Numerically decomposing complex and real tropical curves in any number of dimensions

Daniel Brake

14 July, 2016

Joint work with
Jon Hauenstein
and
Cynthia Vinzant

UNIVERSITY OF NOTRE DAME
College of Science

Outline

Tropicalization

The tropicalization of an equidimensional algebraic variety is typically defined for complex components [G].

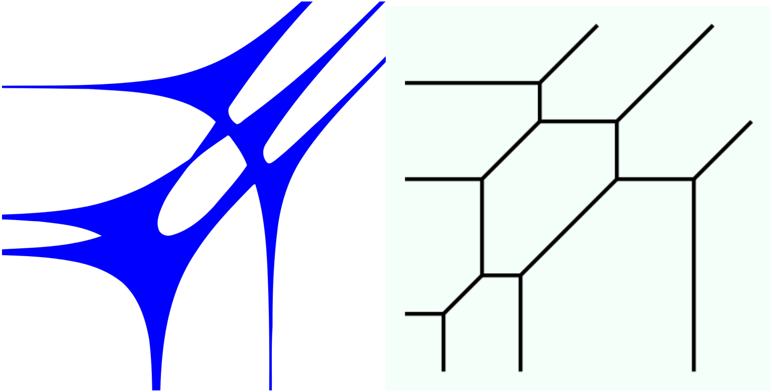
- For complex plane curves, can be defined as limits of ‘amoebas’:

$$\lim_{t\to 0}\left(-\frac{\log|z_1|}{\log t}, -\frac{\log|z_2|}{\log t}\right)$$

- Can also be defined in terms of **Puiseux series**
- Can also be defined in terms of the **max-plus semiring**

Gathmann. Tropical Algebraic Geometry. arXiv:math/0601322v1

Amoeba



A complex amoeba, and its limit of

$$P(z,w) = 50z^3+83z^2w+24zw^2+w^3+392z^2+414zw+50w^2-28z+59w-100$$

Existing methods for computing Tropical Curves

There are implemented ways to compute $\mathbb{C}$ tropical curves:

- `Gfan`, Gröbner bases.
- Homotopy Continuation [JLY], amoebas.
  1. Find possible rays by sampling tentacles,
  2. Compute multiplicities of rays,
  3. Check for completeness.

Neither above method computes the $\mathbb{R}$ tropicalization (or tropicalization of a $\mathbb{R}$ curve), so we will explore what this means, and how to compute it.

Jensen, Leykin, Yu. Computing Tropical Curves Via Homotopy Continuation. arXiv:1408.3105v1

Tropical rays

We represent tropical curves as a collection of **rays**. Think of the rays as integer vectors with equivalence classes.

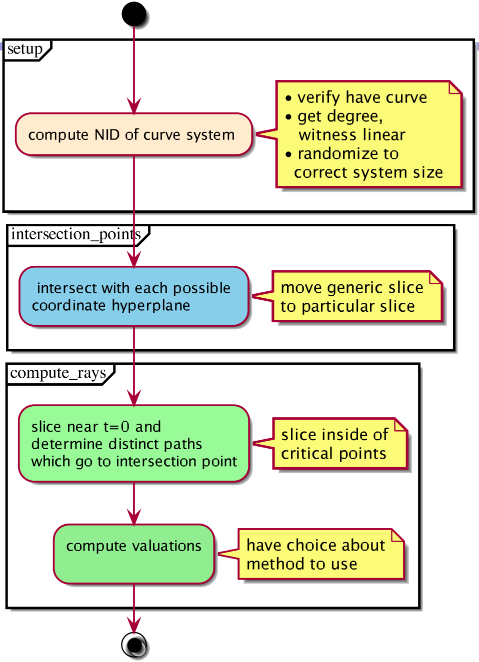
- The entries in the rays are **valuations of Puiseux series**.
- Each branch of the curve passing through **each intersection point** with a coordinate hyperplane has its own ray, and its contribution to the **multiplicity** (complex only, ? for real)
- Valuations are **leading coefficients** of Puisuex series. We compute by **Cauchy's integral formula**.

For complex curves, we compute rays with multiplicity, but not arrangements. For real curves we get a neat arrangement.

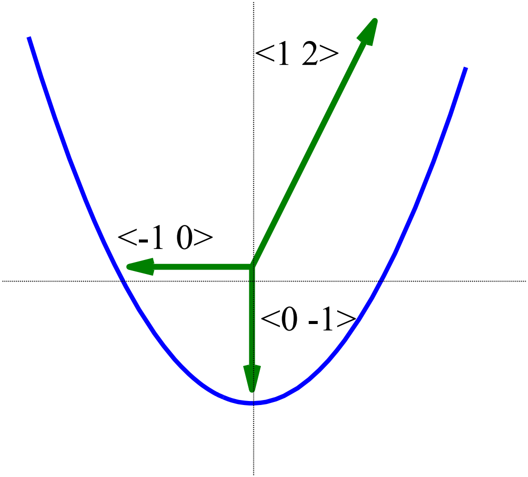
`https://arxiv.org/abs/1605.04203`

Outline

High level


setup
compute NID of curve system
• verify have curve
• get degree, witness linear
• randomize to correct system size
intersection_points
intersect with each possible coordinate hyperplane
move generic slice to particular slice
compute_rays
slice near t=0 and determine distinct paths which go to intersection point
slice inside of critical points
compute valuations
have choice about method to use

Net result



Setup

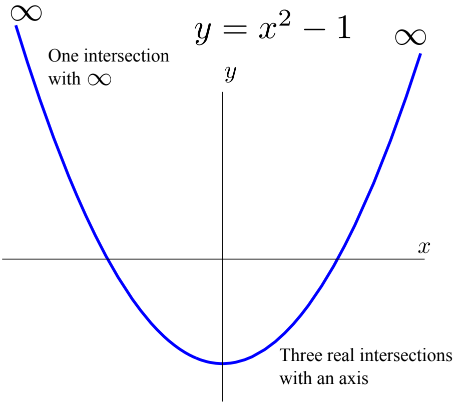
Since our method uses NAG, we turn to the *witness set* as the fundamental unit from which to compute. Let $\mathcal{W}$ denote the witness set. $\mathcal{W} = \{W, \mathcal{L}, f\}$:

- W, witness points. Degree-many of them. $W = \mathcal{V}(f) \cap \mathcal{V}(\mathcal{L})$
- $\mathcal{L}$, witness linears. Dimension-many of them.
- f, system. Possibly randomized.

This is computed using Bertini's `tracktype: 1;`.

Additionally, you must projectivize the curve.
This means, ensure each term has same degree, by multiplying by a new variable (h? z?). Also, randomly linearly patch the system, bringing the point at ∞ to a real finite point, $h = 0$.

A $\mathbb{R}$ example – a parabola

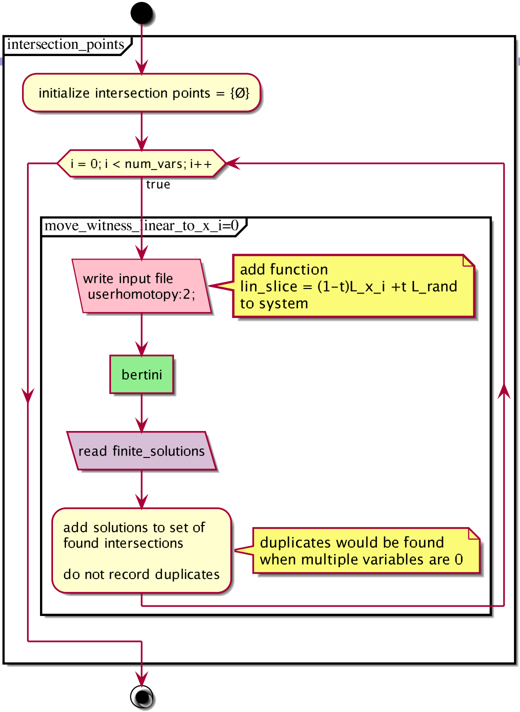


Projectivizing the parabola lets us compute the behaviour at ∞, when we supplement with a random real patch.

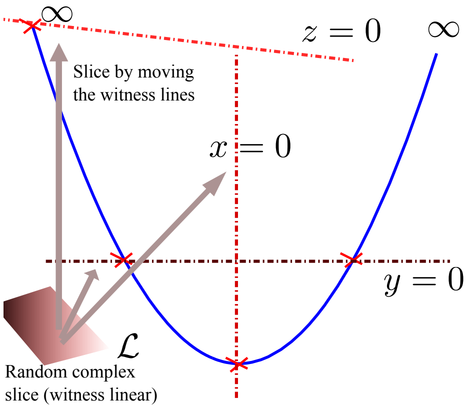
Setup – example

```
CONFIG
TrackType : 1;
END;
INPUT
variable_group h, x, y;

function f1, patch;
f1 = x^2 - y*h - h^2;
constant px,py,ph;
px = 0.0528518678;
py = -0.729144193;
ph = -0.288589658;
patch = px*x + py*y
  + ph*h - 1;
END;
```

```
bertini

...

******* Decomposition by Degree *

Dimension 1: 1 classified component
-----------------------------------
   degree 2: 1 component

*********************************
```


intersection_points
initialize intersection points = {Ø}
i = 0; i < num_vars; i++
true
move_witness_linear_to_x_i=0
write input file
userhomotopy:2;
add function
lin_slice = (1-t)L_x_i +t L_rand
to system
bertini
read finite_solutions
add solutions to set of
found intersections
do not record duplicates
duplicates would be found
when multiple variables are 0

Coordinate intersections – example



Slicing the curve at each coordinate tells us whether to do further computations on that coordinate.

Endgame Operating Zone

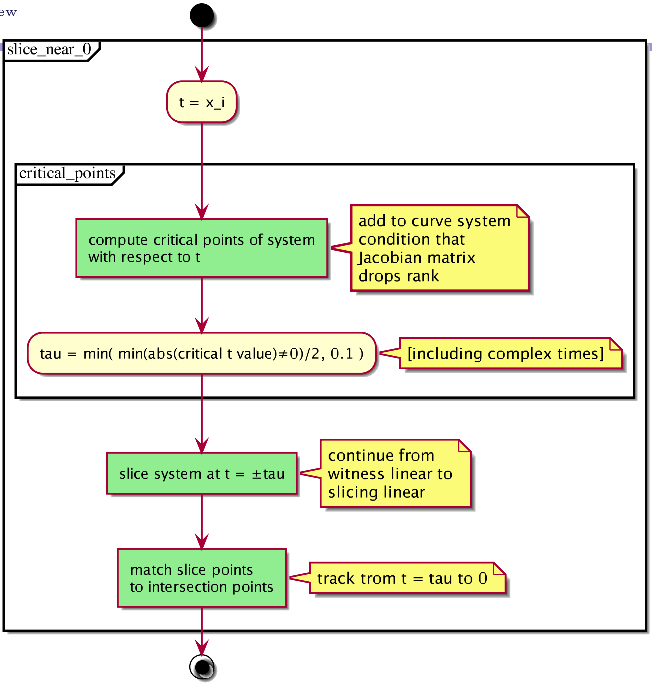
Slice *near* the intersection points. What does *near* mean?

- For curves, *near* means inside the Endgame Operating Zone – a distance defined by critical points with respect to the $x_i = t$ parameterization of the curve.
- Hence, we compute the critical points.

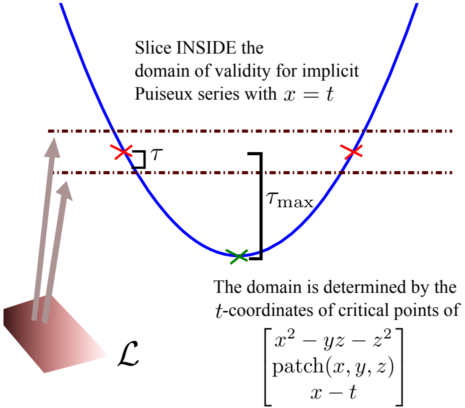
$$\begin{bmatrix} f \\ \begin{bmatrix} Jf \\ Jx_i \end{bmatrix} \cdot v \end{bmatrix} = 0$$

- This is the most computationally expensive part of the algorithm.

Important optimization – do this step once for generic parameterization, then parameter homotopy to each particular one.


slice_near_0
t = x_i
critical_points
compute critical points of system with respect to t
add to curve system condition that Jacobian matrix drops rank
tau = min(min(abs(critical t value)≠0)/2, 0.1)
[including complex times]
slice system at t = ±tau
continue from witness linear to slicing linear
match slice points to intersection points
track trom t = tau to 0

EOZ – example



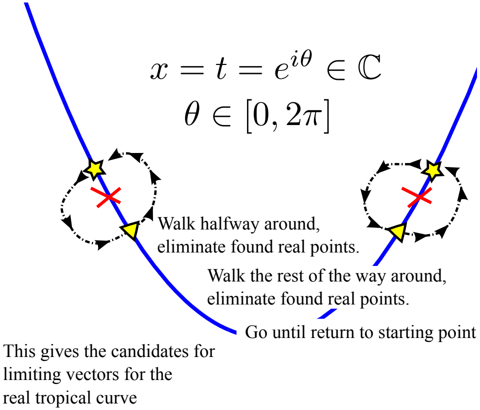
Get the $\mathbb{R}$ points for Puiseux series computation

Monodromy

We need to determine **cycle numbers** c for each path.

- Walk in $\mathbb{C}$ circles around $x_i = t = 0$.
- The # of monodromy loops it takes to return back to same point is c.
- Amazingly, c is the number which turns Puiseux series into Power series!
- Implemented in Bertini_tropical using `userhomotopy`.

Monodromy – example



Doing monodromy loops also determines the distinct paths limiting to the real intersection points.

Valuations

We compute valuations (for each j) by Cauchy's integral formula:

$$\frac{f^{(k)}(0)}{k!} = \frac{1}{2\pi i}\int_\gamma \frac{f(z)}{z^{k+1}}\,dz = \frac{1}{2\pi}\int_0^{2\pi} \frac{f(e^{i\theta})}{(e^{i\theta})^k}\,d\theta$$

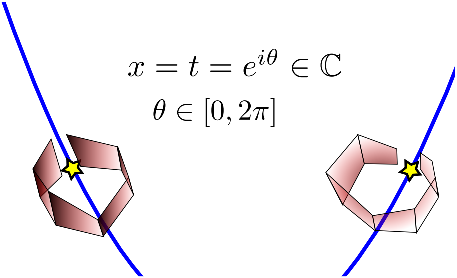
The right formula is what is actually used, with

- $z = t$, and
- $f(z) = x_j(t)$

Where do we get the data for the numerical approximation?

- From the monodromy loops! Two birds!
- Use the exponentially convergent trapezoid rule for periodic function.

Valuations – example

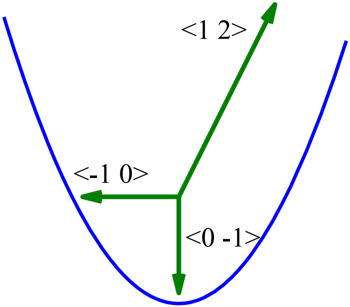


For each variable, find the first k such that

$$\frac{k!}{2\pi} \int_0^{2\pi} \frac{x_j(\theta)}{(e^{i\theta})^k}\, d\theta \neq 0$$

Find the **valuations** (power for first nonzero coefficient of Puiseux series) for each variable, by estimating Cauchy integrals via the trapezoid rule.

Completion

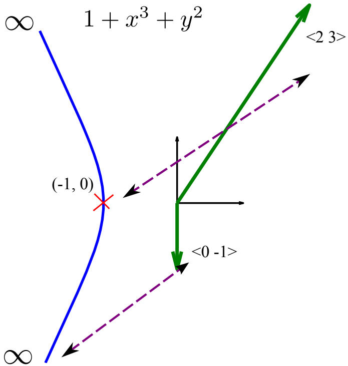


After dehomogenization, we get this picture for the real tropicalization of this parabola

Repeating the steps for each coordinate axis, we get the real tropicalization of this parabola.

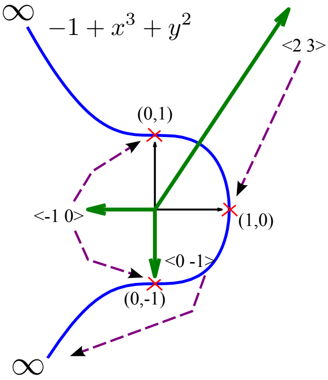
Outline

Elliptic Curve



Sometimes the real curve does not generate all the vectors of the complex tropicalization.

Elliptic Curve 2



The values of coefficients and parameters matter greatly!

Outline

The software implementation

Overview

- Currently available at `http://danielthebrake.org/tropical.html`
- GPL3
- Implemented in Matlab, using Bertini for path tracking
- Storage of all computed data
- Capable of re-running just part of a decomposition
- Visualization options

Inputs

- Main input is a Bertini `input` file.
- Also, takes in completed `tracktype: 1` run, the NID.
- Bertini settings are selected in the input file.
- Decomposition settings are selected in Matlab.
  - real or complex
  - zero valuation threshold
  - default slice value
  - uniqueness distance thresholds
  - and more

How to call

The decomposition algorithm is called when a member of class `tropical_curve` is created.

```
T = tropical_curve(...)
```

is all it takes!

You can manually call the free sub functions, but why make things more complicated?

Output

The main output is a member of class `tropical_curve`.
Important properties include

- `rays` – The computed valuations
- `multiplicities` – The full multiplicities of each ray
- `aux_data` – Holding much computed auxiliary data, such as
  - intersection points,
  - walked Cauchy paths, and
  - Puiseux coefficients.

Completed decomposition is automatically saved to a `.mat` file, date stamped.

Challenges

Hard parts of computing tropical curves numerically include

- Balancing the art of choosing tracking tolerance and settings with compute time.
- Zero-threshold for Puiseux coefficients
- Scaling of systems can be hard, too. Contributes to threshold problems. Scaling of both functions and variables matters
- Other settings, like how to tell if two points are the same, and the number of data points for Puiseux coefficient computation.

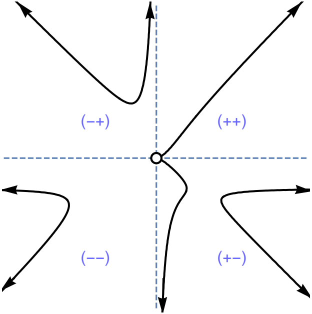
Partial re-computation

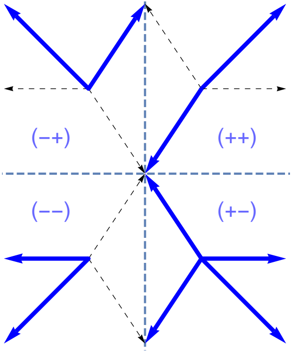
Re-computing an entire decomposition having changed a single setting could be brutal. Hence,

- Saves file output to disk for each step.
- Re-loads saved data when possible (uses file hashes to detect changes in settings)
- Can resume a partial decomposition, or just compute those steps which come after a changed setting.
- Can manually re-run steps, if not satisfied with automatic or default settings.

Real arrangments

Real tropical curve decompositions can be arranged according to orthant sign for path intersections.

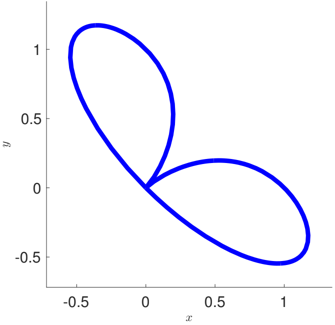


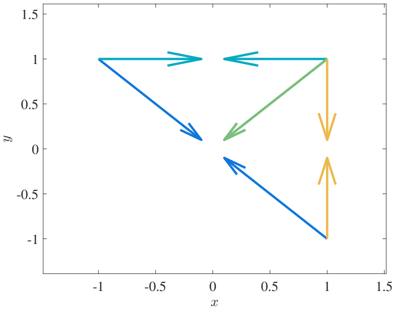


Outline

Quartic

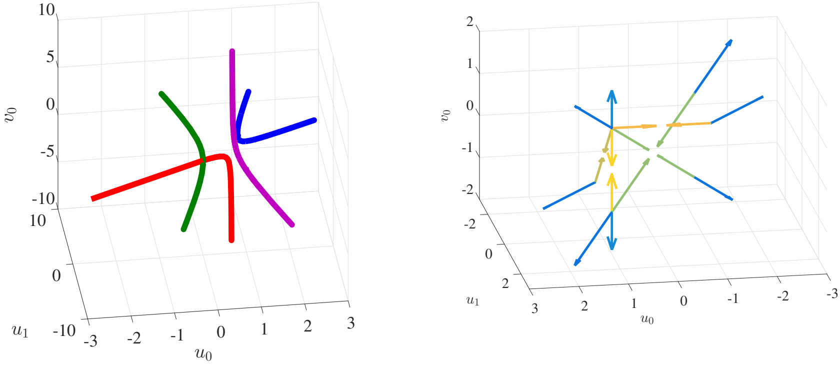
$$x^4 + y^4 - (x-y)^2(x+y)$$





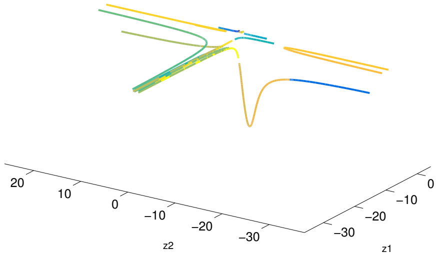
Central curve of linear program

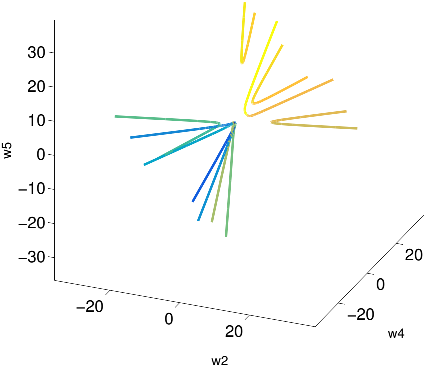
- 18+1 variables
- $\mathbb{C} == \mathbb{R}$, probably rare
- Had scaling problems



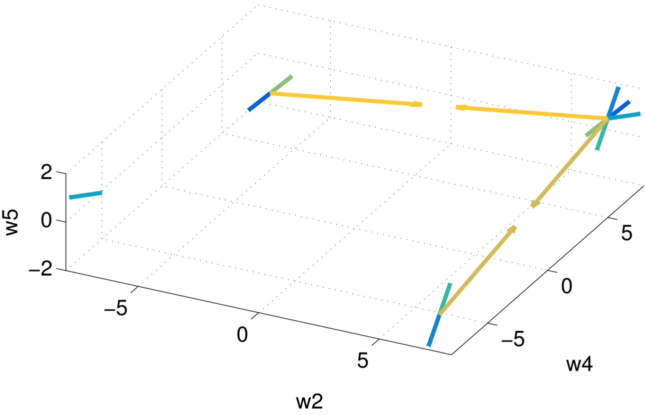
A-polynomial of a knot

- 10+1 variables
- Main example from [JLY]
- $\mathbb{C}! = \mathbb{R}$





A-polynomial of a knot



Outline

Notes

- Valuations can be computed with either
  - the limit definition,
$$\lim_{t\to 0} \frac{|z(t)|}{|t|}$$
  - Cauchy integrals,
$$C\int_{\gamma} \frac{z(t)}{t^k}\,dt$$
- Since monodromy loops are periodic, the trapezoid rule for Cauchy integration converges quickly.
- Since one must determine which slice points connect at $\cap$ points, and one can use monodromy for this, the Cauchy valuation method is superior.
- Our method computes the vectors, but not their complex arrangements as one gets with amoebas. We can arrange them for real curves.

Contrasts between $\mathbb{R}$ and $\mathbb{C}$ Tropicalization

$\mathbb{C}$

- Generic behaviour – largely doesn't depend on parameter values.
- Multiplicities can be seen as total # of Puiseux series at intersections with coordinate hyperplanes.
- Balancing condition to check for correct computation. Sum of all vectors $= 0$.

$\mathbb{R}$

- Non-generic behaviour – strongly depends on parameter values.
- Not clear yet how multiplicity is defined.
- No balancing condition. Unknown how to check for correct computation.

Thank you for your kind attention

https://arxiv.org/abs/1605.04203