Developing a New Tool for Modeling the Topology of Zero Sets of Bivariate Pentanomials

Vaishali Miriyagalla

TAMU

July 24, 2023

Overview

- Terminology and Background
- Motivation and Goals
- Matlab Program
- Results

Near Circuit Polynomials

Support

Def: Given a polynomial f, the **support** is its set of exponent vectors

E.g. $f(x, y) = 1 - x - y + x^4 y + xy^4$, support $\mathcal{A} = \begin{bmatrix} 0 & 1 & 0 & 4 & 1 \\ 0 & 0 & 1 & 1 & 4 \end{bmatrix}$

Near Circuit Polynomials

Support

Def: Given a polynomial f, the **support** is its set of exponent vectors

E.g. $f(x,y) = 1 - x - y + x^4y + xy^4$, support $\mathcal{A} = \begin{bmatrix} 0 & 1 & 0 & 4 & 1 \\ 0 & 0 & 1 & 1 & 4 \end{bmatrix}$

Near Circuit Polynomials

Def: A polynomial whose support $\mathcal{A} = [a_1, \ldots, a_{n+3}] \in \mathbb{Z}^{n\times(n+3)}$ yields $\begin{bmatrix} 1 & \cdots & 1 \\ a_1 & \cdots & a_{n+3} \end{bmatrix}$ having rank $n+1$.

E.g. a **bivariate pentanomial** has 2 variables and 5 terms.

$n =$ the number of variables

Zero sets and Discriminants

Zero set: set of real inputs that make a polynomial evaluate to zero

Zero sets and Discriminants

Zero set: set of real inputs that make a polynomial evaluate to zero

Topology of Zero sets:

- Univariate ($n = 1$): number of zeros or roots
- Bivariate ($n = 2$): number of pieces (connected components)

Zero sets and Discriminants

Zero set: set of real inputs that make a polynomial evaluate to zero

Topology of Zero sets:

- Univariate ($n = 1$): number of zeros or roots
- Bivariate ($n = 2$): number of pieces (connected components)

$\mathcal{A}$-**discriminant polynomial**: polynomial in coefficients of f vanishing when f has a singular zero set

- For near circuits, can simplify to a bivariate polynomial: **reduced**

Zero sets and Discriminants

Zero set: set of real inputs that make a polynomial evaluate to zero

Topology of Zero sets:

- Univariate ($n = 1$): number of zeros or roots
- Bivariate ($n = 2$): number of pieces (connected components)

$\mathcal{A}$-**discriminant polynomial**: polynomial in coefficients of f vanishing when f has a singular zero set

- For near circuits, can simplify to a bivariate polynomial: **reduced**
- Recall quadratics from Algebra 1:
  if $f(x) = ax^2 + bx + c$, then the discriminant $= b^2 - 4ac$

Zero sets and Discriminants

Zero set: set of real inputs that make a polynomial evaluate to zero

Topology of Zero sets:

- Univariate ($n = 1$): number of zeros or roots
- Bivariate ($n = 2$): number of pieces (connected components)

$\mathcal{A}$-**discriminant polynomial**: polynomial in coefficients of f vanishing when f has a singular zero set

- For near circuits, can simplify to a bivariate polynomial: **reduced**
- Recall quadratics from Algebra 1:
  if $f(x) = ax^2 + bx + c$, then the discriminant $= b^2 - 4ac$

Zero sets and Discriminants

Zero set: set of real inputs that make a polynomial evaluate to zero

Topology of Zero sets:

- Univariate ($n = 1$): number of zeros or roots
- Bivariate ($n = 2$): number of pieces (connected components)

$\mathcal{A}$-**discriminant polynomial**: polynomial in coefficients of f vanishing when f has a singular zero set

- For near circuits, can simplify to a bivariate polynomial: **reduced**
- Recall quadratics from Algebra 1:
  if $f(x) = ax^2 + bx + c$, then the discriminant $= b^2 - 4ac$

$\mathcal{A}$-**discriminant variety**: where $\mathcal{A}$-discriminant $= 0$
i.e. critical points/curves where the topology of the zero set changes

Reduced[1] $\mathcal{A}$-discriminant variety for $\mathcal{A} = \begin{bmatrix} 0 & 1 & 0 & 4 & 1 \\ 0 & 0 & 1 & 1 & 4 \end{bmatrix}$

$$
\begin{aligned}
&363087263602825104457728a^{32}b^{8} - 2904698108822600835661824a^{29}b^{11} + \\
&10166443380879102924816384a^{26}b^{14} - 20332886761758205849632768a^{23}b^{17} + \\
&25416108452197757312040960a^{20}b^{20} - 20332886761758205849632768a^{17}b^{23} + \\
&10166443380879102924816384a^{14}b^{26} - 2904698108822600835661824a^{11}b^{29} + \\
&363087263602825104457728a^{8}b^{32} - 726174527205650208915456a^{31}b^{4} + 5798049740657613386809344a^{28}b^{7} + \\
&3128223705478090040593 6128a^{25}b^{10} - 50571247933680984080252928a^{22}b^{13} - \\
&19129025553375088862665 1136a^{19}b^{16} + 48223661844948968014243 4304a^{16}b^{19} - \\
&36318938189571339901835 6736a^{13}b^{22} + 7448962142351708783640 5760a^{10}b^{25} + \\
&1269670774911129037150 6176a^{7}b^{28} - 726174527205650208915456a^{4}b^{31} + 363087263602825104457728a^{30} - \\
&2893351631835012551147520a^{27}b^{3} - 9297323772275431783268 3520a^{24}b^{6} + \\
&13470366556574773615263 7440a^{21}b^{9} + 25351194225538809508921 34400a^{18}b^{12} + \\
&6930726608820725492905672704a^{15}b^{15} + 10397247952186084766590697472a^{12}b^{18} + \\
&1368264254117216589547831296a^{9}b^{21} + 1788103497072364267461 67296a^{6}b^{24} - \\
&97920096402886895358443 52a^{3}b^{27} + 363087263602825104457728b^{30} + \\
&51524645931445780035403776a^{23}b^{2} - 382889518651229479821639 68a^{20}b^{5} - \\
&459434896114086755201292 6976a^{17}b^{8} - 1813816331637440665952767 1808a^{14}b^{11} - \\
&2131928212143098218696356 6592a^{11}b^{14} + 2514558123743644571580497920a^{8}b^{17} - \\
&269737322421295126029533184a^{5}b^{20} - 20941053496075364622925824a^{2}b^{23} - \\
&2551128356732845719499571 2a^{19}b + 2225676679631729339955937280a^{16}b^{4} + \\
&1359100006303368527105490 9440a^{13}b^{7} + 14323107664774924348979937280a^{10}b^{10} - \\
&11483443502644561069909999616a^{7}b^{13} + 384254443547034707078152192a^{4}b^{16} + \\
&33392996500536631555522560ab^{19} - 5579692232310799015608 32a^{15} - \\
&2845499698372999866809843712a^{12}b^{3} - 4692084142913135619868721152a^{9}b^{6} + \\
&8896118143687124537286066176a^{6}b^{9} - 82843494158262383800850 8416a^{3}b^{12} - \\
&557969223231079901560832b^{15} + 16445468110480590903666278 40a^{8}b^{2} - \\
&9711410059602431138149171 20a^{5}b^{5} + 4910693845839500651939758 08a^{2}b^{8} - \\
&3945942476683996789577871 36a^{4}b - 85689226175777908279603 20ab^{4} + 41987654504771523593992227
\end{aligned}
$$

[1] Reduced coefficient vector is $c := [1, 1, 1, a, b]$

Modeling the $\mathcal{A}$-discriminant Variety

Parametrize the $\mathcal{A}$-discriminant variety: **Horn-Kapranov Uniformization**

Modeling the $\mathcal{A}$-discriminant Variety

Parametrize the $\mathcal{A}$-discriminant variety: **Horn-Kapranov Uniformization**

E.g. $f(x,y) = c_1 + c_2 x + c_3 y + c_4 x^4 y + c_5 x y^4$

1. add a row of ones above $\mathcal{A}$ to make $\hat{\mathcal{A}} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 4 & 1 \\ 0 & 0 & 1 & 1 & 4 \end{bmatrix}$

Modeling the $\mathcal{A}$-discriminant Variety

Parametrize the $\mathcal{A}$-discriminant variety: **Horn-Kapranov Uniformization**

E.g. $f(x,y) = c_1 + c_2x + c_3y + c_4x^4y + c_5xy^4$

1. add a row of ones above $\mathcal{A}$ to make $\hat{\mathcal{A}} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 4 & 1 \\ 0 & 0 & 1 & 1 & 4 \end{bmatrix}$
2. let $\mathcal{B}$ be a 2×5 matrix whose columns form a basis for the right nullspace of $\hat{\mathcal{A}}$

Modeling the $\mathcal{A}$-discriminant Variety

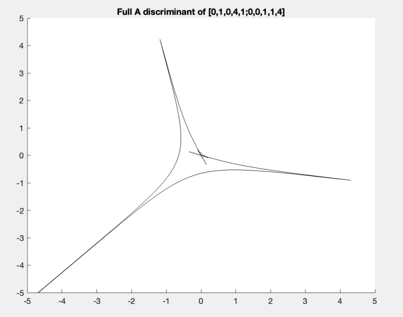
Parametrize the $\mathcal{A}$-discriminant variety: **Horn-Kapranov Uniformization**

E.g. $f(x, y) = c_1 + c_2x + c_3y + c_4x^4y + c_5xy^4$

1. add a row of ones above $\mathcal{A}$ to make $\hat{\mathcal{A}} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 4 & 1 \\ 0 & 0 & 1 & 1 & 4 \end{bmatrix}$
2. let $\mathcal{B}$ be a 2×5 matrix whose columns form a basis for the right nullspace of $\hat{\mathcal{A}}$
3. let λ be the variable of parametrization
4. $\log |\lambda \cdot \mathcal{B}^\top| \cdot \mathcal{B}$ parametrizes the $\mathcal{A}$-discriminant variety

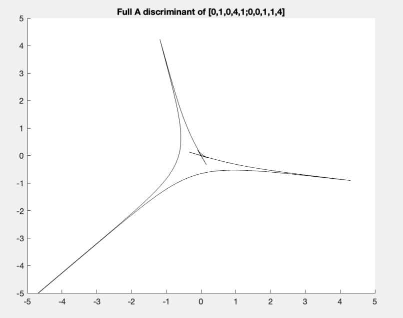
E.g. $f(x,y) = c_1 + c_2x + c_3y + c_4x^4y + c_5xy^4$

Parametrized $\mathcal{A}$-discriminant Variety

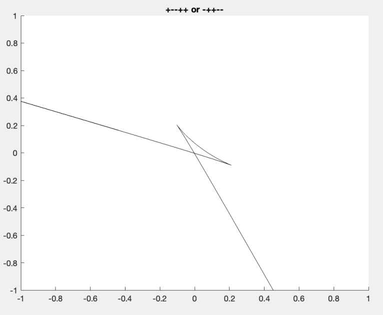


E.g. $f(x,y) = c_1 + c_2 x + c_3 y + c_4 x^4 y + c_5 x y^4$

Parametrized $\mathcal{A}$-discriminant Variety

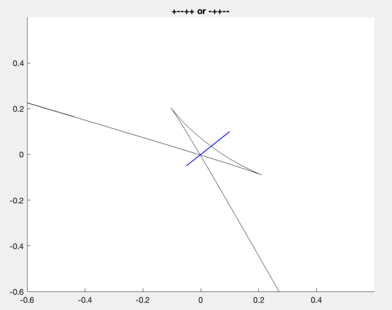


Signed Contour: $+--++$



Zero sets of $f(x, y) = c_1 + c_2x + c_3y + c_4x^4y + c_5xy^4$

Signed Contour: $+ - - + +$

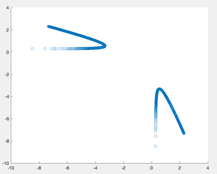


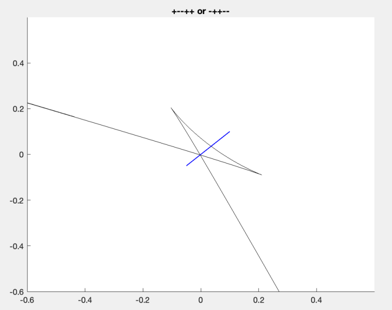
Zero sets of $f(x,y) = c_1 + c_2x + c_3y + c_4x^4y + c_5xy^4$

coefficients:

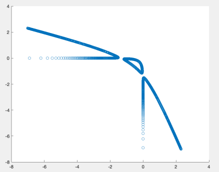
Signed Contour: $+ - - + +$

$[1, -\frac{3}{4}, -\frac{3}{4}, 1, 1]$

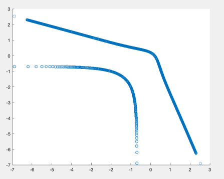




$[1, -1, -1, 1, 1]$



$[\frac{1}{2}, -1, -1, \frac{1}{2}, \frac{1}{2}]$



Motivation and Goals

- Explicitly drawing those zero sets is **NP hard**.

Motivation and Goals

- Explicitly drawing those zero sets is **NP hard**.
- We want to approximate the **isotopy type** (rough shape and number) of pieces of a zero set.

Motivation and Goals

- Explicitly drawing those zero sets is **NP hard**.
- We want to approximate the **isotopy type** (rough shape and number) of pieces of a zero set.
- Matlab Program:
  1. Draw parametrized $\mathcal{A}$-discriminant variety and signed contours

Motivation and Goals

- Explicitly drawing those zero sets is **NP hard**.
- We want to approximate the **isotopy type** (rough shape and number) of pieces of a zero set.
- Matlab Program:
  1. Draw parametrized $\mathcal{A}$-discriminant variety and signed contours
  2. Find which signed contours may have inner chambers

Motivation and Goals

- Explicitly drawing those zero sets is **NP hard**.
- We want to approximate the **isotopy type** (rough shape and number) of pieces of a zero set.
- Matlab Program:
  1. Draw parametrized $\mathcal{A}$-discriminant variety and signed contours
  2. Find which signed contours may have inner chambers
  3. Determine isotopy type for outer chambers using Triangulations and Viro's Patchworking

Motivation and Goals

- Explicitly drawing those zero sets is **NP hard**.
- We want to approximate the **isotopy type** (rough shape and number) of pieces of a zero set.
- Matlab Program:
  1. Draw parametrized $\mathcal{A}$-discriminant variety and signed contours
  2. Find which signed contours may have inner chambers
  3. Determine isotopy type for outer chambers using Triangulations and Viro's Patchworking
- Continue developing approximations to determine which chamber a given coefficient vector lies in.

Program: Detecting Cusps and Inner Chambers

Rusekshih 2017

For near-circuits, there are at most n cusps within one signed contour.

Program: Detecting Cusps and Inner Chambers

Rusekshih 2017

For near-circuits, there are at most n cusps within one signed contour.

- For $n = 2 \rightarrow$ at most 2 cusps within one signed contour.
- If one signed contour has two cusps, we may have an inner chamber.

Program: Detecting Cusps and Inner Chambers

Rusekshih 2017

For near-circuits, there are at most n cusps within one signed contour.

- For $n = 2 \rightarrow$ at most 2 cusps within one signed contour.
- If one signed contour has two cusps, we may have an inner chamber.
- To eliminate the possibility of having an inner chamber, detect signed contours with two cusps (cusp: $\frac{dx}{dt} = 0$ and $\frac{dy}{dt} = 0$).

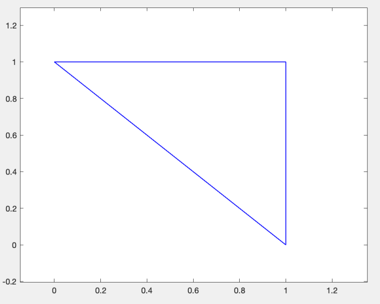
Background on Triangulations and Viro's Patchworking

Simple e.g. $f(x) = c_1 x + c_2 y - c_3 xy$: $\mathcal{A} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$

Background on Triangulations and Viro's Patchworking

Simple e.g. $f(x) = c_1x + c_2y - c_3xy$: $\mathcal{A} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$

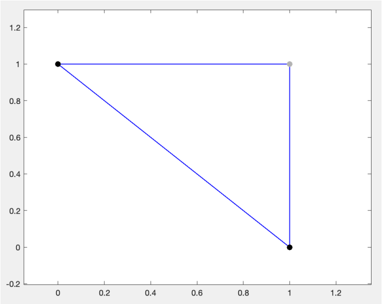
1. Plot exponent vectors as points and draw convex polytope
2. Label signs at each vertex
3. Draw outer normals from edges with vertices of opposite signs and connect them



Background on Triangulations and Viro's Patchworking

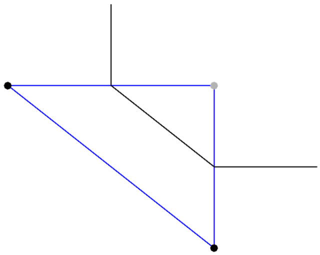
Simple e.g. $f(x) = c_1x + c_2y - c_3xy$: signs $= + + -$

1. Plot exponent vectors as points and draw convex polytope
2. Label signs at each vertex
3. Draw outer normals from edges with vertices of opposite signs and connect them



Background on Triangulations and Viro's Patchworking

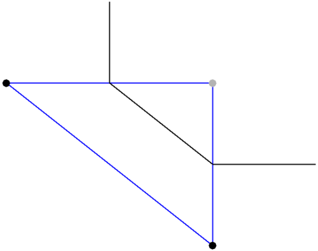
Simple e.g. $f(x) = c_1 x + c_2 y - c_3 xy$



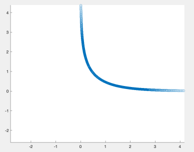
1. Plot exponent vectors as points and draw convex polytope
2. Label signs at each vertex
3. Draw outer normals from edges with vertices of opposite signs and connect them

Background on Triangulations and Viro's Patchworking

Simple e.g. $f(x) = c_1x + c_2y - c_3xy$



1. Plot exponent vectors as points and draw convex polytope
2. Label signs at each vertex
3. Draw outer normals from edges with vertices of opposite signs and connect them



actual zero set

Program: Isotopy Type (part 1: Triangulations)

Recall e.g. $\mathcal{A} = \begin{bmatrix} 0 & 1 & 0 & 4 & 1 \\ 0 & 0 & 1 & 1 & 4 \end{bmatrix}$

Program: Isotopy Type (part 1: Triangulations)

Recall e.g. $\mathcal{A} = \begin{bmatrix} 0 & 1 & 0 & 4 & 1 \\ 0 & 0 & 1 & 1 & 4 \end{bmatrix}$

Plot columns as coordinates

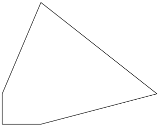
Draw convex polytope

Program: Isotopy Type (part 1: Triangulations)

Recall e.g. $\mathcal{A} = \begin{bmatrix} 0 & 1 & 0 & 4 & 1 \\ 0 & 0 & 1 & 1 & 4 \end{bmatrix}$

Plot columns as coordinates

Draw convex polytope

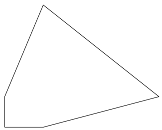


Program: Isotopy Type (part 1: Triangulations)

Recall e.g. $\mathcal{A} = \begin{bmatrix} 0 & 1 & 0 & 4 & 1 \\ 0 & 0 & 1 & 1 & 4 \end{bmatrix}$

Plot columns as coordinates

Draw convex polytope



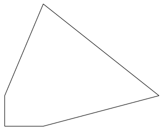
Triangulating with Five Vertices

Program: Isotopy Type (part 1: Triangulations)

Recall e.g. $\mathcal{A} = \begin{bmatrix} 0 & 1 & 0 & 4 & 1 \\ 0 & 0 & 1 & 1 & 4 \end{bmatrix}$

Plot columns as coordinates

Draw convex polytope



Triangulating with Five Vertices

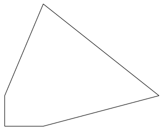
- Add $-\log|c|$ as third row to support $\mathcal{A}$, where $c = [c_1, c_2, c_3, c_4, c_5]$

Program: Isotopy Type (part 1: Triangulations)

Recall e.g. $\mathcal{A} = \begin{bmatrix} 0 & 1 & 0 & 4 & 1 \\ 0 & 0 & 1 & 1 & 4 \end{bmatrix}$

Plot columns as coordinates

Draw convex polytope



Triangulating with Five Vertices

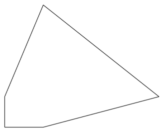
- Add $-\log|c|$ as third row to support $\mathcal{A}$, where $c = [c_1, c_2, c_3, c_4, c_5]$
- Compute convex hull of lifted support

Program: Isotopy Type (part 1: Triangulations)

Recall e.g. $\mathcal{A} = \begin{bmatrix} 0 & 1 & 0 & 4 & 1 \\ 0 & 0 & 1 & 1 & 4 \end{bmatrix}$

Plot columns as coordinates

Draw convex polytope



Triangulating with Five Vertices

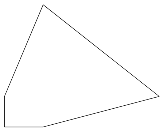
- Add $-\log|c|$ as third row to support $\mathcal{A}$, where $c = [c_1, c_2, c_3, c_4, c_5]$
- Compute convex hull of lifted support
- Determine which triangle faces have positive inner normals

Program: Isotopy Type (part 1: Triangulations)

Recall e.g. $\mathcal{A} = \begin{bmatrix} 0 & 1 & 0 & 4 & 1 \\ 0 & 0 & 1 & 1 & 4 \end{bmatrix}$

Plot columns as coordinates

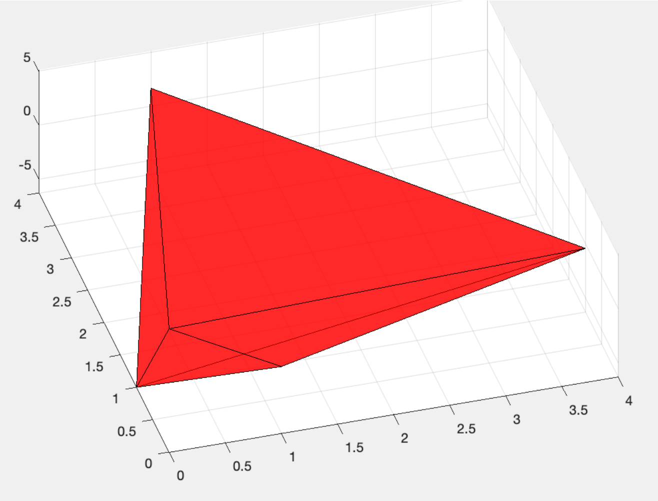
Draw convex polytope



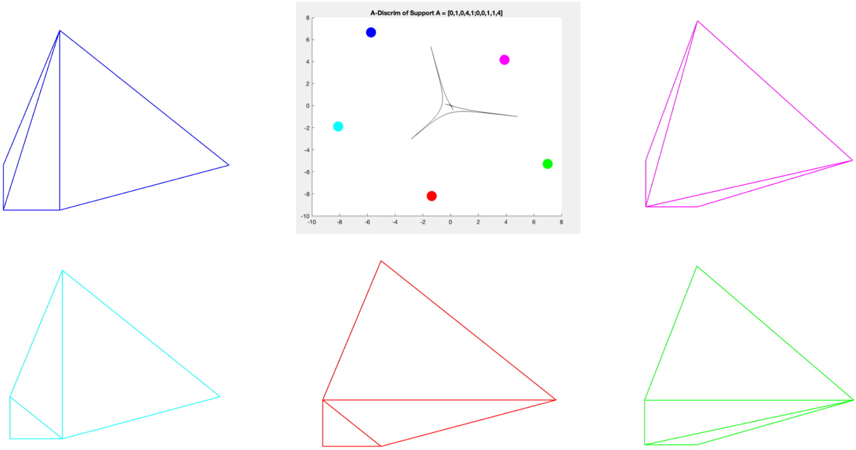
Triangulating with Five Vertices

- Add $-\log|c|$ as third row to support $\mathcal{A}$, where $c = [c_1, c_2, c_3, c_4, c_5]$
- Compute convex hull of lifted support
- Determine which triangle faces have positive inner normals
- These triangles form triangulation

Lifted Triangulation Example



Triangulations of $f(x) = c_1 + c_2x + c_3y + c_4x^4y + c_5xy^4$



Program: Isotopy Type (part 2: Viro Patchworking)

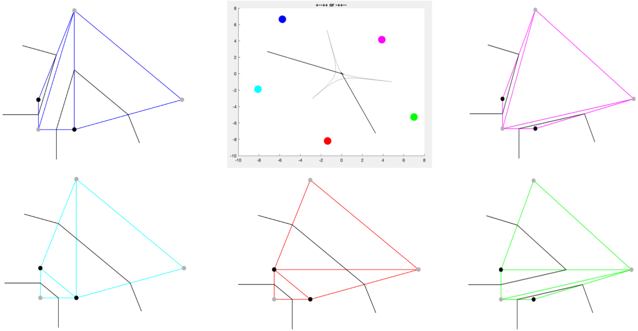
Label signs of vertices, note edges with vertices of opposite signs

- Outer edges: draw outer normals
- Inner edges: connect to through adjacent triangles

Program: Isotopy Type (part 2: Viro Patchworking)

Label signs of vertices, note edges with vertices of opposite signs

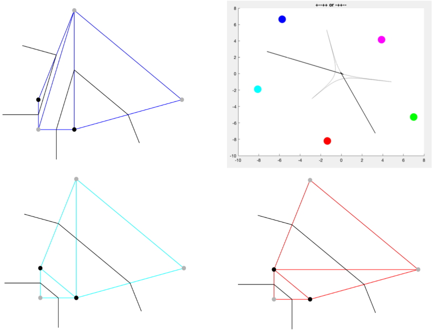
- Outer edges: draw outer normals
- Inner edges: connect to through adjacent triangles



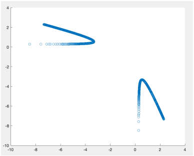
Program: Isotopy Type (part 2: Viro Patchworking)

Label signs of vertices, note edges with vertices of opposite signs

- Outer edges: draw outer normals
- Inner edges: connect to through adjacent triangles



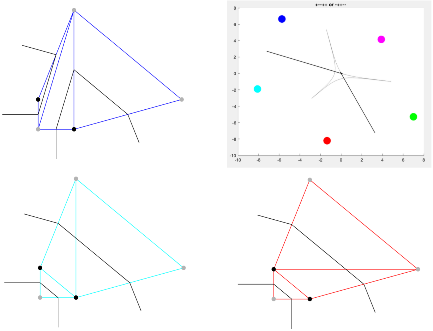
Recall Actual Zero Sets



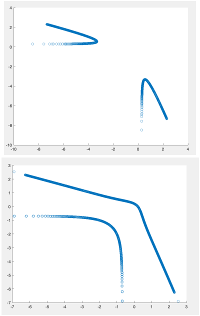
Program: Isotopy Type (part 2: Viro Patchworking)

Label signs of vertices, note edges with vertices of opposite signs

- Outer edges: draw outer normals
- Inner edges: connect to through adjacent triangles



Recall Actual Zero Sets



Which side of the signed contour are my coefficients in?

- Plotting the point with the signed contour is visually trivial, but parametrization prevents us from using inequalities to determine the sidedness.

Which side of the signed contour are my coefficients in?

- Plotting the point with the signed contour is visually trivial, but parametrization prevents us from using inequalities to determine the sidedness.
- Ellen Chlachidze (2022) developed a more efficient approximation involving a simpler inequality
  - curve based on the directions of the signed contour as it extends to infinity (let these infinite directions be called rays)

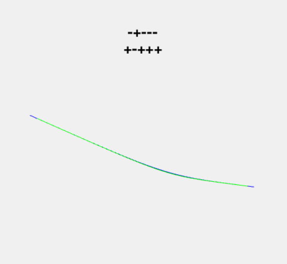
Which side of the signed contour are my coefficients in?

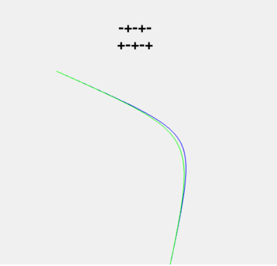
- Plotting the point with the signed contour is visually trivial, but parametrization prevents us from using inequalities to determine the sidedness.
- Ellen Chlachidze (2022) developed a more efficient approximation involving a simpler inequality
  - curve based on the directions of the signed contour as it extends to infinity (let these infinite directions be called rays)
- Problem: This approximation fails if the signed contour has a cusp.

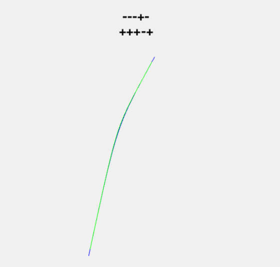
Which side of the signed contour are my coefficients in?

- Plotting the point with the signed contour is visually trivial, but parametrization prevents us from using inequalities to determine the sidedness.
- Ellen Chlachidze (2022) developed a more efficient approximation involving a simpler inequality
  - curve based on the directions of the signed contour as it extends to infinity (let these infinite directions be called rays)
- Problem: This approximation fails if the signed contour has a cusp.

$$\mathcal{A} = \begin{bmatrix} 0 & 1 & 2 & 1 & 4 \\ 0 & 2 & 1 & 4 & 1 \end{bmatrix}$$







$$\mathcal{A} = \begin{bmatrix} 0 & 1 & 2 & 1 & 4 \\ 0 & 2 & 1 & 4 & 1 \end{bmatrix}$$

Ideas for Approximating Signed Contours with 1 Cusp

Ideas for Approximating Signed Contours with 1 Cusp

1. Use two of Ellen's approximations: ray 1 to cusp, cusp to ray 2

Ideas for Approximating Signed Contours with 1 Cusp

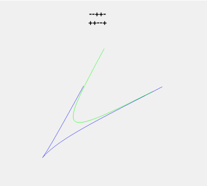
1. Use two of Ellen's approximations: ray 1 to cusp, cusp to ray 2
2. Map a simpler $\mathcal{A}$-discriminant that contains a cusp onto our cusp:

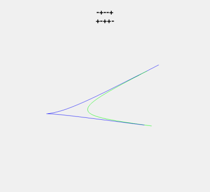
Ideas for Approximating Signed Contours with 1 Cusp

1. Use two of Ellen's approximations: ray 1 to cusp, cusp to ray 2
2. Map a simpler $\mathcal{A}$-discriminant that contains a cusp onto our cusp:
   - $\mathcal{A}$-discriminant of cubic (Support $= [0, 1, 2, 3]$) has a cusp
   - Solving for sidedness of the cubic A-discriminant (not parametrization) is better

Results: Approximating Signed Contours with Cusps

Ellen's Approximations

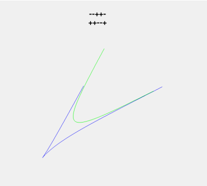


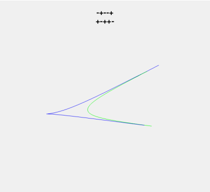


$$\mathcal{A} = \begin{bmatrix} 0 & 1 & 2 & 1 & 4 \\ 0 & 2 & 1 & 4 & 1 \end{bmatrix}$$

Results: Approximating Signed Contours with Cusps

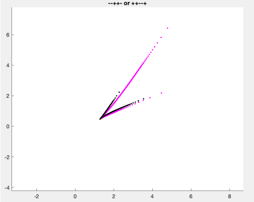
Ellen's Approximations

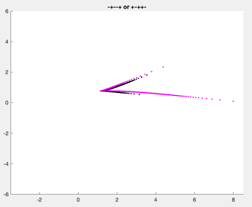




$$\mathcal{A} = \begin{bmatrix} 0 & 1 & 2 & 1 & 4 \\ 0 & 2 & 1 & 4 & 1 \end{bmatrix}$$

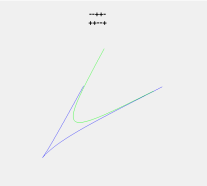
Two of Ellen's Approximations

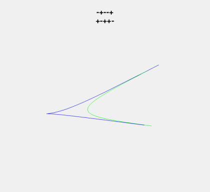




Results: Approximating Signed Contours with Cusps

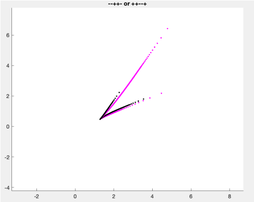
Ellen's Approximations

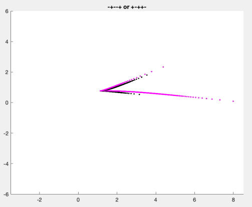




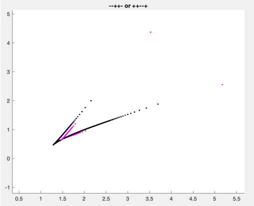
$$\mathcal{A} = \begin{bmatrix} 0 & 1 & 2 & 1 & 4 \\ 0 & 2 & 1 & 4 & 1 \end{bmatrix}$$

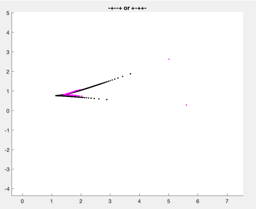
Two of Ellen's Approximations





Mapping Cubic Cusp





Thank you!