DTU

Algorithmic Algebraic Geometry

Project

Find an efficient algorithm to speed up real root counting for univariate tetranomials with high probability. Approach will be by approximating A-discriminant contours in a new way.

Why do we care about discriminant varieties?

Why do we care about discriminant varieties?

- **Root Behavior:** the discriminant variety is the set of coefficients that define a polynomial with degenerate roots.

Why do we care about discriminant varieties?

- **Root Behavior:** the discriminant variety is the set of coefficients that define a polynomial with degenerate roots.
- **Degenerate Roots:** Degenerate roots help describe transitions in number of real roots and closeness to degeneracy governs hardness of numerical solving.

Why do we care about discriminant varieties?

- **Root Behavior:** the discriminant variety is the set of coefficients that define a polynomial with degenerate roots.
- **Degenerate Roots:** Degenerate roots help describe transitions in number of real roots and closeness to degeneracy governs hardness of numerical solving.
- **Topological Behavior:** More generally, degenerate roots describe transitions in the isotopy type of a (varying) real algebraic surface.

Real polynomial systems occur in applications like...

Real polynomial systems occur in applications like...

- **Algebraic Statistics:** Where there is an unknown probabilistic model, and you are solving for some hidden probabilities that govern the model. This entails solving polynomial systems for roots in the interval [0,1].

Real polynomial systems occur in applications like...

- **Algebraic Statistics:** Where there is an unknown probabilistic model, and you are solving for some hidden probabilities that govern the model. This entails solving polynomial systems for roots in the interval [0,1].
- **Computational Biochemistry:** Where you are trying to predict possible equilibrium concentrations for certain compounds in a complicated chemical reaction. Here, one usually finds sparse polynomial systems in many variables, but of low degree.

Real polynomial systems occur in applications like...

- **Algebraic Statistics:** Where there is an unknown probabilistic model, and you are solving for some hidden probabilities that govern the model. This entails solving polynomial systems for roots in the interval [0,1].
- **Computational Biochemistry:** Where you are trying to predict possible equilibrium concentrations for certain compounds in a complicated chemical reaction. Here, one usually finds sparse polynomial systems in many variables, but of low degree.
- **Discretizing Partial Differential Equations:** In certain physical modelling problems, one is trying to approximate the solutions of a very complicated differential equation. So one then uses a numerical scheme to approximate the solution, and this usually involves expanding into a basis of polynomials. Getting information about the solution a PDE can then be reduced to solving a structured polynomial system, many times, with varying coefficients, over the real numbers.

Easy Example

Using the following tetranomial:

$$c_0 + c_1 x + c_2 x^2 + c_3 x^3 \tag{1}$$

Easy Example

Using the following tetranomial:

$$c_0 + c_1 x + c_2 x^2 + c_3 x^3 \tag{1}$$

yields a manageable discriminant:

$$-27 c_0^2 c_3^2 + 18 c_0 c_1 c_2 c_3 - 4 c_0 c_2^3 - 4 c_1^3 c_3 + c_1^2 c_2^2$$

Harder Example

Using a nastier tetranomial:

$$c_0 + c_1 x^3 + c_2 x^5 + c_3 x^{19} \tag{2}$$

Harder Example

Using a nastier tetranomial:

$$c_0 + c_1 x^3 + c_2 x^5 + c_3 x^{19} \tag{2}$$

yields a nastier result!:

$$1978419655660313589123979\, c_0^{16} c_3^5 + 6093825838807983035604992 c_0^{12} c_1^3 c_2^2 c_3^4 - 4166308590611436407824000 c_0^{10} c_1 c_2^7 c_3^3 + 4136784303514917397331968 c_0^8 c_1^6 c_2^4 c_3^3 - 1680626254018160036413440 c_0^6 c_1^{11} c_2 c_3^3 + 5465538956966243292282880 c_0^6 c_1^4 c_2^9 c_3^2 + 3040596925589240487608320 c_0^4 c_1^9 c_2^6 c_3^2 + 9103573347707241984000 c_0^4 c_1^2 c_2^{14} c_3 + 2441097252432707688857600 c_0^2 c_1^{14} c_2^3 c_3^2 - 1103132840914428362752 c_0^2 c_1^7 c_2^{11} c_3 + 34725021329868800000 c_0^2 c_2^{19} + 4980620899901578936320 c_1^{19} c_3^2 - 4889673564157063987200 c_1^{12} c_2^8 c_3 + 12000967371602657280 c_1^5 c_2^{16}$$

Moving Forward...

We need a better way to plot the zero sets of complicated polynomials!
We will use the clever Horn-Kapranov Uniformization to reduce the dimension of the parameter space!

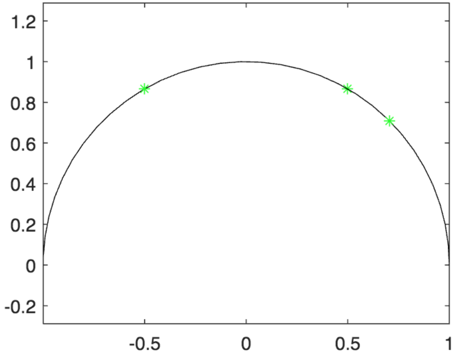
Horn-Kapranov Uniformaization

A way to efficiently parameterize discriminant varieties. For $A = \begin{bmatrix} a_1 & a_2 & a_3 & a_4 \end{bmatrix}$, let $\hat{A}$ be the 2x4 matrix defined by appending a row of 1s to the top of A and let B in $\mathbb{Z}^{4x2}$ be any matrix whose columns form a basis for the right nullspace of $\hat{A}$. Then the (logarithmic, reduced) Horn-Kapranov Uniformization for A is the function

$$\xi_A([\lambda_1 : \lambda_2]) := (Log|[\lambda_1, \lambda_2]B^T|)B$$

which defines a map from $\mathbb{P}^1_{\mathbb{R}}$ to $\mathbb{R}^2$.

Horn-Kapranov Uniformization II



For nicer plots, we use: $(\lambda_1 : \lambda_2) = (cos\theta, sin\theta)$
This brings our plots from: $[\lambda_1 : \lambda_2] \in P_R^1$
to: $(\lambda_1 : \lambda_2) \in$ Unit Semi-Circle.

Amoeba

If f is any polynomial in $C[x_1, \ldots, x_n]$ then its amoeba is the set

$$\{(\log|x_1|, \ldots, \log|x_n|) \mid f(x_1, \ldots, x_n) = 0 \ , \ x_i \in \mathbb{C} \setminus \{0\}\}$$

.

Amoeba

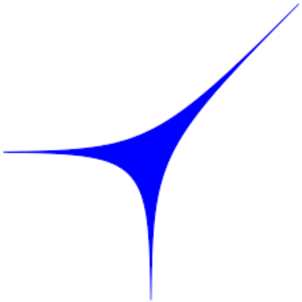


Figure: This is the Ameoba for $1 + x_1 + x_2$.

Image obtained from: https://en.wikipedia.org/wiki/Amoeba$_($*mathematics*)

Complexity of Amoeba Membership

Complexity of Amoeba Membership

- Amoebae look cool but are not trivial to draw.

Complexity of Amoeba Membership

- Amoebae look cool but are not trivial to draw.
- Deciding if a rational point (x,y) lies in a 2-dimensional amoeba is already NP-hard!

Plaisted, 1984; *Avendano*, *Kogan*, *Rojas*, *Rusek*, 2013

Complexity of Amoeba Membership

- Amoebae look cool but are not trivial to draw.
- Deciding if a rational point (x,y) lies in a 2-dimensional amoeba is already NP-hard!

Plaisted, 1984; *Avendano*, *Kogan*, *Rojas*, *Rusek*, 2013

- The boundary of the last amoeba is defined by the graphs of “simple” transcendental function, e.g., $y = Log(1 + e^x)$.

Complexity of Amoeba Membership

- Amoebae look cool but are not trivial to draw.
- Deciding if a rational point (x,y) lies in a 2-dimensional amoeba is already NP-hard!

Plaisted, 1984; *Avendano*, *Kogan*, *Rojas*, *Rusek*, 2013

- The boundary of the last amoeba is defined by the graphs of “simple” transcendental function, e.g., $y = Log(1 + e^x)$.
- Deciding if a rational point lies on or near such a curve gets us into interesting problems involving Diophantine approximation!

What to do about Amoeba membership?!

What to do about Amoeba membership?!

- Alternative: Approximate each amoeba boundary curve by a piecewise linear curve.

What to do about Amoeba membership?!

- Alternative: Approximate each amoeba boundary curve by a piecewise linear curve.
- Such curves can be extracted from the Horn-Kapranov Uniformization.

What to do about Amoeba membership?!

- Alternative: Approximate each amoeba boundary curve by a piecewise linear curve.
- Such curves can be extracted from the Horn-Kapranov Uniformization.
- Do they work well with random polynomials/points?

What to do about Amoeba membership?!

- Alternative: Approximate each amoeba boundary curve by a piecewise linear curve.
- Such curves can be extracted from the Horn-Kapranov Uniformization.
- Do they work well with random polynomials/points?
- Experiments show: So-so...

Experimentation!!!

The ultimate goal of our experimentation is to understand how well tropical discriminant chambers approximate true sign chambers.

Matlab Code Round 1

Matlab Code Round 1

- Looks at a specific chamber of the amoeba

Matlab Code Round 1

- Looks at a specific chamber of the amoeba
- Uses a tropical approximation of the curve to check which side of the curve (and by proxy, the coefficient space) a point resides in

Matlab Code Round 1

- Looks at a specific chamber of the amoeba
- Uses a tropical approximation of the curve to check which side of the curve (and by proxy, the coefficient space) a point resides in
- Tests a set number of i.i.d. random points to see if they are within that chamber

Matlab Code Round 1

- Looks at a specific chamber of the amoeba
- Uses a tropical approximation of the curve to check which side of the curve (and by proxy, the coefficient space) a point resides in
- Tests a set number of i.i.d. random points to see if they are within that chamber
- Yields an accuracy percentage

Using a Tropical, Linear Approximation:

We use the piecewise function y=0 and y=x to approximate the curve of the amoeba.

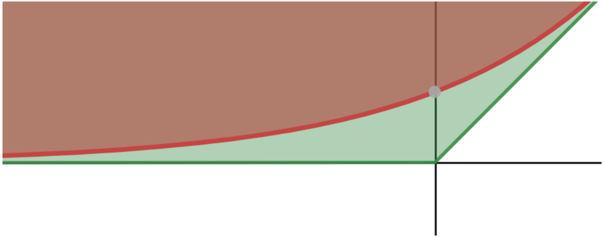


Figure: $y >= log(1 + e^x)$ and $y >= 0$ and $y >= x$

Results are so-so:

Testing 1000 i.i.d. random points:

Trials:	1	2	3	4	5
%:	62%	65%	63%	60%	65%

Testing 10,000 i.i.d. random points:

Trials:	1	2	3	4	5
%:	63%	63%	65%	64%	64%

Testing 100,000 i.i.d. random points:

Trials:	1	2	3	4	5
%:	64%	63%	64%	64%	64%

Complexity Issue for Chamber Membership

- Deciding a polynomial inequality, involving a polynomial of degree d in n variables with coefficients all of absolute value $<= H$, at an input rational point $p = (a_1/b_1, ..., a_n/b_n)$, is a highly non-trivial problem!

Complexity Issue for Chamber Membership

- Deciding a polynomial inequality, involving a polynomial of degree d in n variables with coefficients all of absolute value $<= H$, at an input rational point $p = (a_1/b_1, ..., a_n/b_n)$, is a highly non-trivial problem!
- We will use a little trick to get around this! We will change x and y to logarithmic values to yield a more manageable equation to test our inequalities.

Alternative: Simplified Horn-Kapranov

Alternative: Simplified Horn-Kapranov

- The arcs of the discriminant contour are defined by linear combinations of logarithms.

Alternative: Simplified Horn-Kapranov

- The arcs of the discriminant contour are defined by linear combinations of logarithms.
- Approximate each arc by just 2 logarithms: This should also yield easier Diophantine approximation.

Matlab Code Round 2

Matlab Code Round 2

- Exponent values are plugged in

Matlab Code Round 2

- Exponent values are plugged in
- Code runs through two loops to apply the Horn-Kapranov Uniformization

Matlab Code Round 2

- Exponent values are plugged in
- Code runs through two loops to apply the Horn-Kapranov Uniformization
- The associated amoeba to the polynomial is given along with each quadrant

Example:

We look at a family of polynomials

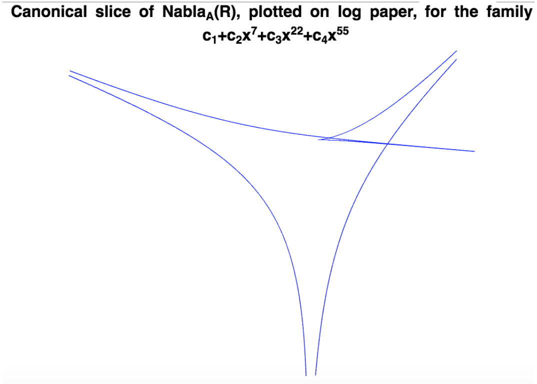
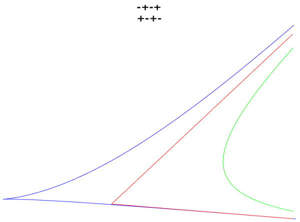
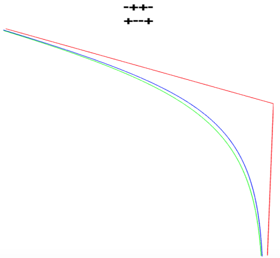
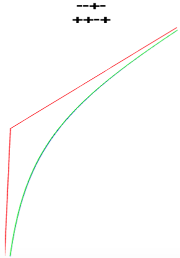


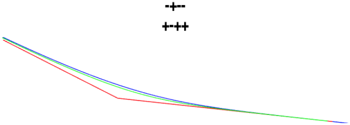
Figure: This is the Ameoba for $c_1 + c_2x^7 + c_3x^{22} + c_4x^{55}$.

Much Closer Approximations (for the most part):









Testing the Coefficient Space

Testing the Coefficient Space

- Setting the coefficients of the polynomial plots a point in the quadrant!

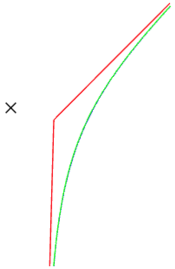
Testing the Coefficient Space

- Setting the coefficients of the polynomial plots a point in the quadrant!
- This shows us where the polynomial lies in coefficient space!

Plotting Polynomials as Points:

--+-

++-+



$$-1 - x^{7} + x^{22} - x^{55}$$

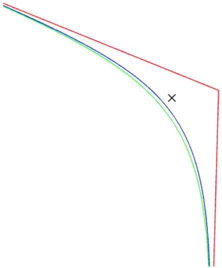
--+-

++-+

$$-2 - 2x^{7} + 10x^{22} - x^{55}$$

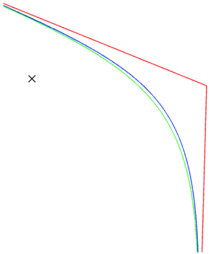
More Points:

-++-
+--+



$-1 + x^7 + x^{22} - x^{55}$

-++-
+--+



$-20 + 10x^7 + 2x^{22} - 10x^{55}$

Obtaining a Real Root Count:

Plugging the previous polynomial examples into Maple will give the real roots.

Results:

$f1 := -1 - x^7 + x^{22} - x^{55};$

$realroot(f1);$

$$f1 := -x^{55} + x^{22} - x^7 - 1$$

$$\left[\left[-\frac{30953}{32768}, -\frac{61903}{65536}\right]\right]$$

$f2 := -2 - 2 \cdot x^7 + 10 \cdot x^{22} - x^{55};$

$realroot(f2);$

$$f2 := -x^{55} + 10\,x^{22} - 2\,x^7 - 2$$

$$\left[\left[-\frac{118191}{131072}, -\frac{472761}{524288}\right], \left[\frac{61}{64}, \frac{977}{1024}\right], \left[\frac{139993}{131072}, \frac{279989}{262144}\right]\right]$$

$f3 := -1 + x^7 + x^{22} - x^{55};$

$realroot(f3);$

$$f3 := -x^{55} + x^{22} + x^7 - 1$$

$$\left[[-1, -1], \left[\frac{125029}{131072}, \frac{250335}{262144}\right], [1, 1]\right]$$

$f4 := -10 + 2\,x^7 + 10\,x^{22} - 20\,x^{55};$

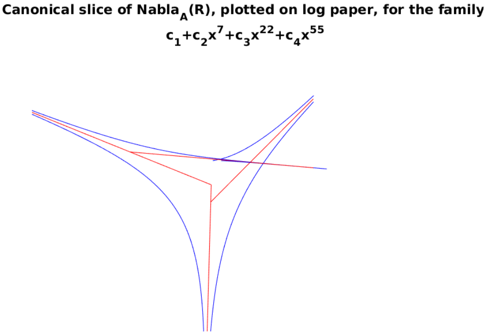
$realroot(f4)$

$$f4 := -20\,x^{55} + 10\,x^{22} + 2\,x^7 - 10$$

$$\left[\left[-\frac{1001}{1024}, -\frac{125}{128}\right]\right]$$

Figure: Using Maple software

Now we can see which region future polynomials lie in which will give us the number of real roots!



With many thanks...

- Thank you Dr. Rojas for the Matlab code!
- Thank you to Dr. Rojas and TA Joshua Goldstein for the guidance!
- Thank you to the NSF and Texas A & M for making this research experience possible!