset f^{-1}(T).$$

Lifting Trees

Lifting under f defines a function

$$\lambda_f : \text{trees in } (\mathbb{C}, M) \longrightarrow \text{trees in } (\mathbb{C}, M)$$

Fact: The (topological) Hubbard tree is a fixed point for λ_f.

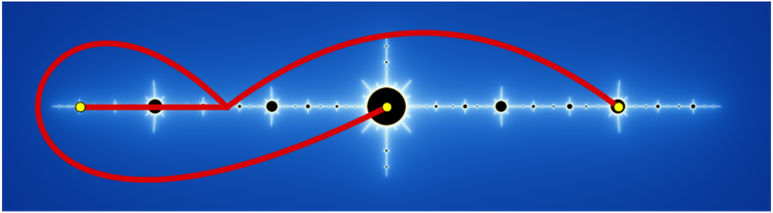
This is because the Hubbard tree T satisfies

$$T \subset f^{-1}(T).$$

Basic Algorithm: Iterate λ_f and hope to hit the Hubbard tree.

Iterated Lifting for the Airplane

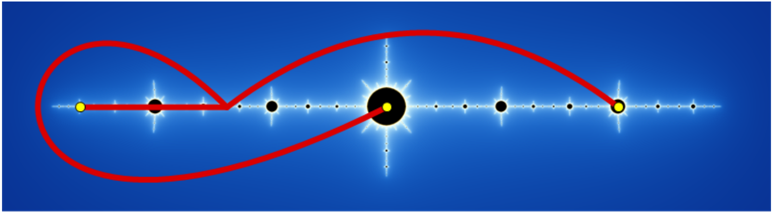
Let $f(z) \approx z^2 - 1.755$ be the airplane polynomial.



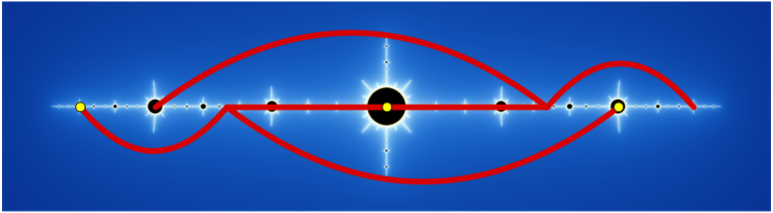
original tree T_0

Iterated Lifting for the Airplane

Let $f(z) \approx z^2 - 1.755$ be the airplane polynomial.



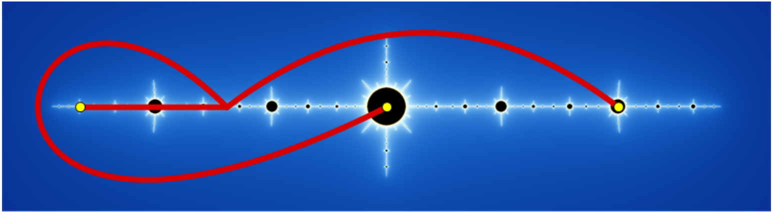
original tree T_0



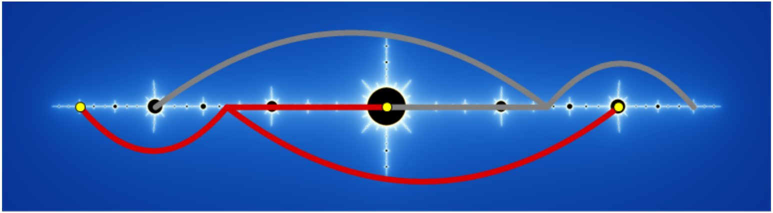
preimage $f^{-1}(T_0)$

Iterated Lifting for the Airplane

Let $f(z) \approx z^2 - 1.755$ be the airplane polynomial.



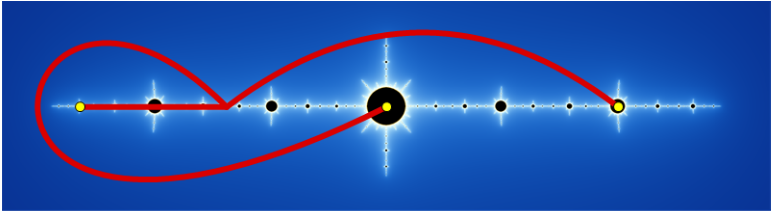
original tree T_0



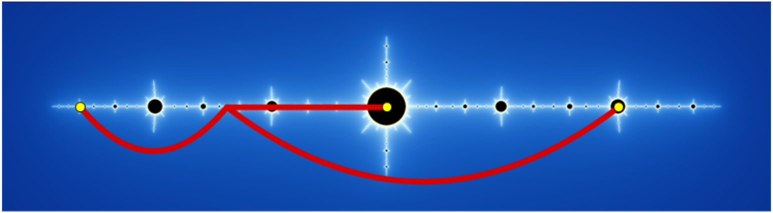
preimage $f^{-1}(T_0)$

Iterated Lifting for the Airplane

Let $f(z) \approx z^2 - 1.755$ be the airplane polynomial.



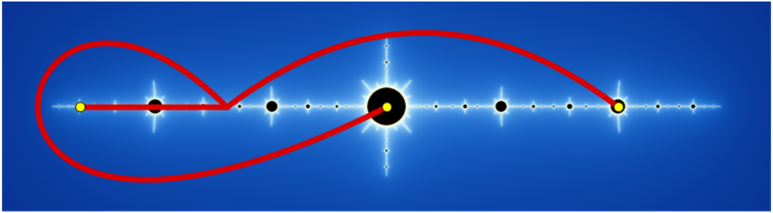
original tree T_0



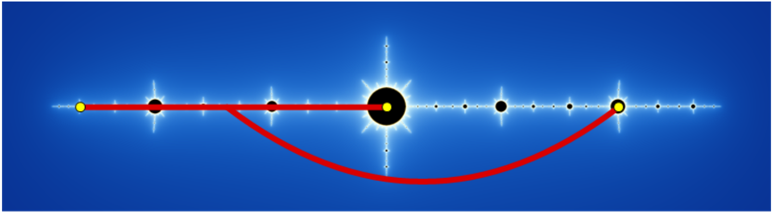
lift of T_0

Iterated Lifting for the Airplane

Let $f(z) \approx z^2 - 1.755$ be the airplane polynomial.



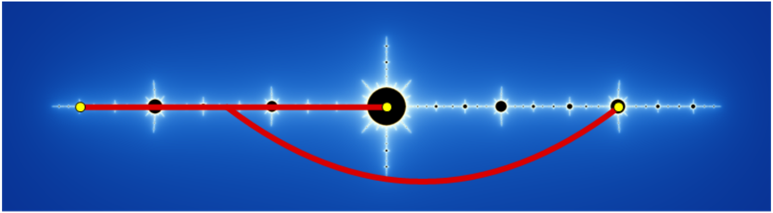
original tree T_0



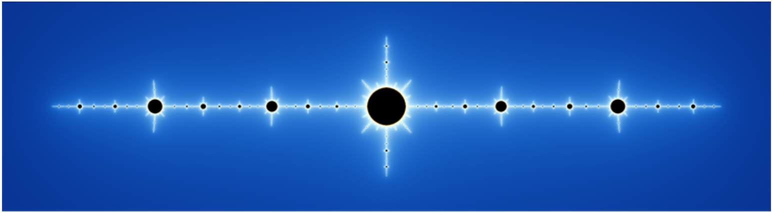
lift of T_0

Iterated Lifting for the Airplane

Let $f(z) \approx z^2 - 1.755$ be the airplane polynomial.

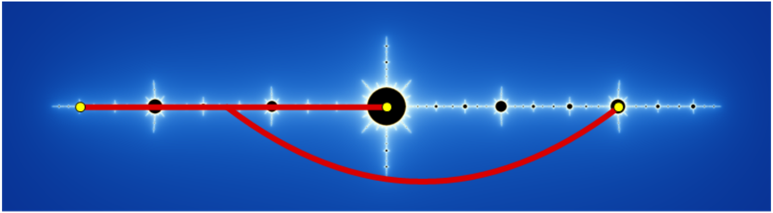


first lift T_1

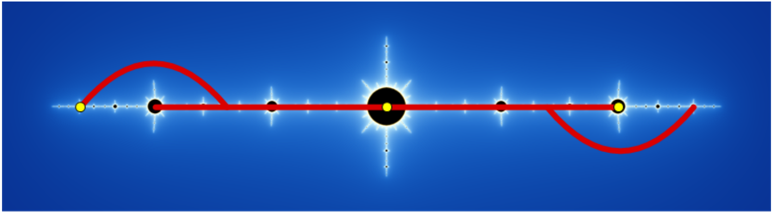


Iterated Lifting for the Airplane

Let $f(z) \approx z^2 - 1.755$ be the airplane polynomial.



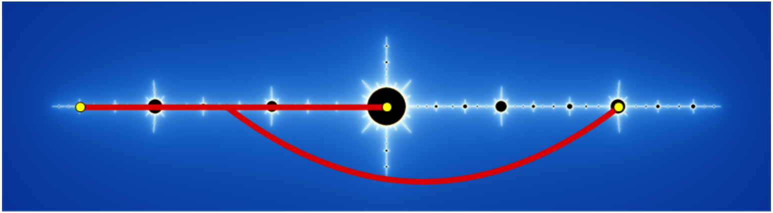
first lift T_1



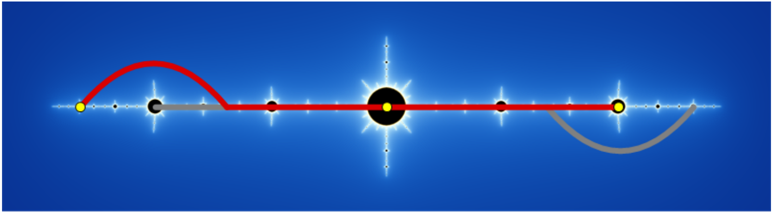
preimage $f^{-1}(T_1)$

Iterated Lifting for the Airplane

Let $f(z) \approx z^2 - 1.755$ be the airplane polynomial.



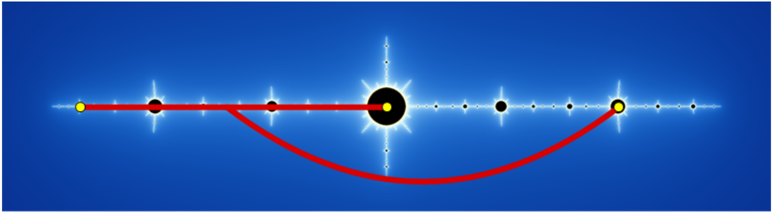
first lift T_1



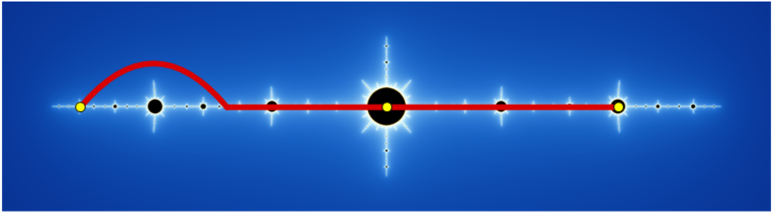
preimage $f^{-1}(T_1)$

Iterated Lifting for the Airplane

Let $f(z) \approx z^2 - 1.755$ be the airplane polynomial.



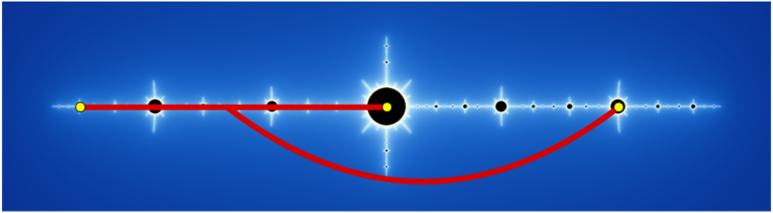
first lift T_1



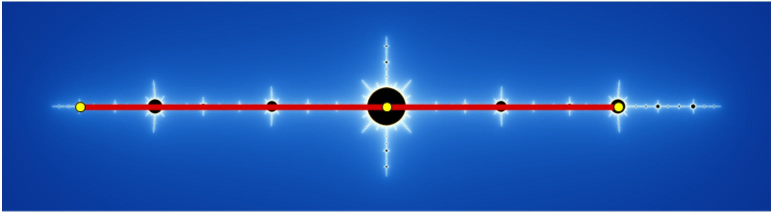
second lift T_2

Iterated Lifting for the Airplane

Let $f(z) \approx z^2 - 1.755$ be the airplane polynomial.



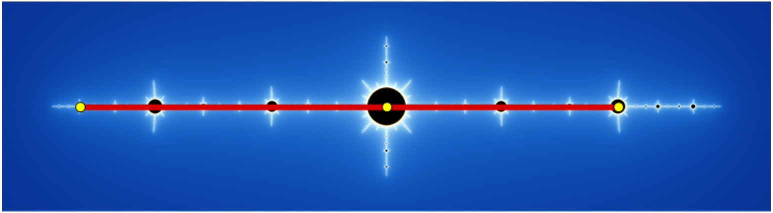
first lift T_1



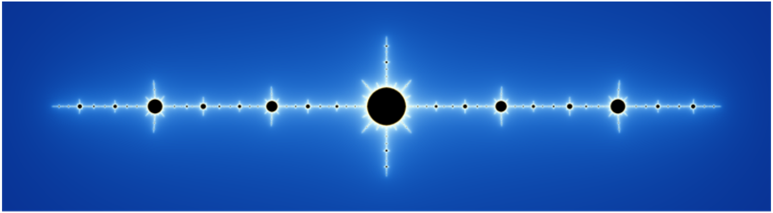
second lift T_2

Iterated Lifting for the Airplane

Let $f(z) \approx z^2 - 1.755$ be the airplane polynomial.

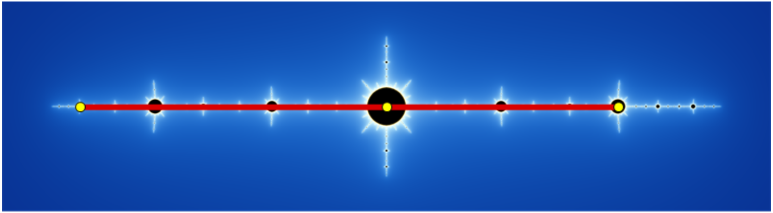


second lift T_2

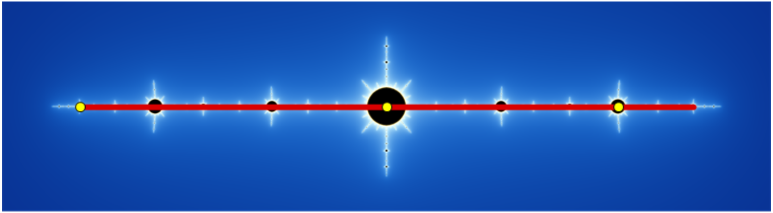


Iterated Lifting for the Airplane

Let $f(z) \approx z^2 - 1.755$ be the airplane polynomial.



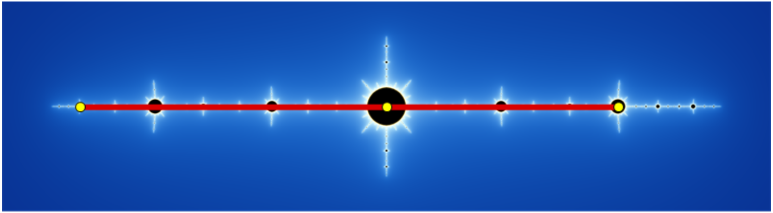
second lift T_2



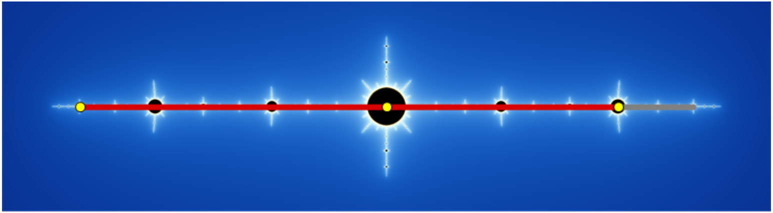
preimage $f^{-1}(T_2)$

Iterated Lifting for the Airplane

Let $f(z) \approx z^2 - 1.755$ be the airplane polynomial.



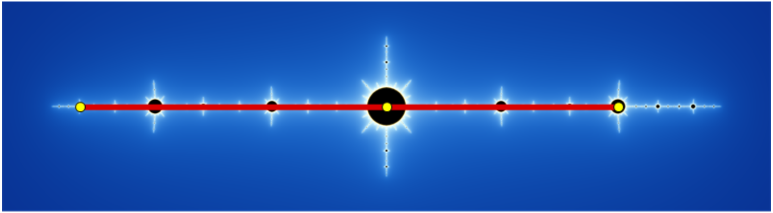
second lift T_2



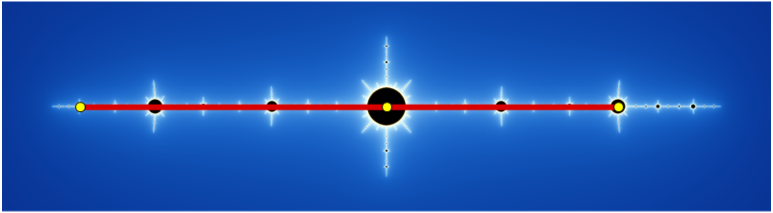
preimage $f^{-1}(T_2)$

Iterated Lifting for the Airplane

Let $f(z) \approx z^2 - 1.755$ be the airplane polynomial.

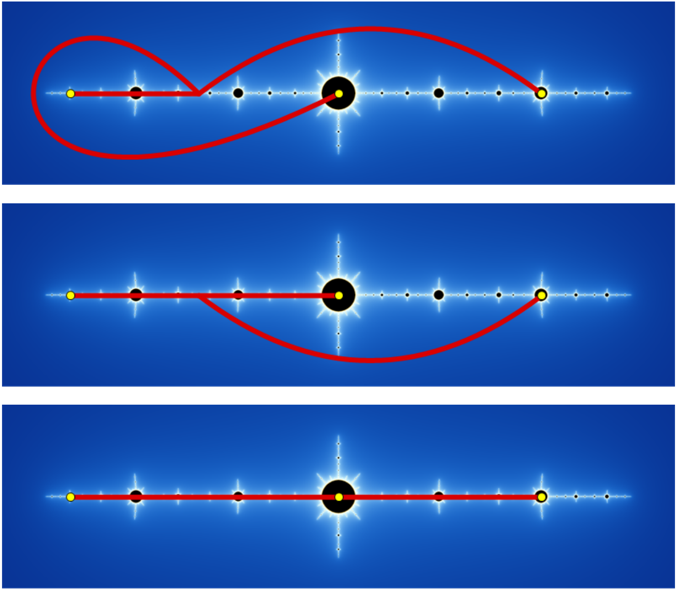


second lift T_2



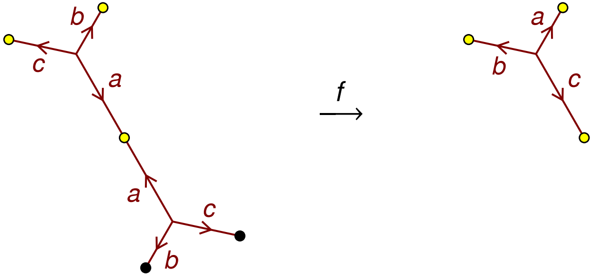
third lift T_3

Iterated Lifting for the Airplane



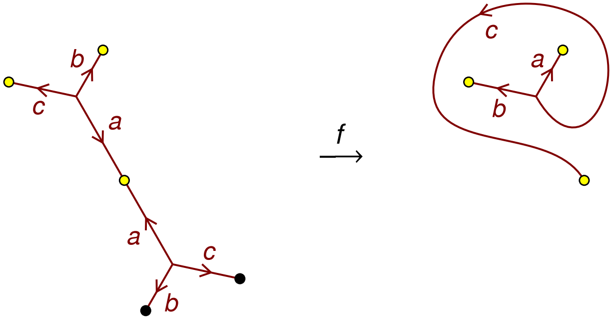
Example: A Twisted Rabbit

This is the ***rabbit polynomial*** $f(z) \approx z^2 - 0.12 + 0.74i$.

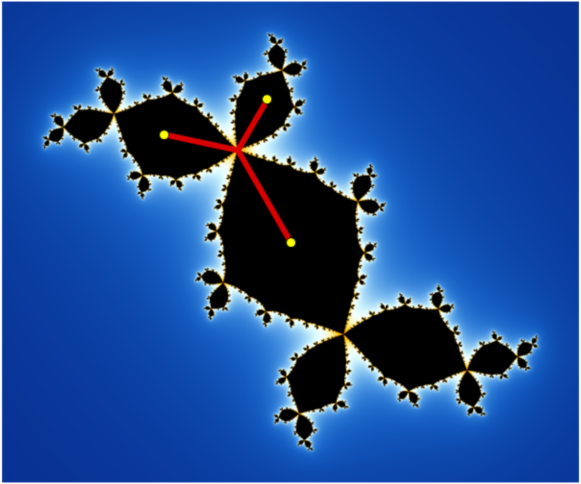


Example: A Twisted Rabbit

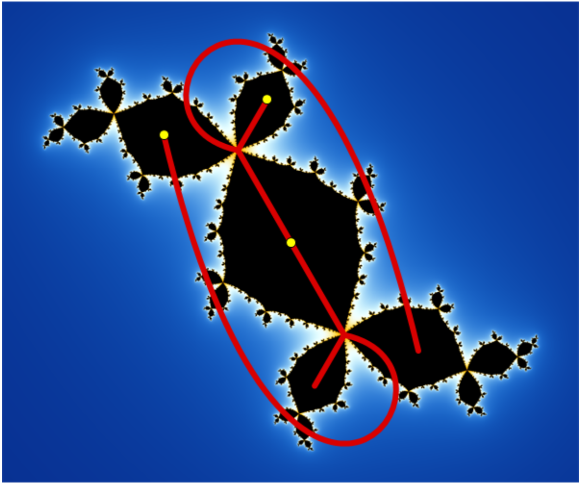
Composing with a Dehn twist gives a “twisted rabbit”.



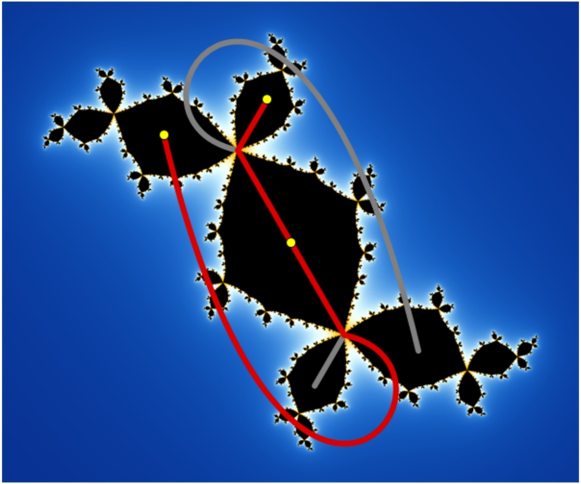
Example: A Twisted Rabbit



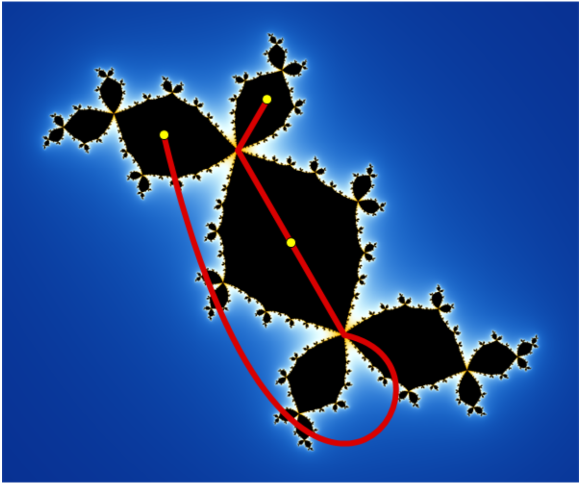
Example: A Twisted Rabbit



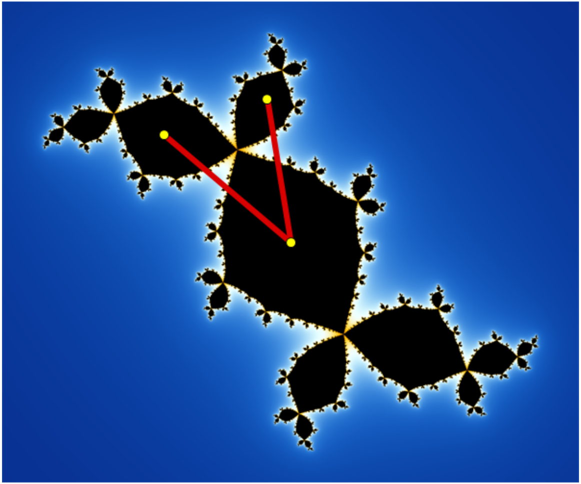
Example: A Twisted Rabbit



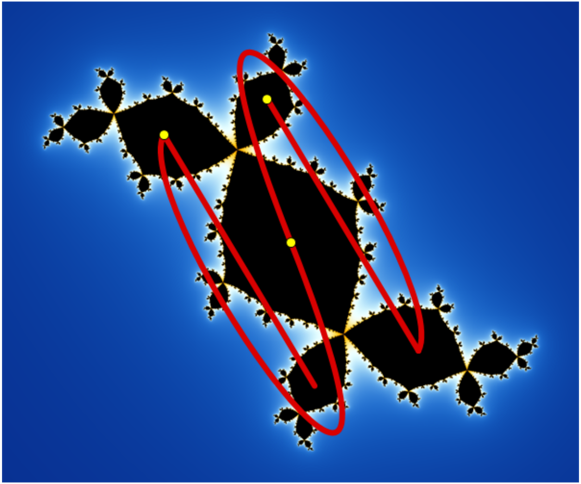
Example: A Twisted Rabbit



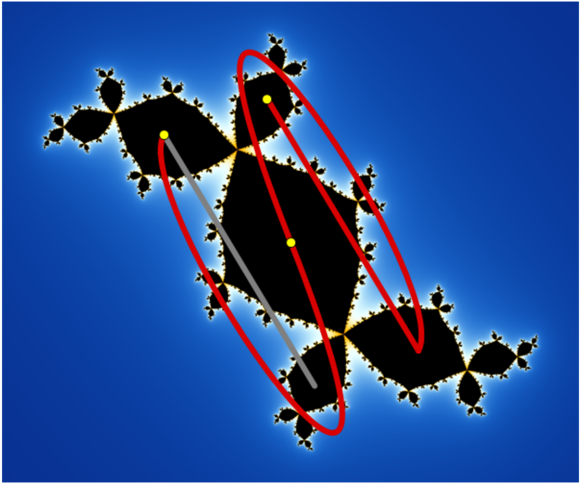
Example: A Twisted Rabbit



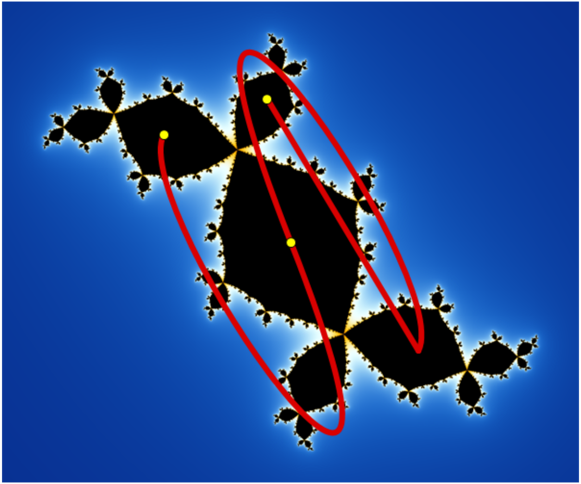
Example: A Twisted Rabbit



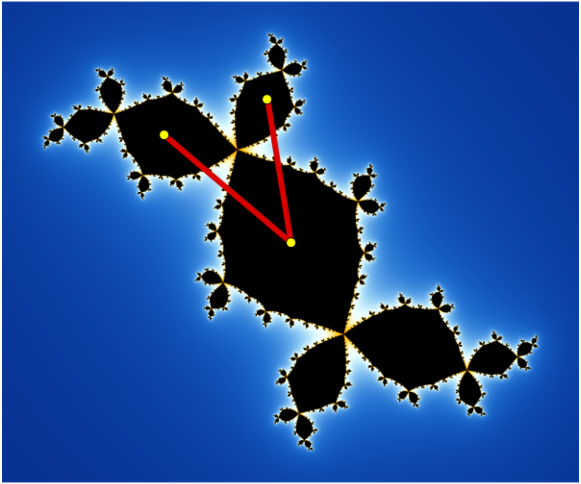
Example: A Twisted Rabbit



Example: A Twisted Rabbit

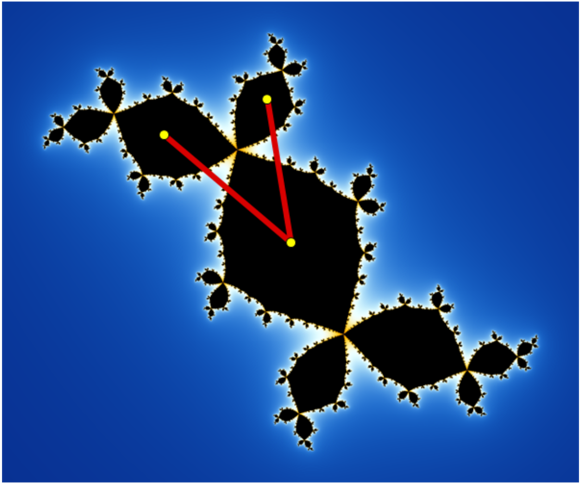


Example: A Twisted Rabbit



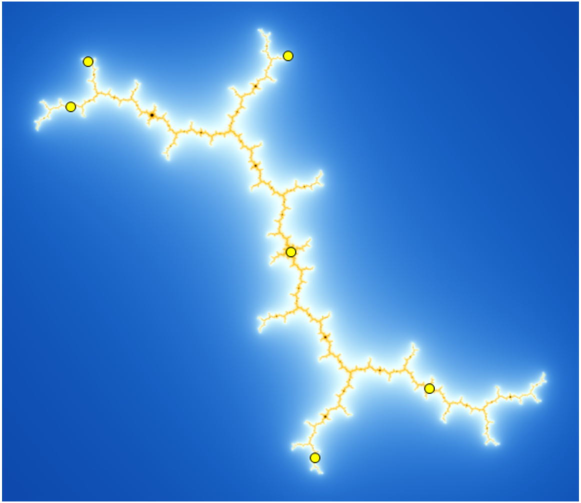
Example: A Twisted Rabbit

It's an airplane!



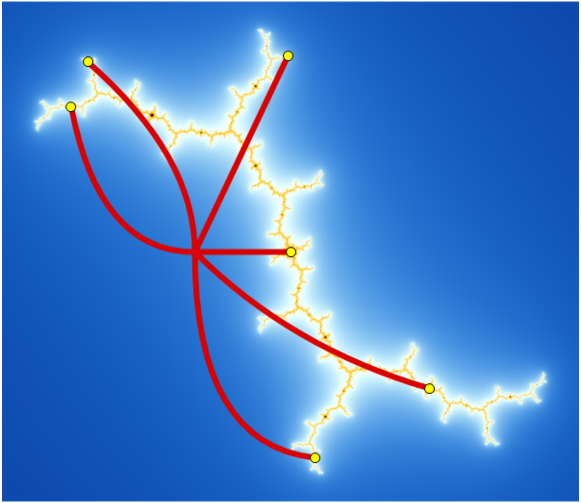
More Marked Points

Things don't get much harder with more marked points.



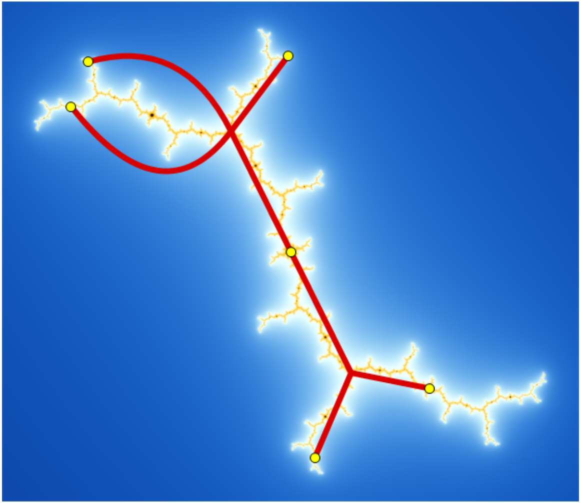
More Marked Points

Things don't get much harder with more marked points.



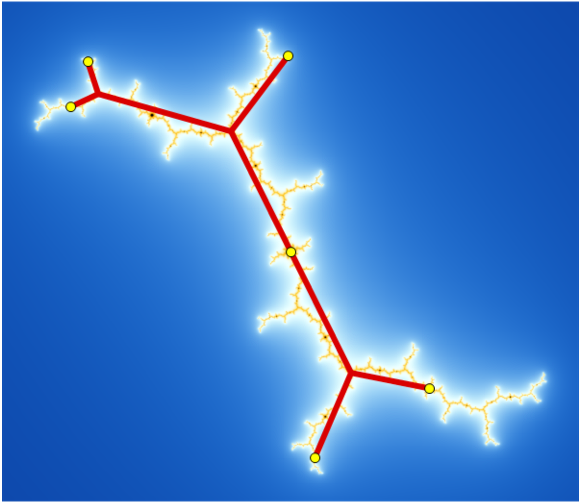
More Marked Points

Things don’t get much harder with more marked points.



More Marked Points

Things don't get much harder with more marked points.

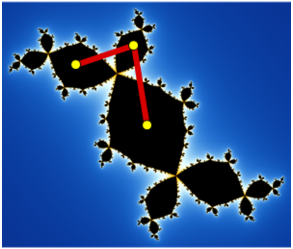


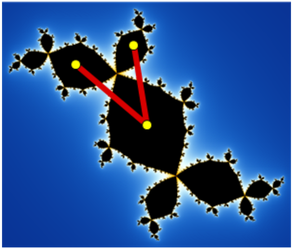
A Complication

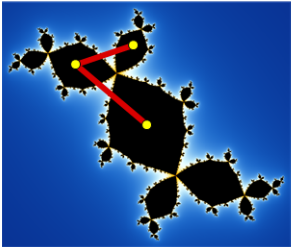
Unfortunately, you **don't** always hit the Hubbard tree.

A Complication

Unfortunately, you **don't** always hit the Hubbard tree.







A Complication

Unfortunately, you **don't** always hit the Hubbard tree.

Theorem (BLMW 2019)

Every marked polynomial has a finite ***nucleus*** of trees that are periodic under λ_f. Iterated lifting always lands in the nucleus.

A Complication

Unfortunately, you **don't** always hit the Hubbard tree.

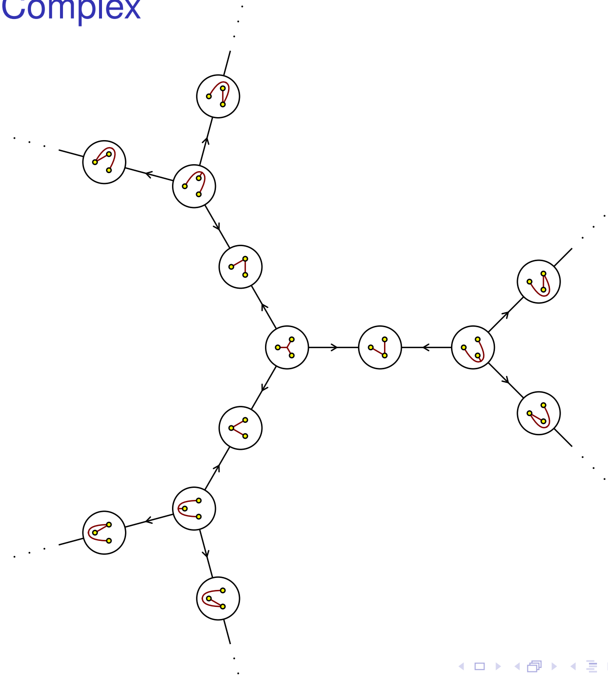
Theorem (BLMW 2019)

Every marked polynomial has a finite ***nucleus*** of trees that are periodic under λ_f. Iterated lifting always lands in the nucleus.

So the algorithm must include a resolution procedure to find the Hubbard tree once we land in the nucleus.

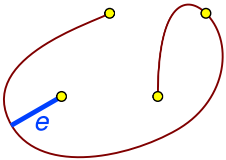
Dynamics of λ_f

The Tree Complex



Collapsing Subforests

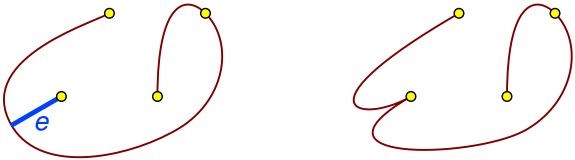
Let T be a tree in $(\mathbb{C}, M)$, and let e be an edge of T whose endpoints do not both lie in P.



tree T

Collapsing Subforests

Let T be a tree in $(\mathbb{C}, M)$, and let e be an edge of T whose endpoints do not both lie in P.

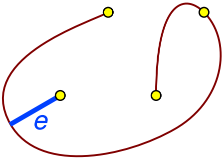


tree T $\qquad$ T/e

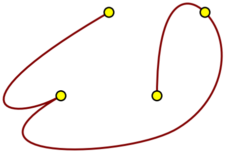
Then collapsing e to a point yields another tree T/e in $(\mathbb{C}, M)$.

Collapsing Subforests

Let T be a tree in $(\mathbb{C}, M)$, and let e be an edge of T whose endpoints do not both lie in P.



tree T



T/e

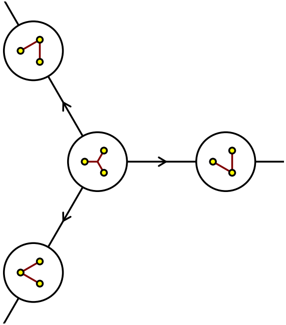
Then collapsing e to a point yields another tree T/e in $(\mathbb{C}, M)$.

More generally, we can collapse any subforest of T as long as no pair of marked points are identified.

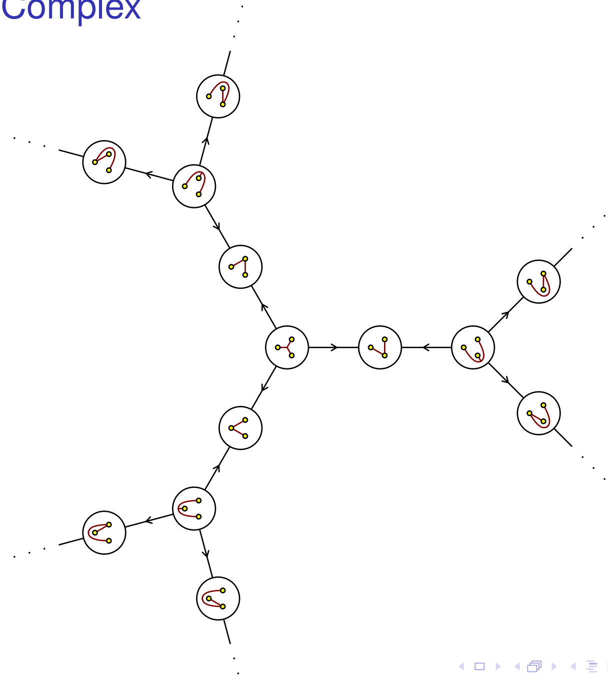
The Tree Complex

The ***tree complex*** has:

- One vertex for each tree in $(\mathbb{C}, M)$, and
- A directed edge $T \longrightarrow T'$ for each forest collapse.

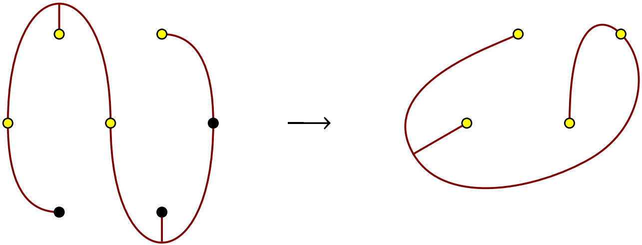


The Tree Complex



Lifting Forest Collapses

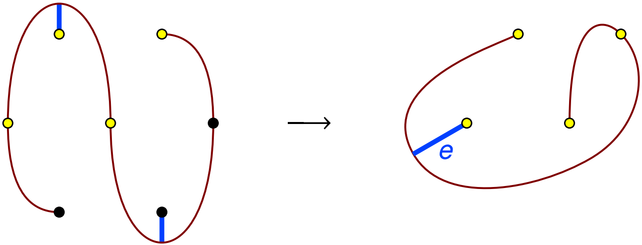
Any forest collapse $T \longrightarrow T'$ lifts to $f^{-1}(T)$.



preimage $f^{-1}(T)$ tree T

Lifting Forest Collapses

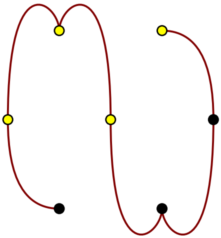
Any forest collapse $T \longrightarrow T'$ lifts to $f^{-1}(T)$.



preimage $f^{-1}(T)$ tree T

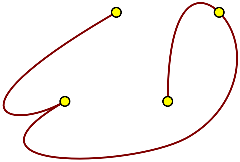
Lifting Forest Collapses

Any forest collapse $T \longrightarrow T'$ lifts to $f^{-1}(T)$.



preimage $f^{-1}(T')$

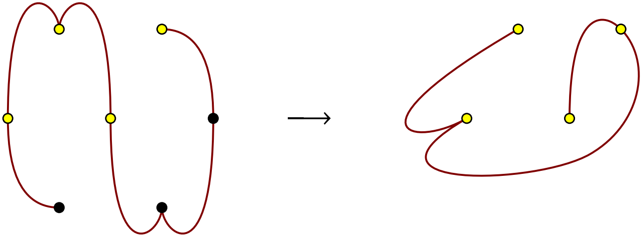
$\longrightarrow$



collapsed tree T'

Lifting Forest Collapses

Any forest collapse $T \longrightarrow T'$ lifts to $f^{-1}(T)$.



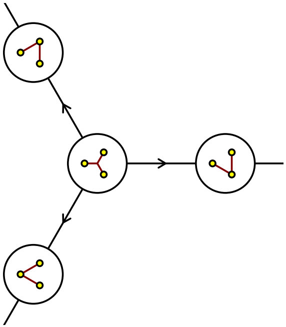
preimage $f^{-1}(T')$ collapsed tree T'

It follows that either

$$\lambda_f(T) \to \lambda_f(T') \qquad \text{or} \qquad \lambda_f(T) = \lambda_f(T').$$

The Tree Complex

So λ_f induces a non-expanding map on the tree complex. This is the ***lifting map***.



The Tree Complex

So λ_f induces a non-expanding map on the tree complex. This is the ***lifting map***.

Theorem (BLMW 2019)

If f is a polynomial, then every tree in $(\mathbb{C}, M)$ is either periodic or pre-periodic under λ_f.

The Tree Complex

So λ_f induces a non-expanding map on the tree complex. This is the ***lifting map***.

Theorem (BLMW 2019)

If f is a polynomial, then every tree in $(\mathbb{C}, M)$ is either periodic or pre-periodic under λ_f.

Proof.

Since the Hubbard tree T is fixed and λ_f is non-expanding, each ball in the complex centered at T maps into itself. Such a ball has finitely many trees. □

The Tree Complex

So λ_f induces a non-expanding map on the tree complex. This is the ***lifting map***.

Theorem (BLMW 2019)

If f is a polynomial, then every tree in $(\mathbb{C}, M)$ is either periodic or pre-periodic under λ_f.

The Tree Complex

So λ_f induces a non-expanding map on the tree complex. This is the ***lifting map***.

Theorem (BLMW 2019)

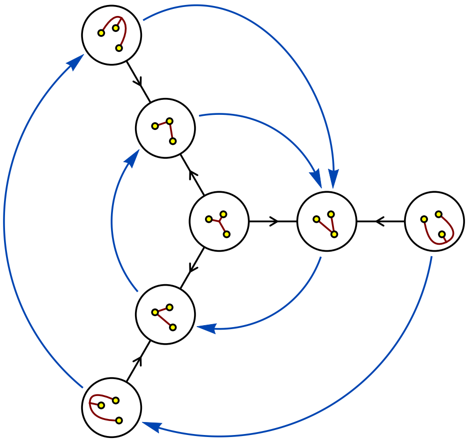
If f is a polynomial, then every tree in $(\mathbb{C}, M)$ is either periodic or pre-periodic under λ_f.

Theorem (BLMW 2019)

Every periodic tree lies in the ball of radius 2 centered at the Hubbard tree.

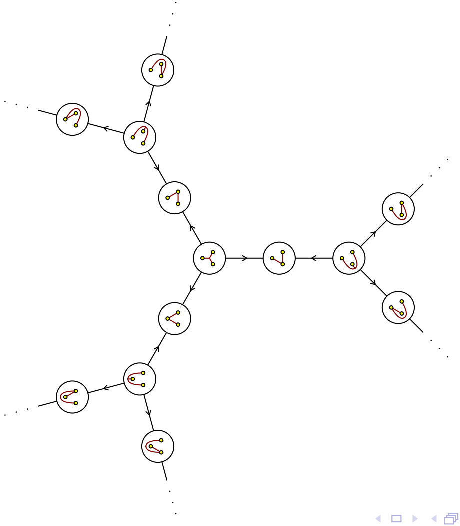
Example: The Rabbit Nucleus

The nucleus for the rabbit is the 1-neighborhood of the Hubbard tree.



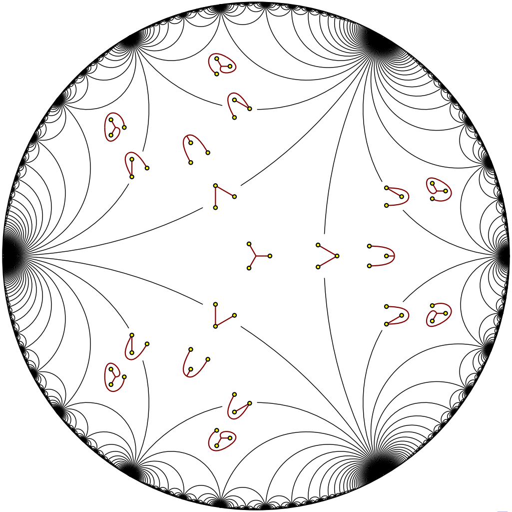
What's Going On?

The tree complex is actually the spine of a certain simplicial subdivision of Teichmüller space (discovered by Penner).



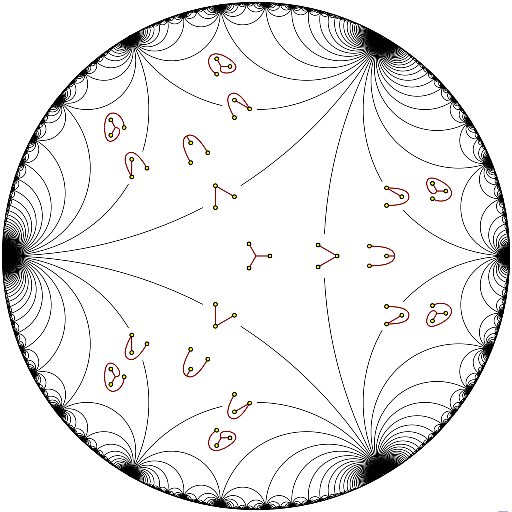
What's Going On?

The tree complex is actually the spine of a certain simplicial subdivision of Teichmüller space (discovered by Penner).



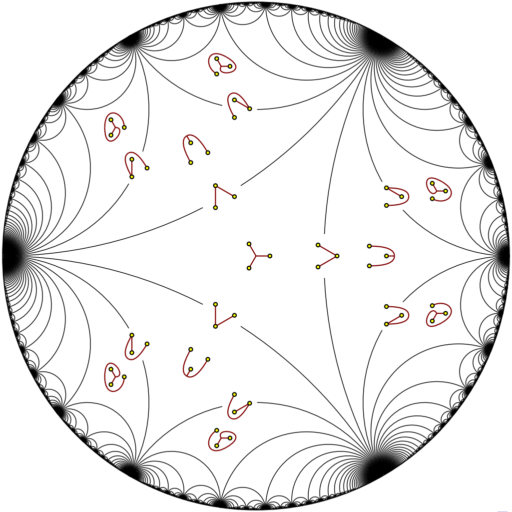
What's Going On?

Each tree corresponds to an open simplex. Different points in the simplex correspond to different metrics on the tree.



What's Going On?

The lifting map λ_f seems to be a combinatorial version of Thurston's pullback map $\sigma_f : \mathcal{T} \to \mathcal{T}$.



Finding the Hubbard Tree

The Story So Far

So far: We can iterate lifting until we find a periodic tree.

This gets us within 2 of the Hubbard tree.

Questions

1. How do we get to the Hubbard tree itself?
2. How would we even recognize the Hubbard tree if we found it?

Invariant Trees

A tree T in $(\mathbb{C}, M)$ is ***invariant*** if $\lambda_f(T) = T$. Up to isotopy, such a tree satisfies

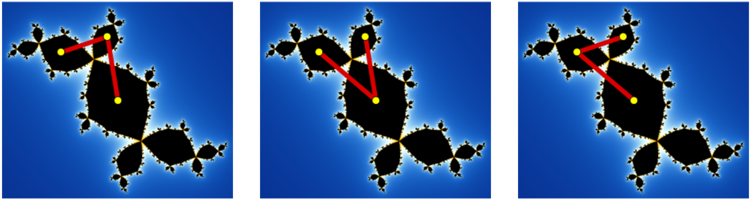
$$T \subset f^{-1}(T).$$

Invariant Trees

A tree T in $(\mathbb{C}, M)$ is ***invariant*** if $\lambda_f(T) = T$. Up to isotopy, such a tree satisfies

$$T \subset f^{-1}(T).$$

Note that periodic trees are invariant for f^k.



Invariant Trees

A tree T in $(\mathbb{C}, M)$ is ***invariant*** if $\lambda_f(T) = T$. Up to isotopy, such a tree satisfies

$$T \subset f^{-1}(T).$$

Note that periodic trees are invariant for f^k.

Question

How do we tell whether an invariant tree T is the Hubbard tree?

Invariant Trees

A tree T in $(\mathbb{C}, M)$ is ***invariant*** if $\lambda_f(T) = T$. Up to isotopy, such a tree satisfies

$$T \subset f^{-1}(T).$$

Note that periodic trees are invariant for f^k.

Question

How do we tell whether an invariant tree T is the Hubbard tree?

Answer

By the Alexander method, it suffices for there to exist *any* polynomial with Hubbard tree T (and corresponding preimage).

Poirier's Conditions

Alfredo Poirier completely classified possible Hubbard trees in 1993.

Theorem (Poirier's Conditions)

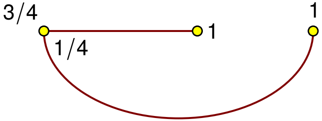
An invariant tree T for (f, M) is a topological Hubbard tree if and only if

1. **(Angle Condition)** T has an invariant angle assignment, and
2. **(Expanding Condition)** Every forward-invariant subforest of T contains a critical point.

The Angle Condition

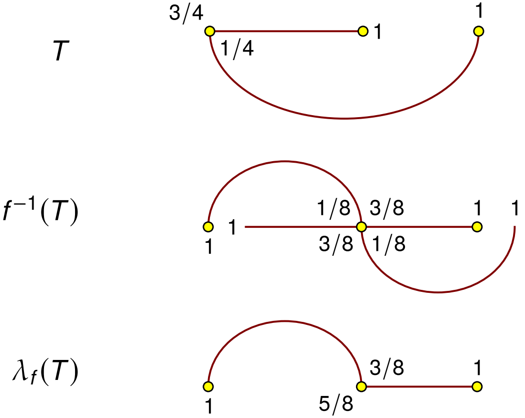
Here is an ***angle assignment*** for a tree T.

T



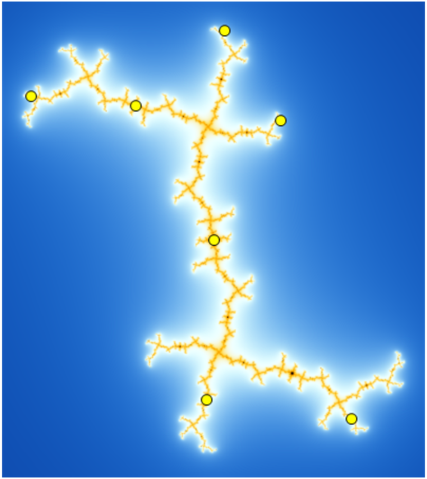
The Angle Condition

We can ***lift*** the angle assignment to $\lambda_f(T)$.



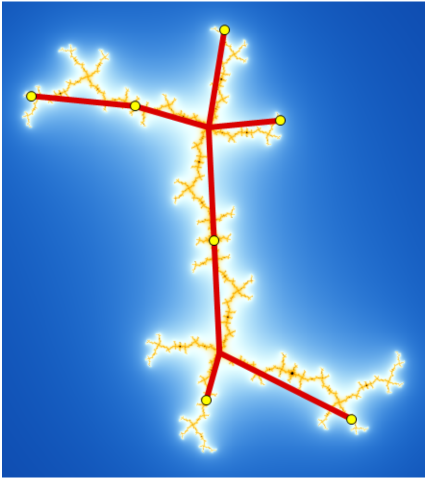
The Angle Condition

An invariant tree satisfies the ***angle condition*** if there exists an angle assignment that lifts to itself.



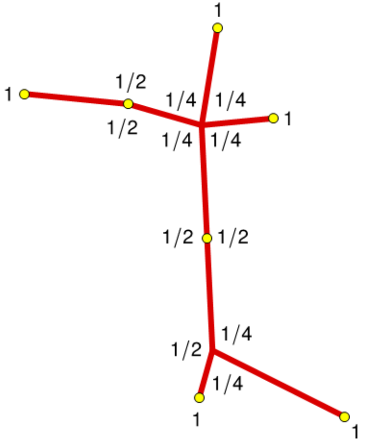
The Angle Condition

An invariant tree satisfies the ***angle condition*** if there exists an angle assignment that lifts to itself.



The Angle Condition

An invariant tree satisfies the ***angle condition*** if there exists an angle assignment that lifts to itself.



The Angle Condition

An invariant tree satisfies the ***angle condition*** if there exists an angle assignment that lifts to itself.

Theorem (BLMW 2019)

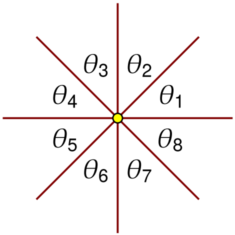
Every invariant tree is adjacent to an invariant tree that satisfies the angle condition.

The Angle Condition

An invariant tree satisfies the ***angle condition*** if there exists an angle assignment that lifts to itself.

Theorem (BLMW 2019)

Every invariant tree is adjacent to an invariant tree that satisfies the angle condition.

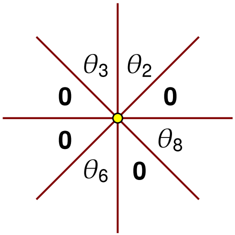


The Angle Condition

An invariant tree satisfies the ***angle condition*** if there exists an angle assignment that lifts to itself.

Theorem (BLMW 2019)

Every invariant tree is adjacent to an invariant tree that satisfies the angle condition.

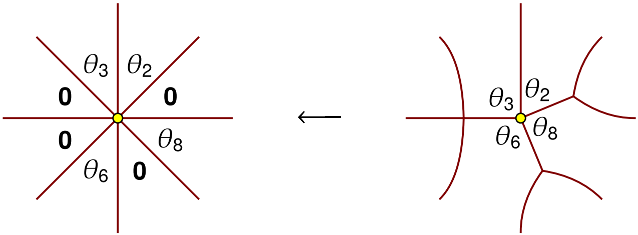


The Angle Condition

An invariant tree satisfies the ***angle condition*** if there exists an angle assignment that lifts to itself.

Theorem (BLMW 2019)

Every invariant tree is adjacent to an invariant tree that satisfies the angle condition.



The Expanding Condition

Let T be an invariant tree for (f, M).

A proper, nonempty subforest $S \subset T$ is ***forward invariant*** if $f(S) \subset S$.

We say that T satisfies the ***expanding condition*** if every forward invariant subforest of T contains a critical point.

Theorem (BLMW 2019)

Every invariant tree that satisfies the angle condition is adjacent to the Hubbard tree.

The Algorithm

So given an (f, M), the algorithm is as follows:

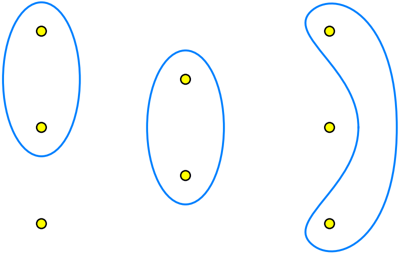
1. Start with any tree in $(\mathbb{C}, M)$ and iterate lifting until you find a periodic tree T.
2. Check if T satisfies the angle condition. If it doesn't, move to an adjacent tree T' that does.
3. Check if T' satisfies the expanding condition. If it doesn't, move to an adjacent tree T'' that does.

Then T'' is the topological Hubbard tree.

The Obstructed Case

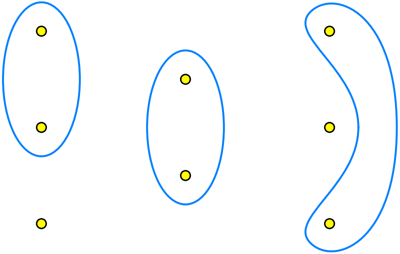
The Canonical Obstruction

Every obstructed (f, M) has a special collection of curves called the ***canonical obstruction***.



The Canonical Obstruction

Every obstructed (f, M) has a special collection of curves called the ***canonical obstruction***.

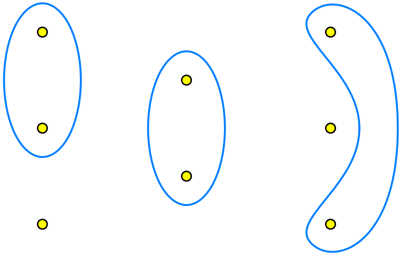


These are the curves whose hyperbolic lengths go to zero.

Pilgrim (2001) proved that the canonical obstruction is fully invariant under f, and is a Thurston obstruction.

The Canonical Obstruction

Every obstructed (f, M) has a special collection of curves called the ***canonical obstruction***.

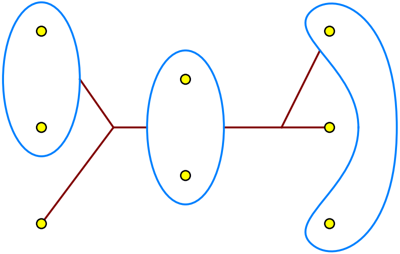


The curves of the canonical obstruction bound disjoint disks.

Selinger (2013) proved that the map on the exterior is Thurston equivalent to a polynomial.

The Canonical Obstruction

Every obstructed (f, M) has a special collection of curves called the ***canonical obstruction***.

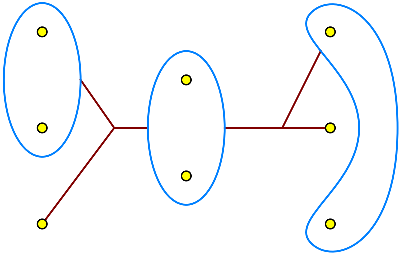


The curves of the canonical obstruction bound disjoint disks.

Selinger (2013) proved that the map on the exterior is Thurston equivalent to a polynomial.

The Canonical Obstruction

Every obstructed (f, M) has a special collection of curves called the ***canonical obstruction***.



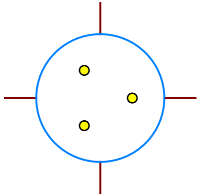
We call this the ***Hubbard bubble tree*** for the obstructed map.

When (f, M) is obstructed, we can use the tree lifting algorithm to find the Hubbard bubble tree.

Normal Form

Incidentally, each bubble has:

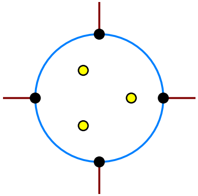
- Points of M in the interior, and



Normal Form

Incidentally, each bubble has:

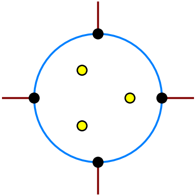
- Points of M in the interior, and
- Points on the boundary where it touches the tree.



Normal Form

Incidentally, each bubble has:

- Points of M in the interior, and
- Points on the boundary where it touches the tree.

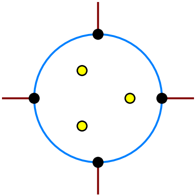


So we can think of each bubble as a marked disk.

Normal Form

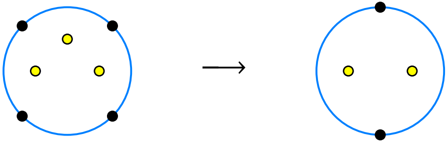
Incidentally, each bubble has:

- Points of M in the interior, and
- Points on the boundary where it touches the tree.



So we can think of each bubble as a marked disk.

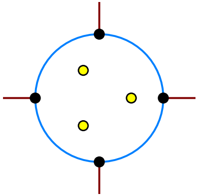
Maps between bubbles are homeomorphisms and branched covers that send marked points to marked points.



Normal Form

Incidentally, each bubble has:

- Points of M in the interior, and
- Points on the boundary where it touches the tree.



So we can think of each bubble as a marked disk.

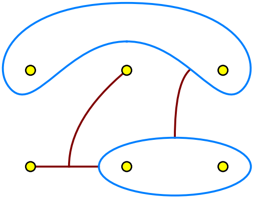
Maps between bubbles are homeomorphisms and branched covers that send marked points to marked points.

The Hubbard bubble tree together with these maps is a complete description of (f, M) up to isotopy. We call it the ***normal form***.

Finding the Hubbard Bubble Tree

In general, a ***bubble tree*** consists of:

1. Finitely many essential curves in $(\mathbb{C}, M)$ with disjoint interiors.
2. A tree on the exterior of these curves.

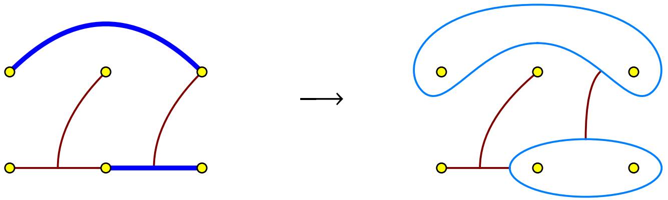


Finding the Hubbard Bubble Tree

In general, a ***bubble tree*** consists of:

1. Finitely many essential curves in $(\mathbb{C}, M)$ with disjoint interiors.
2. A tree on the exterior of these curves.

Bubble trees can be obtained from trees by collapsing subforests.



Finding the Hubbard Bubble Tree

In general, a ***bubble tree*** consists of:

1. Finitely many essential curves in $(\mathbb{C}, M)$ with disjoint interiors.
2. A tree on the exterior of these curves.

Bubble trees can be obtained from trees by collapsing subforests.

This defines the ***augmented tree complex***.

Finding the Hubbard Bubble Tree

In general, a ***bubble tree*** consists of:

1. Finitely many essential curves in $(\mathbb{C}, M)$ with disjoint interiors.
2. A tree on the exterior of these curves.

Bubble trees can be obtained from trees by collapsing subforests. This defines the ***augmented tree complex***.

Theorem (BLMW 2019)

For an obstructed (f, M), the sequence of lifts eventually lands in the 2-neighborhood of the Hubbard bubble tree in the augmented complex.

The End