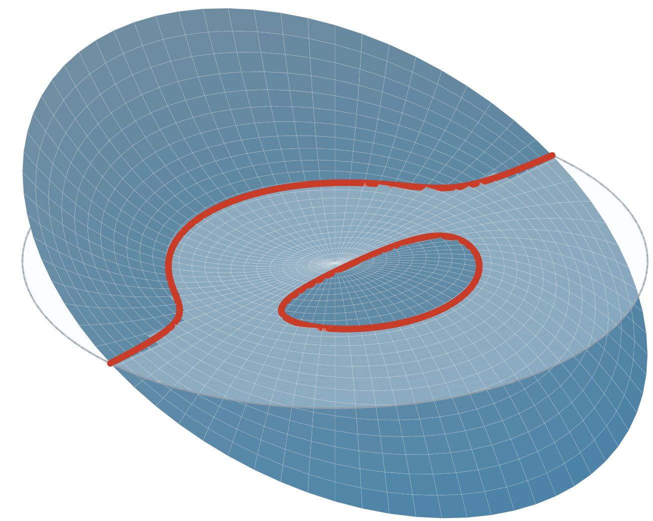
Fast Isotopy Computation for T-Curves

DOxML 2026 · Institute of Science Tokyo

13 July 2026



Z. Geiselmann, M. Joswig, L. Kastner, K. Mundinger,
S. Pokutta, **C. Spiegel**, M. Wack, M. Zimmer

ZUSE INSTITUTE BERLIN

Results are joint work with ...

Michael Joswig
TU Berlin / MPI-MiS

Lars Kastner
TU Berlin

Marcel Wack
TU Berlin

Zoe Geiselmann
TU Berlin

Discrete Mathematics & Geometry · TU Berlin

Sebastian Pokutta
ZIB / TU Berlin

Konrad Mundinger
ZIB / TU Berlin

Max Zimmer
ZIB / TU Berlin

Interactive Optimization & Learning · ZIB

The trivial topology of $\mathbb{R}[x]$

QUESTION

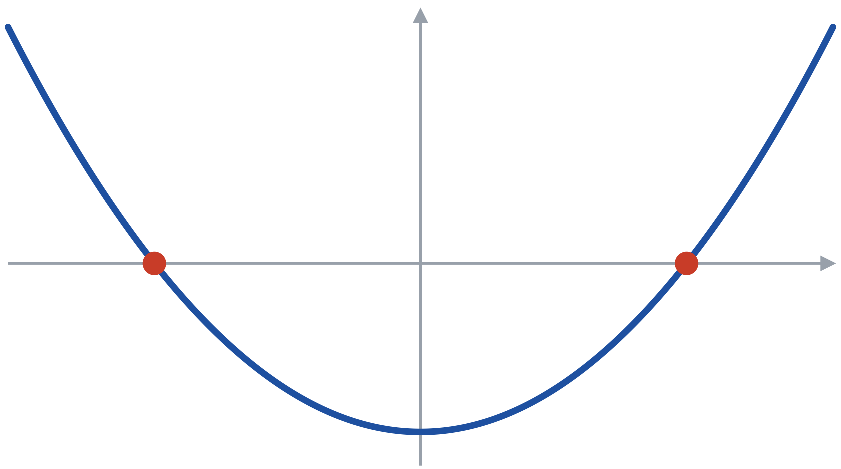
What *isotopy types*[1] can the nonsingular zero set of a polynomial in $\mathbb{R}[x]$ of degree d realize in $\mathbb{RP}^1$?

1 An ambient isotopy of a space X is a continuous map $F : X \times [0, 1] \to X$ with $F_0 = \mathrm{id}_X$ and every F_t a homeomorphism. Two subsets $A, B \subseteq X$ are *ambient isotopic* if some such F has $F_1(A) = B$. This partitions the subsets of X into *isotopy types*.

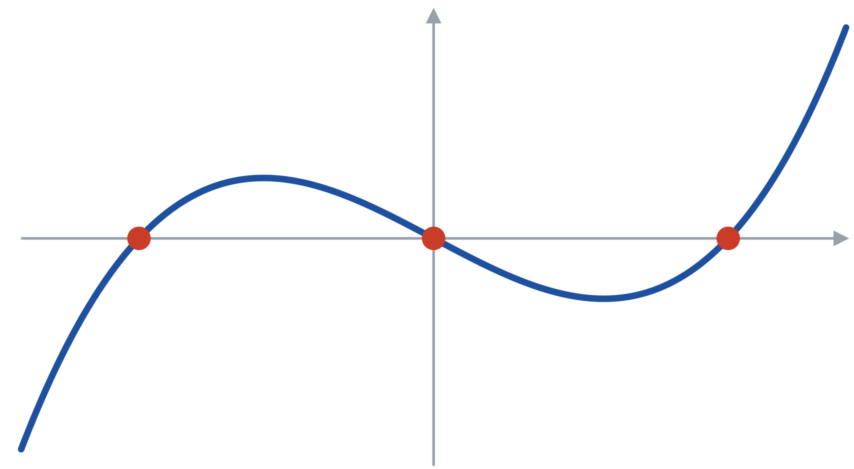
The trivial topology of $\mathbb{R}[x]$

QUESTION

What *isotopy types*[1] can the nonsingular zero set of a polynomial in $\mathbb{R}[x]$ of degree d realize in $\mathbb{RP}^1$?



$p(x) = (x + 1)(x - 1) = x^2 - 1$



$q(x) = (x + 1)x(x - 1) = x^3 - x$

1 An ambient isotopy of a space X is a continuous map $F : X \times [0, 1] \to X$ with $F_0 = \mathrm{id}_X$ and every F_t a homeomorphism. Two subsets $A, B \subseteq X$ are *ambient isotopic* if some such F has $F_1(A) = B$. This partitions the subsets of X into *isotopy types*.

The trivial topology of $\mathbb{R}[x]$

QUESTION

What *isotopy types*[1] can the nonsingular zero set of a polynomial in $\mathbb{R}[x]$ of degree d realize in $\mathbb{RP}^1$?

$p(x) = (x+1)(x-1) = x^2 - 1$

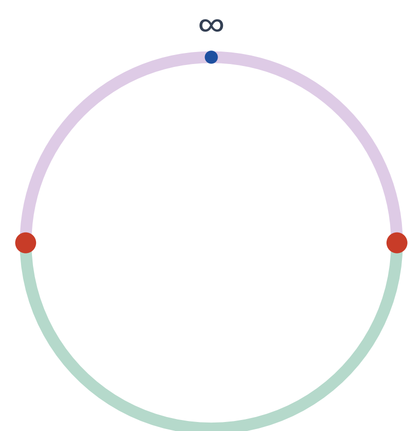
$q(x) = (x+1)x(x-1) = x^3 - x$

1 An ambient isotopy of a space X is a continuous map $F : X \times [0,1] \to X$ with $F_0 = \mathrm{id}_X$ and every F_t a homeomorphism. Two subsets $A, B \subseteq X$ are *ambient isotopic* if some such F has $F_1(A) = B$. This partitions the subsets of X into *isotopy types*.

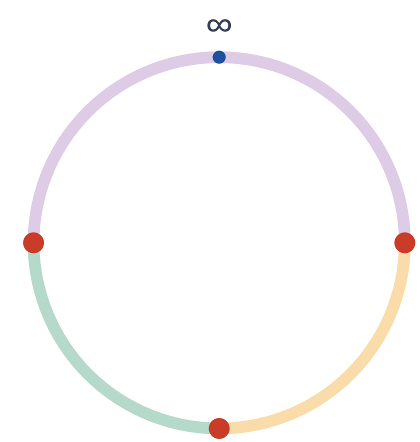
The trivial topology of $\mathbb{R}[x]$

QUESTION

What *isotopy types*[1] can the nonsingular zero set of a polynomial in $\mathbb{R}[x]$ of degree d realize in $\mathbb{RP}^1$?



$p(x) = (x+1)(x-1) = x^2 - 1$



$q(x) = (x+1)x(x-1) = x^3 - x$

1 An ambient isotopy of a space X is a continuous map $F : X \times [0,1] \to X$ with $F_0 = \mathrm{id}_X$ and every F_t a homeomorphism. Two subsets $A, B \subseteq X$ are *ambient isotopic* if some such F has $F_1(A) = B$. This partitions the subsets of X into *isotopy types*.

ZUSE INSTITUTE BERLIN

The trivial topology of $\mathbb{R}[x]$

QUESTION

What *isotopy types*[1] can the nonsingular zero set of a polynomial in $\mathbb{R}[x]$ of degree d realize in $\mathbb{RP}^1$?

Answer. Precisely those of r distinct points, where $0 \le r \le d$ and $r \equiv d \pmod 2$.

Proof. **Necessity** follows by Gauss: every polynomial has d complex roots, counted with multiplicity, and non-real roots come in complex-conjugate pairs. **Sufficiency** follows from the explicit construction

$$p_r(x) = \prod_{k=1}^{r}(x - a_k) \cdot \prod_{j=1}^{\frac{d-r}{2}}(x^2 + b_j)$$

for pairwise distinct $a_k \in \mathbb{R}$ and pairwise distinct $b_j > 0$. □

Things get a bit harder when making the jump to two variables...

1 An ambient isotopy of a space X is a continuous map $F : X \times [0,1] \to X$ with $F_0 = \mathrm{id}_X$ and every F_t a homeomorphism. Two subsets $A, B \subseteq X$ are *ambient isotopic* if some such F has $F_1(A) = B$. This partitions the subsets of X into *isotopy types*.

The more difficult topology of $\mathbb{R}[x, y]$

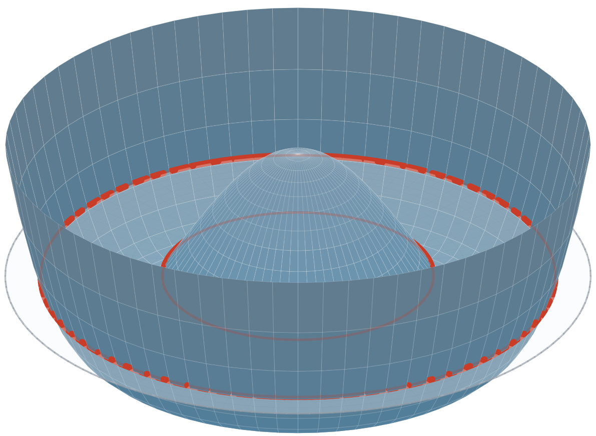
QUESTION · Hilbert, 1900, Problem 16

What isotopy types can the nonsingular zero set of a polynomial in $\mathbb{R}[x, y]$ of degree d realize in $\mathbb{RP}^2$?

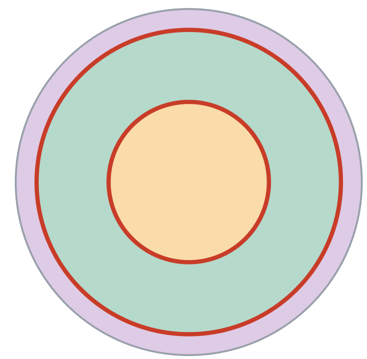
The more difficult topology of $\mathbb{R}[x, y]$

QUESTION · Hilbert, 1900, Problem 16

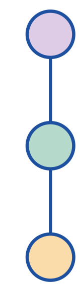
What isotopy types can the nonsingular zero set of a polynomial in $\mathbb{R}[x, y]$ of degree d realize in $\mathbb{RP}^2$?



→



→



GRAPH IN $\mathbb{R}^3$

$f(x, y) = \left(x^2 + y^2 - 1\right)\left(x^2 + y^2 - 4\right) + \frac{1}{4}$

REGIONS IN $\mathbb{R}^2$

$\{(x, y) \mid f(x, y) = 0\}$

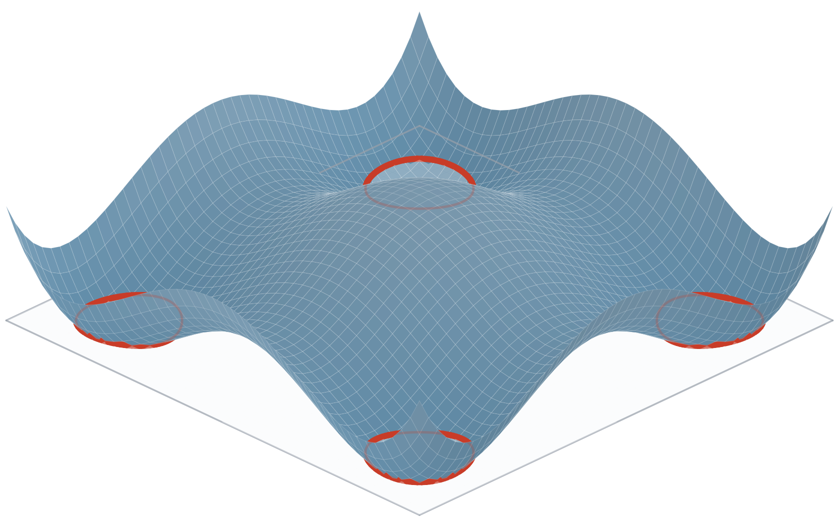
ROOTED TREE

$\langle 1 \langle 1 \rangle \rangle$

The more difficult topology of $\mathbb{R}[x, y]$

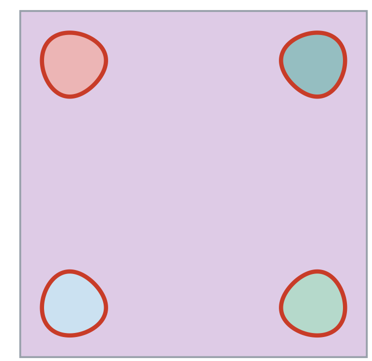
QUESTION · Hilbert, 1900, Problem 16

What isotopy types can the nonsingular zero set of a polynomial in $\mathbb{R}[x, y]$ of degree d realize in $\mathbb{RP}^2$?



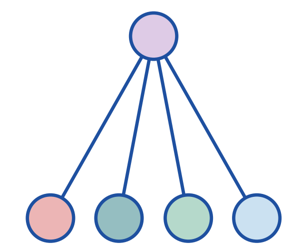
GRAPH IN $\mathbb{R}^3$

$$g(x, y) = \left(x^2 - 1\right)^2 + \left(y^2 - 1\right)^2 - \frac{1}{4}$$



REGIONS IN $\mathbb{R}^2$

$$\{(x, y) \mid g(x, y) = 0\}$$



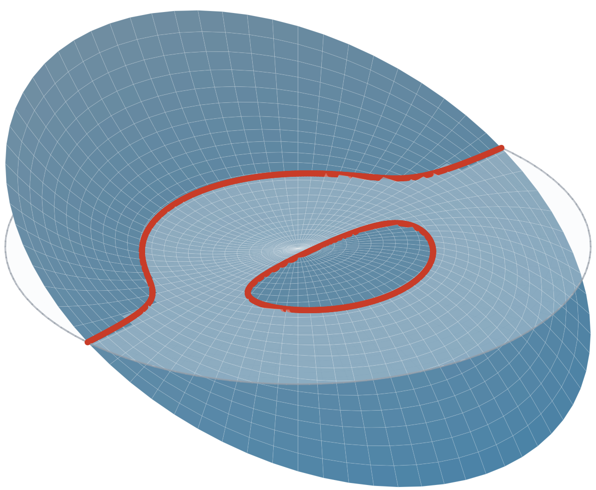
ROOTED TREE

$$\langle 4 \rangle$$

The more difficult topology of $\mathbb{R}[x, y]$

QUESTION · Hilbert, 1900, Problem 16

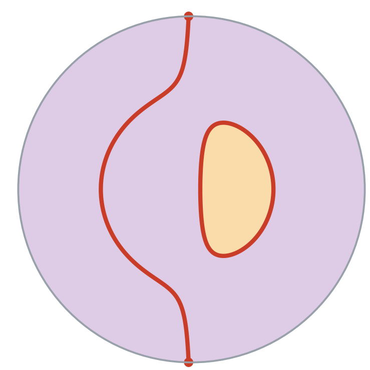
What isotopy types can the nonsingular zero set of a polynomial in $\mathbb{R}[x, y]$ of degree d realize in $\mathbb{RP}^2$?



GRAPH IN $\mathbb{R}^3$

$h(x, y) = y\left(x^2 + y^2 - 1\right) - \frac{1}{10}$

→



REGIONS IN $\mathbb{R}^2$

$\{(x, y) \mid h(x, y) = 0\}$

→

ROOTED TREE

$\langle J \sqcup 1 \rangle$

ZUSE INSTITUTE BERLIN

The more difficult topology of $\mathbb{R}[x, y]$

QUESTION · Hilbert, 1900, Problem 16

What isotopy types can the nonsingular zero set of a polynomial in $\mathbb{R}[x, y]$ of degree d realize in $\mathbb{RP}^2$?

Two things make this a harder question:

- The zero set is now the **disjoint union of circles**[1] embedded in $\mathbb{RP}^2$.
- $\mathbb{RP}^1$ is orientable, whereas $\mathbb{RP}^2$ is not: it contains a **Möbius band**.

Nesting and the distinguished Möbius region (the *root*) carry information beyond component count.

REMARK

These zero sets are called *(real plane) algebraic curves*. The *Viro notation* records their nesting:

- $\langle X \rangle$ is X inside another oval,
- $X \sqcup Y$ is the disjoint union of X and Y,
- kX denotes k disjoint copies of X,
- J denotes the unique pseudo-line present for odd d.

Here p counts ovals inside an even number of other ovals, and n those inside an odd number.

1 For even d, every component is an *oval* and hence bounds a disk. For odd d, one component is a *pseudo-line* whose neighborhood is a Möbius band; all others are ovals.

Classifying isotopy types: exclude & construct

The following degrees have been classified:

degree	# isotopy types	
2, 3, 4, 5	2, 2, 6, 8	Harnack · Hilbert
6	56	Gudkov '74
7	121	Viro '84

For degree 8, only *M-curves* — those attaining Harnack's bound — are (almost) understood:

THEOREM · Orevkov, 2002

Exactly 89 isotopy types occur for *flexible M*-curves; 83 have known algebraic realizations.

Classifying isotopy types: exclude & construct

The following degrees have been classified:

degree	# isotopy types	
2, 3, 4, 5	2, 2, 6, 8	Harnack · Hilbert
6	56	Gudkov '74
7	121	Viro '84

For degree 8, only *M-curves* — those attaining Harnack's bound — are (almost) understood:

THEOREM · Orevkov, 2002

Exactly 89 isotopy types occur for *flexible M-curves*; 83 have known algebraic realizations.

---

1 **Exclude** non-realizable isotopy types:

Harnack 1876
$\leq (d-1)(d-2)/2+1$ circles

Gudkov–Rokhlin 1971/72
$d = 2k: \quad p - n \equiv k^2 \pmod 8$

Bézout
nesting depth $\leq \lfloor d/2 \rfloor$

2 **Construct** explicit curves for everything else — more on that now ...

Computing a general polynomial's isotopy type requires resultants, Gröbner bases, CAD, and real-root isolation; too costly at scale. Can coefficient restrictions make the topology more readily accessible?

From polynomials to combinatorial patchworks

THEOREM · Viro, 1989; 2006

Let $\Delta_d \coloneqq \{(i,j) : i,j \geq 0, i+j \leq d\}$. For every $\sigma : \Delta_d \to \{-1,1\}$ and every strictly convex $v : \Delta_d \to \mathbb{R}_{\geq 0}$ inducing a unimodular[1] triangulation T, there exists $t_0 > 0$ so that for any $0 < t \leq t_0$ the type of

$$p_t(x,y) = \sum_{(i,j)\in\Delta_d} \sigma_{i,j}\, t^{v(i,j)}\, x^i y^j$$

can be determined solely from σ and the T of Δ_d induced by v.

1 Any triangulation coming from a strictly convex lifting function v is *regular* by definition. Regularity can be checked by solving an LP. *Unimodularity* is a separate condition requiring every triangle to have normalized lattice area 1.

From polynomials to combinatorial patchworks

THEOREM · Viro, 1989; 2006

Let $\Delta_d := \{(i,j) : i,j \geq 0, i+j \leq d\}$. For every $\sigma : \Delta_d \to \{-1,1\}$ and every strictly convex $v : \Delta_d \to \mathbb{R}_{\geq 0}$ inducing a unimodular[1] triangulation T, there exists $t_0 > 0$ so that for any $0 < t \leq t_0$ the type of

$$p_t(x,y) = \sum_{(i,j)\in\Delta_d} \sigma_{i,j}\, t^{v(i,j)}\, x^i y^j$$

can be determined solely from σ and the T of Δ_d induced by v.

Definition. This is called *combinatorial patchworking*: we restrict attention to the combinatorial space of regular, unimodular triangulations T with sign distributions σ on their vertices. A real algebraic curve obtained by this construction is called a T*-curve*.

OUR CONTRIBUTION

Viro described how to recover the isotopy type through geometric cutting, gluing, and contraction. A naïve implementation has estimated complexity $O\left(d^5\right)$. We describe and implement a fast, efficient, and scalable algorithm that derives the type directly from (T,σ) in near-linear time.

1 Any triangulation coming from a strictly convex lifting function v is *regular* by definition. Regularity can be checked by solving an LP. *Unimodularity* is a separate condition requiring every triangle to have normalized lattice area 1.

Computing the isotopy types of combinatorial patchworks

ALGORITHM

Input Triangulation T and signs σ.

Reflect T into four quadrants; extend σ by monomial parity.

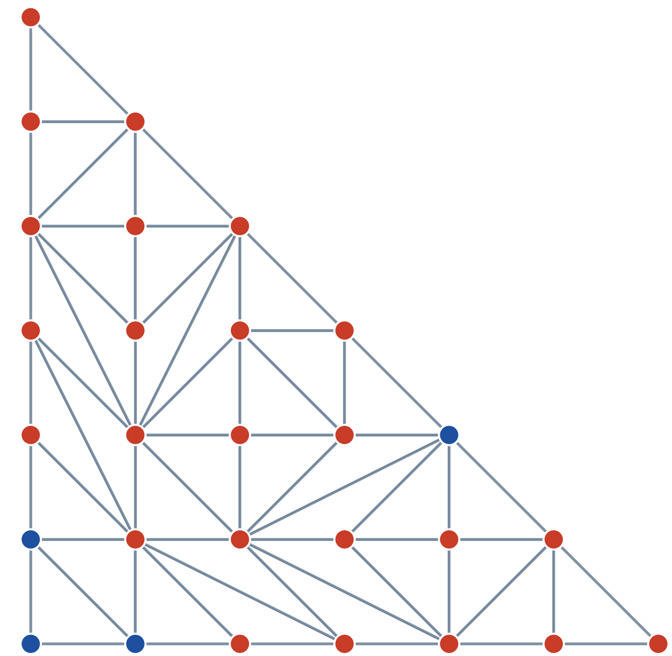
Components Delete crossed edges; compute the connected components.

Regions Merge components across antipodal boundary pairs into regions.

Root Find the Möbius region: crossed-edge adjacency (odd d), odd antipodal cycle (even d).

Canonicalize Build the region-adjacency graph from crossed edges; root and canonicalize.

Output Isotopy type in Viro notation.



Computing the isotopy types of combinatorial patchworks

ALGORITHM

Input Triangulation T and signs σ.

Reflect T into four quadrants; extend σ by monomial parity.

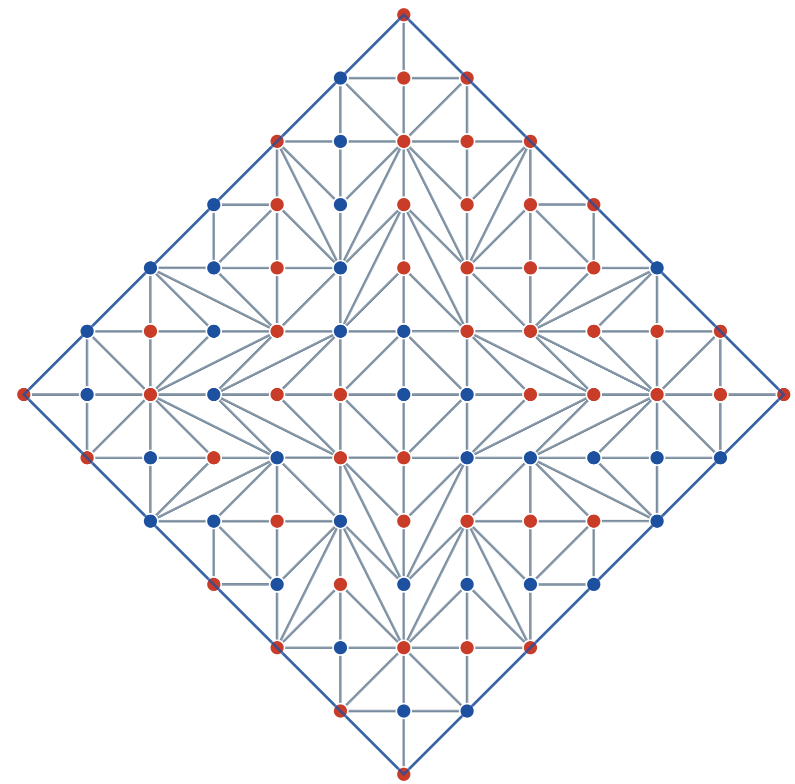
Components Delete crossed edges; compute the connected components.

Regions Merge components across antipodal boundary pairs into regions.

Root Find the Möbius region: crossed-edge adjacency (odd d), odd antipodal cycle (even d).

Canonicalize Build the region-adjacency graph from crossed edges; root and canonicalize.

Output Isotopy type in Viro notation.



Computing the isotopy types of combinatorial patchworks

ALGORITHM

Input Triangulation T and signs σ.

Reflect T into four quadrants; extend σ by monomial parity.

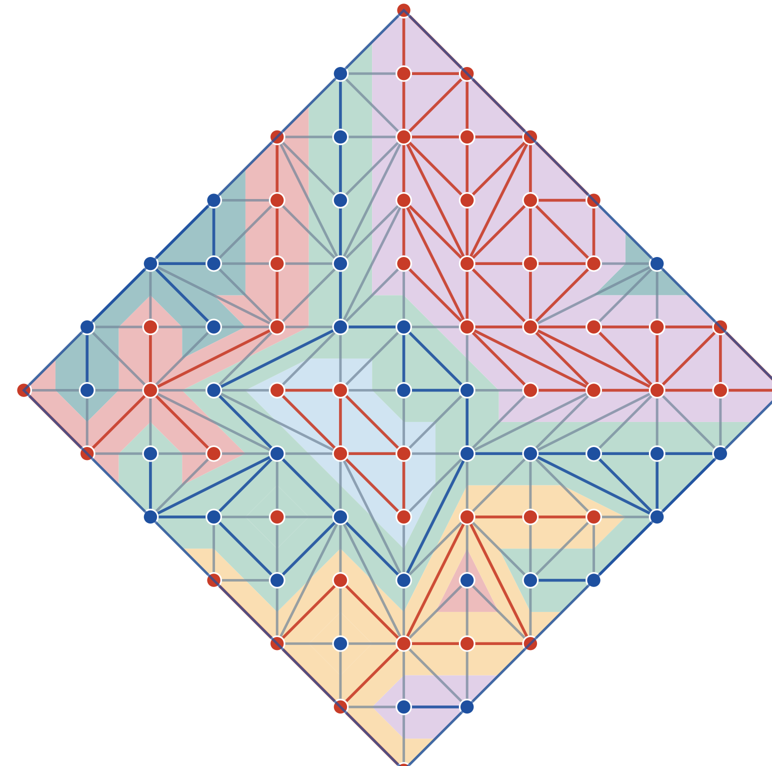
Components Delete crossed edges; compute the connected components.

Regions Merge components across antipodal boundary pairs into regions.

Root Find the Möbius region: crossed-edge adjacency (odd d), odd antipodal cycle (even d).

Canonicalize Build the region-adjacency graph from crossed edges; root and canonicalize.

Output Isotopy type in Viro notation.



Computing the isotopy types of combinatorial patchworks

ALGORITHM

Input Triangulation T and signs σ.

Reflect T into four quadrants; extend σ by monomial parity.

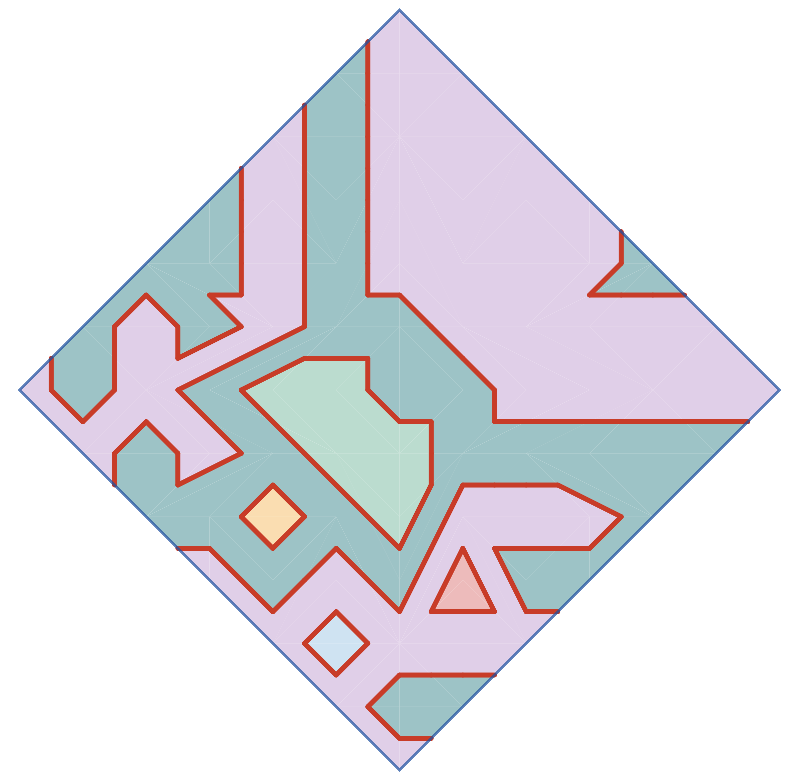
Components Delete crossed edges; compute the connected components.

Regions Merge components across antipodal boundary pairs into regions.

Root Find the Möbius region: crossed-edge adjacency (odd d), odd antipodal cycle (even d).

Canonicalize Build the region-adjacency graph from crossed edges; root and canonicalize.

Output Isotopy type in Viro notation.



Computing the isotopy types of combinatorial patchworks

ALGORITHM

Input Triangulation T and signs σ.

Reflect T into four quadrants; extend σ by monomial parity.

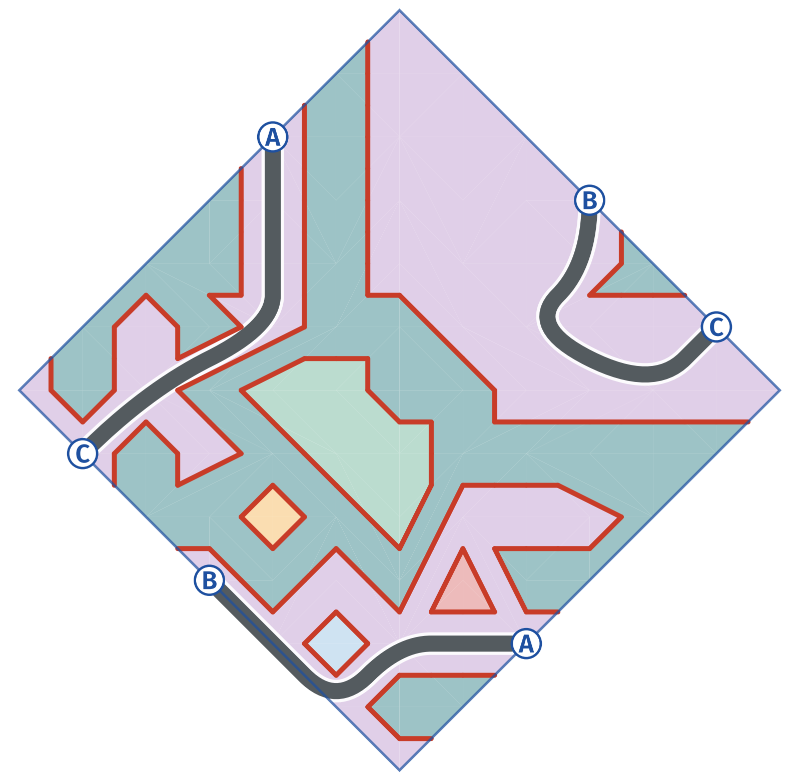
Components Delete crossed edges; compute the connected components.

Regions Merge components across antipodal boundary pairs into regions.

Root Find the Möbius region: crossed-edge adjacency (odd d), odd antipodal cycle (even d).

Canonicalize Build the region-adjacency graph from crossed edges; root and canonicalize.

Output Isotopy type in Viro notation.



Computing the isotopy types of combinatorial patchworks

ALGORITHM

Input Triangulation T and signs σ.

Reflect T into four quadrants; extend σ by monomial parity.

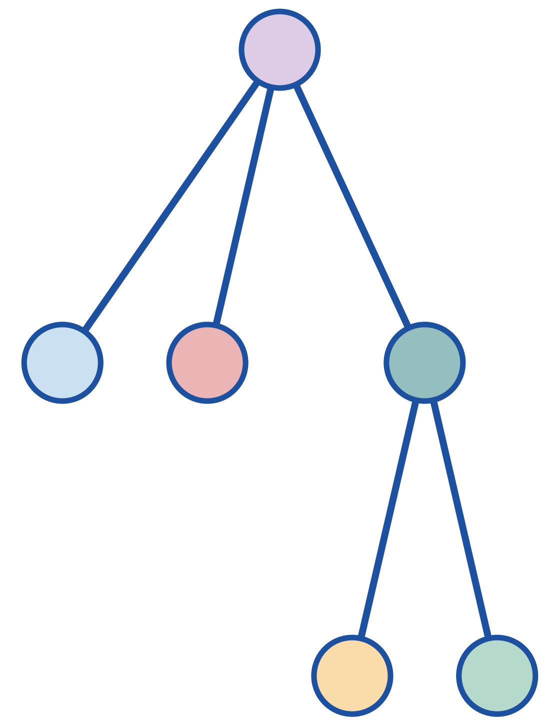
Components Delete crossed edges; compute the connected components.

Regions Merge components across antipodal boundary pairs into regions.

Root Find the Möbius region: crossed-edge adjacency (odd d), odd antipodal cycle (even d).

Canonicalize Build the region-adjacency graph from crossed edges; root and canonicalize.

Output Isotopy type in Viro notation.



Computing the isotopy types of combinatorial patchworks

ALGORITHM

Input Triangulation T and signs σ.

Reflect T into four quadrants; extend σ by monomial parity.

Components Delete crossed edges; compute the connected components.

Regions Merge components across antipodal boundary pairs into regions.

Root Find the Möbius region: crossed-edge adjacency (odd d), odd antipodal cycle (even d).

Canonicalize Build the region-adjacency graph from crossed edges; root and canonicalize.

Output Isotopy type in Viro notation.

$$\langle 2 \sqcup 1\langle 2\rangle\rangle$$

Computing the isotopy types of combinatorial patchworks

ALGORITHM

Input Triangulation T and signs σ.

Reflect T into four quadrants; extend σ by monomial parity.

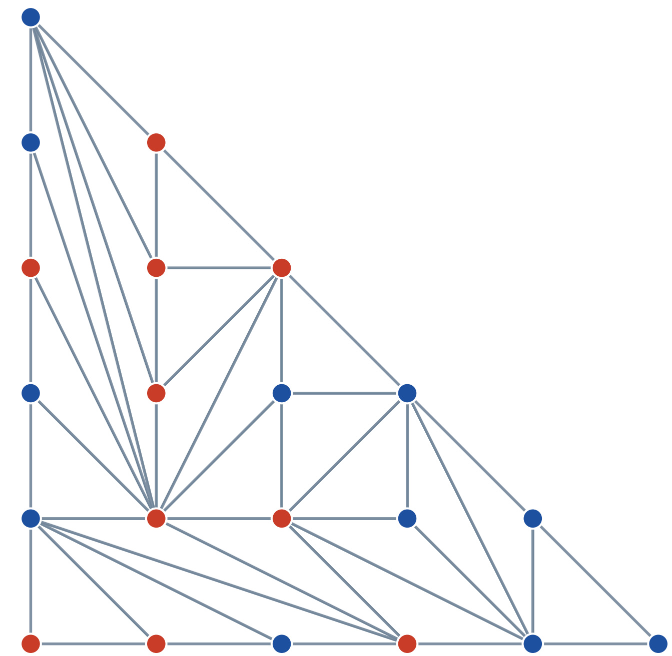
Components Delete crossed edges; compute the connected components.

Regions Merge components across antipodal boundary pairs into regions.

Root Find the Möbius region: crossed-edge adjacency (odd d), odd antipodal cycle (even d).

Canonicalize Build the region-adjacency graph from crossed edges; root and canonicalize.

Output Isotopy type in Viro notation.



Computing the isotopy types of combinatorial patchworks

ALGORITHM

Input Triangulation T and signs σ.

Reflect T into four quadrants; extend σ by monomial parity.

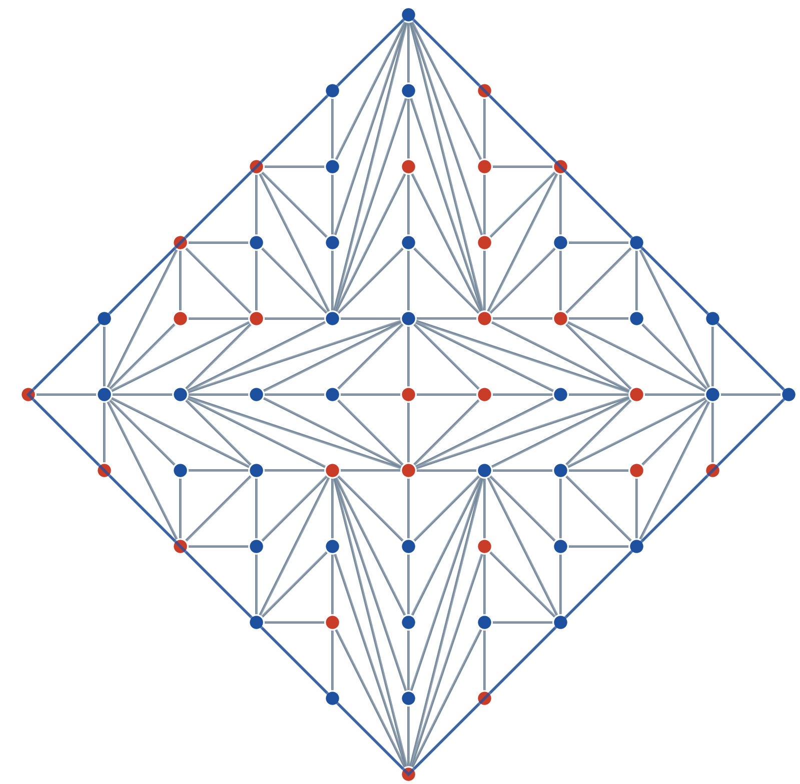
Components Delete crossed edges; compute the connected components.

Regions Merge components across antipodal boundary pairs into regions.

Root Find the Möbius region: crossed-edge adjacency (odd d), odd antipodal cycle (even d).

Canonicalize Build the region-adjacency graph from crossed edges; root and canonicalize.

Output Isotopy type in Viro notation.



Computing the isotopy types of combinatorial patchworks

ALGORITHM

Input Triangulation T and signs σ.

Reflect T into four quadrants; extend σ by monomial parity.

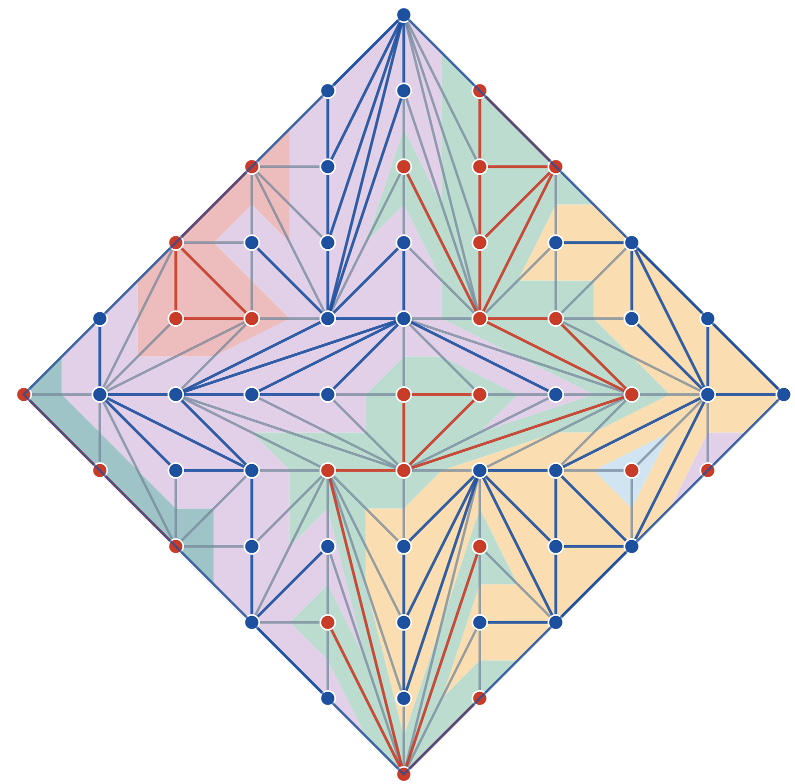
Components Delete crossed edges; compute the connected components.

Regions Merge components across antipodal boundary pairs into regions.

Root Find the Möbius region: crossed-edge adjacency (odd d), odd antipodal cycle (even d).

Canonicalize Build the region-adjacency graph from crossed edges; root and canonicalize.

Output Isotopy type in Viro notation.



ZUSE INSTITUTE BERLIN

Computing the isotopy types of combinatorial patchworks

ALGORITHM

Input Triangulation T and signs σ.

Reflect T into four quadrants; extend σ by monomial parity.

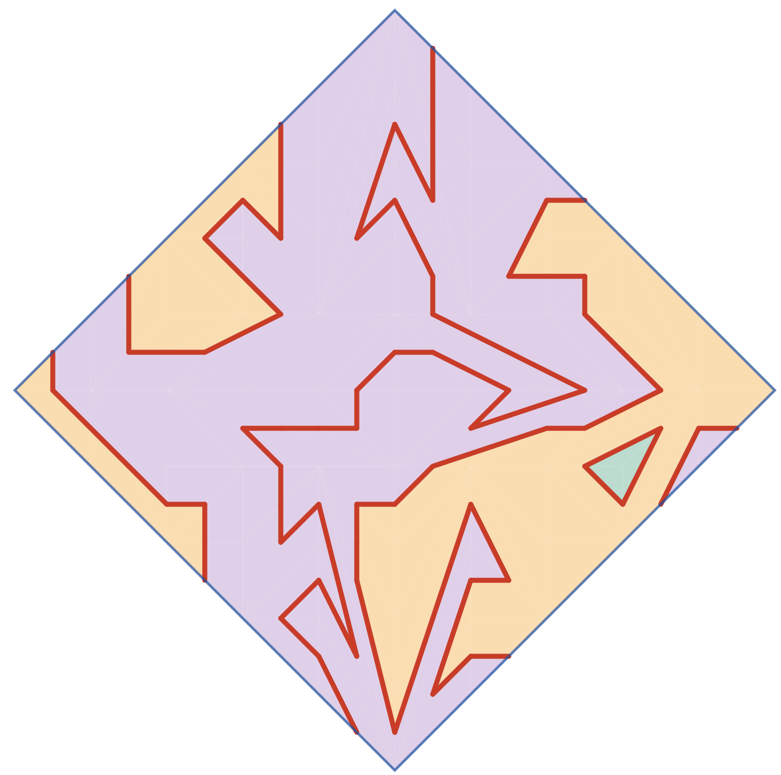
Components Delete crossed edges; compute the connected components.

Regions Merge components across antipodal boundary pairs into regions.

Root Find the Möbius region: crossed-edge adjacency (odd d), odd antipodal cycle (even d).

Canonicalize Build the region-adjacency graph from crossed edges; root and canonicalize.

Output Isotopy type in Viro notation.



Computing the isotopy types of combinatorial patchworks

ALGORITHM

Input Triangulation T and signs σ.

Reflect T into four quadrants; extend σ by monomial parity.

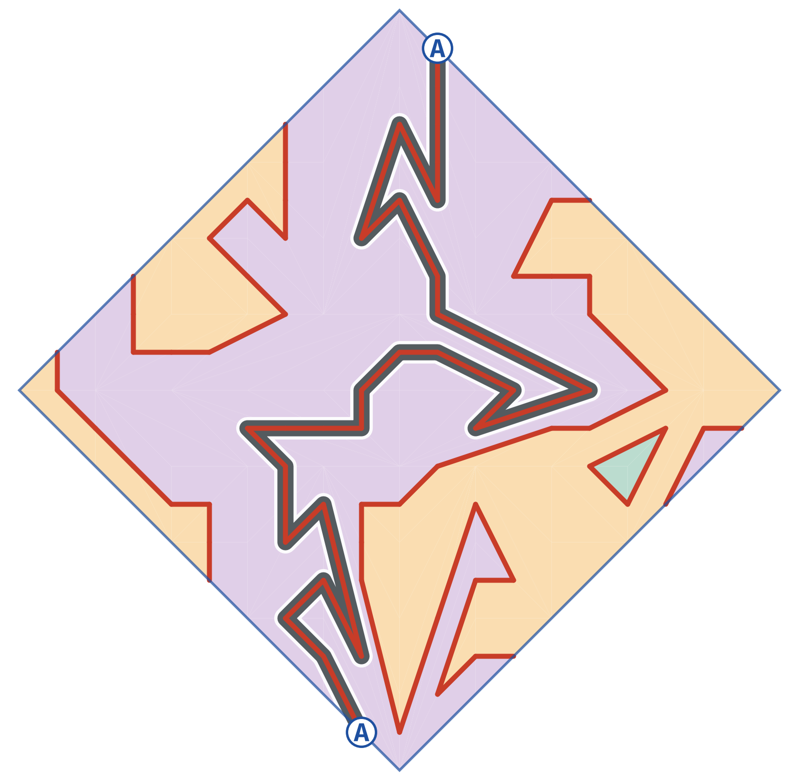
Components Delete crossed edges; compute the connected components.

Regions Merge components across antipodal boundary pairs into regions.

Root Find the Möbius region: crossed-edge adjacency (odd d), odd antipodal cycle (even d).

Canonicalize Build the region-adjacency graph from crossed edges; root and canonicalize.

Output Isotopy type in Viro notation.



Computing the isotopy types of combinatorial patchworks

ALGORITHM

Input Triangulation T and signs σ.

Reflect T into four quadrants; extend σ by monomial parity.

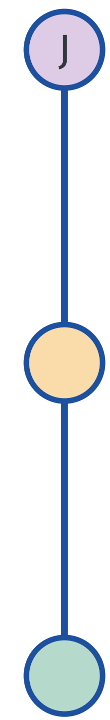
Components Delete crossed edges; compute the connected components.

Regions Merge components across antipodal boundary pairs into regions.

Root Find the Möbius region: crossed-edge adjacency (odd d), odd antipodal cycle (even d).

Canonicalize Build the region-adjacency graph from crossed edges; root and canonicalize.

Output Isotopy type in Viro notation.



Computing the isotopy types of combinatorial patchworks

ALGORITHM

Input Triangulation T and signs σ.

Reflect T into four quadrants; extend σ by monomial parity.

Components Delete crossed edges; compute the connected components.

Regions Merge components across antipodal boundary pairs into regions.

Root Find the Möbius region: crossed-edge adjacency (odd d), odd antipodal cycle (even d).

Canonicalize Build the region-adjacency graph from crossed edges; root and canonicalize.

Output Isotopy type in Viro notation.

$$\langle J \sqcup 1\langle 1\rangle\rangle$$

Why exhaustive sweeps are feasible

EFFICIENCY

The reflected triangulation has $O(d^2)$ vertices and edges. Union–find dominates the $O(d^2\alpha(d^2))$ running time (α: inverse Ackermann), so the algorithm is near-linear in input size.

PARALLELISM

The data structures fit in thread-local memory, so sweeps are naturally suited to GPU execution and reach billions of classifications per second.

SYMMETRY

Symmetries reduce the number of sign distributions to sweep by a factor of 8 and the number of triangulations by a factor of ~ 6.

In practice we sweep one axis at a time: all sign distributions for a fixed triangulation, or all symmetric triangulations for a fixed sign distribution.

d	M	**symmetric** T	**signs** σ	T **fixed: all** σ	σ **fixed: all symmetric** T
4	4	74	4,096	milliseconds	milliseconds
5	7	1,194	262K	milliseconds	milliseconds
6	11	62,960	34M	<1 s	<1 s
7	16	4.7M	8.6B	≈ 30 s	≈ 5 s
8	22	1.2B	4.4T	<1 h	minutes

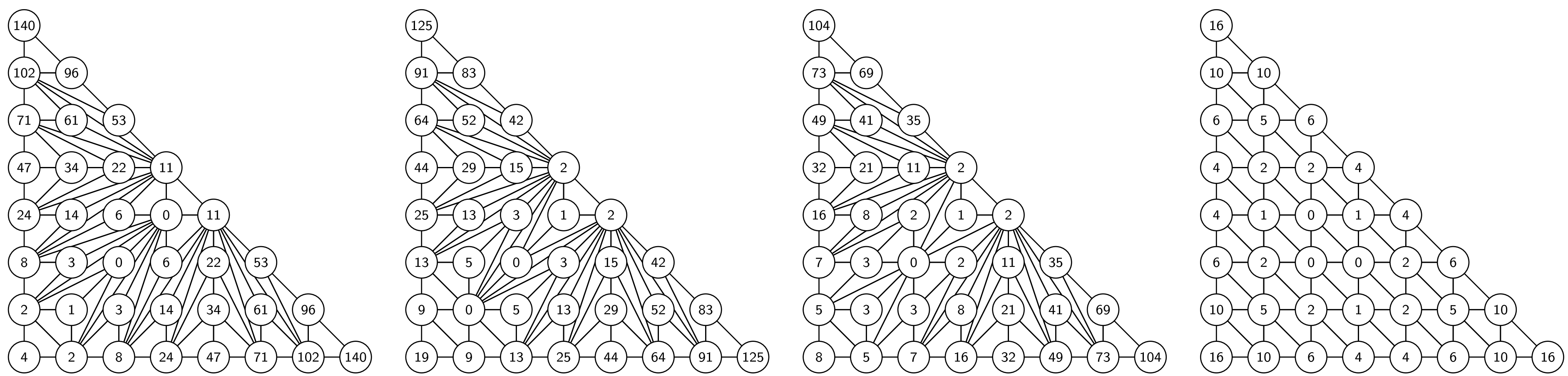
Rough wall times from current GPU benchmarks; degree-8 performance depends strongly on symmetry, batching, and hardware.

Degree seven: every isotopy type is a T-curve

Before our work, the largest explicit list contained 24 degree-seven T-curves (De Loera–Wicklin).

THEOREM · Geiselmann et al., 2026

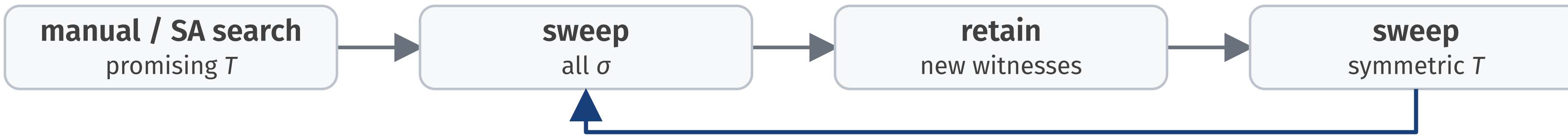
Every isotopy type of degree 7 is realizable as a T-curve. Just *four* triangulations of Δ_7 suffice.



centrifugal split frayed honeycomb

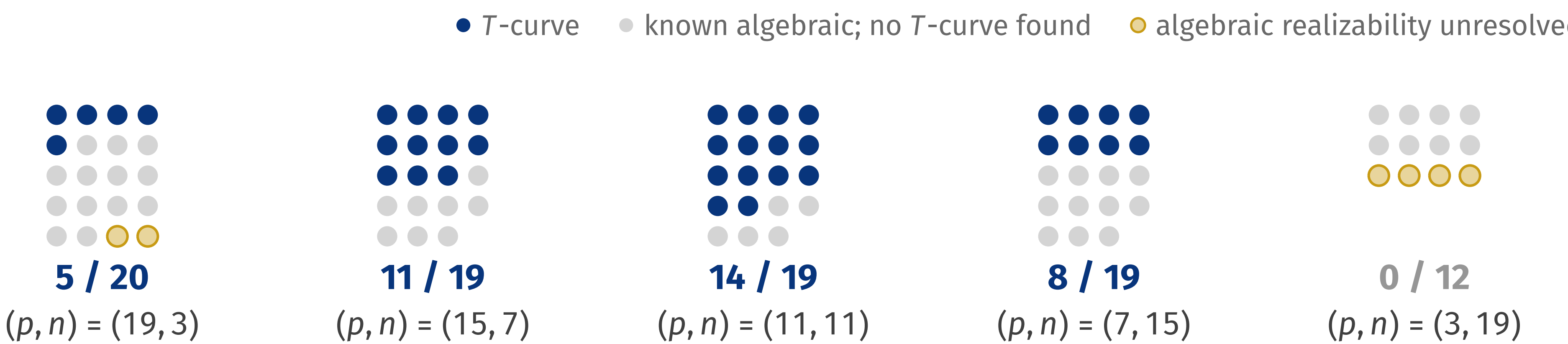
Process. We searched manually for promising triangulations and swept all (symmetry-reduced) 8.6B sign distributions on each. These four cover all 121 isotopy types; centrifugal alone realizes 115.

Degree eight: $2,368$ isotopy types from patchworking



PROPOSITION · Geiselmann et al., 2026

At least $2,368$ isotopy types of degree 8 are realizable as T-curves, including 38 M-curves.



One column is suspiciously empty...

When does patchworking cease to be complete?

Combinatorial patchworking is tractable precisely because it restricts the polynomial space. Itenberg proved that this restriction misses isotopy types in high degree, but the first failure was unknown.

PROBLEM · Itenberg–Viro, 1996

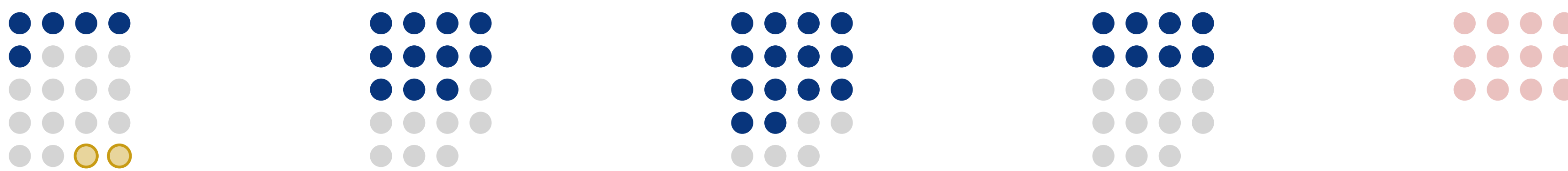
Does this failure already occur in degree 7 or 8?

DEGREE 7 · COMPUTATION

Complete: every isotopy type is a T-curve.

DEGREE 8 · THEORY

Incomplete: $(p, n) = (3, 19)$ cannot occur for T-curves.



5 / 20	**11 / 19**	**14 / 19**	**8 / 19**	**0 / 12**
$(p, n) = (19, 3)$	$(p, n) = (15, 7)$	$(p, n) = (11, 11)$	$(p, n) = (7, 15)$	$(p, n) = (3, 19)$

Degree 8 is the first degree at which combinatorial patchworking misses an isotopy type.

Papers and references

THEORETICAL RESULTS

121 Patchworked Curves of Degree Seven
Geiselmann et al. (2026) · arXiv:2602.06888

COMPUTATIONAL METHODOLOGY

Fast Isotopy Computation for T-Curves
Geiselmann et al. (2026) · arXiv:2604.09221
Accepted for the ICMS 2026 proceedings (Springer LNCS).

References

A. Harnack (1876). "Über die Vieltheiligkeit der ebenen algebraischen Curven." *Mathematische Annalen* 10(2), 189–198. DOI.

D. Hilbert (1891). "Ueber die reellen Züge algebraischer Curven." *Mathematische Annalen* 38(1), 115–138. DOI.

D. Hilbert (1900). "Mathematische Probleme." *Nachrichten von der Königl. Gesellschaft der Wissenschaften zu Göttingen, Math.-Phys. Klasse*, 253–297. DOI.

D. A. Gudkov (1971). "Construction of a New Series of M-curves." *Doklady Akademii Nauk SSSR* 200(6), 1269–1272.

V. A. Rokhlin (1972). "Proof of Gudkov's Hypothesis." *Functional Analysis and Its Applications* 6(2), 136–138. DOI.

D. A. Gudkov (1974). "The Topology of Real Projective Algebraic Varieties." *Russian Mathematical Surveys* 29(4), 1–79. DOI.

O. Ya. Viro (1984). "Gluing of Plane Real Algebraic Curves and Constructions of Curves of Degrees 6 and 7." In *Topology (Leningrad, 1982)*, Lecture Notes in Mathematics 1060, Springer, 187–200. DOI.

O. Ya. Viro (1989). "Real Plane Algebraic Curves: Constructions with Controlled Topology." *Algebra i Analiz* 1(5), 1–73.

I. Itenberg and O. Viro (1996). "Patchworking Algebraic Curves Disproves the Ragsdale Conjecture." *The Mathematical Intelligencer* 18(4), 19–28. DOI.

J. A. De Loera and F. J. Wicklin (1998). "On the Need of Convexity in Patchworking." *Advances in Applied Mathematics* 20(2), 188–219. DOI.

S. Yu. Orevkov (2002). "Classification of Flexible M-Curves of Degree 8 up to Isotopy." *Geometric and Functional Analysis* 12(4), 723–755. DOI.

O. Viro (2006). "Patchworking Real Algebraic Varieties." arXiv.

ご清聴ありがとうございました

Thank you!