Geometry of $\mathbb{R}$ Roots of 9×9 Polynomial Systems

Lacey Eagan & Luis Feliciano

July 17, 2018

Abstract

Chemical reactions are constantly taking place within every single animal and plant cell, and these reactions can be modeled by Chemical Reaction Networks. The particular chemical reaction that we chose to study through Chemical Reaction Networks is Phosphorylation. After modeling Phosphorylation, we use Mass Action Kinetics to derive nine reaction rate equations with nine unknowns which we can then solve to find the equilibria of our Chemical Reaction Network. We are interested in the equilibria because we would like to see when they take place and under what conditions. After reducing our 9×9 system to a much simpler 2×2 system, we apply A-Discriminants to obtain a better idea of what our roots will look like. Discriminant Varieties divide the coefficient space into regions where the underlying number of real roots is constant, and we use Linear Programming to see which coefficient sign vectors yield simpler root counting. Solving for the real roots of polynomial systems becomes more challenging as we add more terms and variables. However, we can use convex geometry to find metric estimates of real roots in a fraction of the time! Here we will develop some of the tools necessary to build up to our main tool the Archimedean Tropical Variety, a piece-wise linear construction guaranteed to be close to our complex zero set, and a signed variant that has the same isotopy type as the positive zero set with high probability. We will show the results of applying these constructions to a family of 9×9 polynomial systems arising from chemical reaction networks.

1 Introduction

2 Preliminary Information

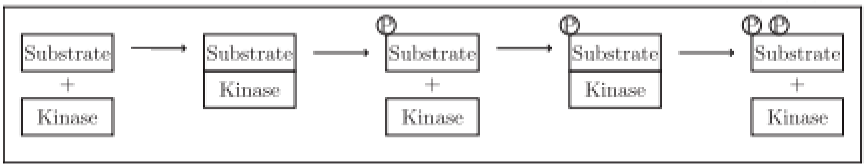
In order to get the 9×9 system of equations that we worked with, we must first discuss the biology behind this process.

2.1 Phosphorylation

Chemical reactions take place within every single cell of any living being; in other words, every plant and animal. The chemical reaction we chose to model is Phosphorylation.

Definition 2.1. Phosphorylation is the enzyme-mediated edition of a phosphate group to a protein substrate.

Within the cell, there are many different organelles. Within the Ribosome, protein are made from chains of Amino Acid. These proteins then make up the DNA and RNA of the cell which control most major functions. Phosphorylation then activates, or deactivates, certain functions of the cell by adding, or subtracting, phosphate groups to the Protein Substrate. The proteins particularly being used in our case are Kinase, Phosphatase, and Protein Substrate. These each have their own specific functions within Phosphorylation. Kinase has the function of starting Phosphorylation while Phosphatase has the function of starting Dephosphorylation. The Protein Substrate is unique because its function is to hold a protein (Kinase or Phosphatase at the respected time during the chemical reaction). The Substrate will absorb the Protein and add a phosphate group to itself. This will result in a modified Substrate that will soon attach to a different protein to repeat the process. An example of what that looks like:



To describe this picture, a Protein Substrate with no phosphate groups stands alone within a cell. At the same time, a Kinase protein stands alone within the same cell. Once triggered to preform a function, the Substrate and the Kinase attach and bind together. The Kinase is then absorbed and a phosphate group is added to the Substrate. The new (modified) Substrate group, now with one phosphate group, floats alone until a new Kinase is attached. This process repeats until the cell signals to being Dephosphorylation. In that case, the same process repeats, but phosphate groups are being removed.

2.2 Chemical Reaction Networks

Definition 2.2. A Chemical Reaction Network is a weighted directed graph with complexes are the vertices and and arrows labeled by reaction rate constants as the edges.

While we are using a Chemical Reaction Network to model Phosphorylation, this general graph could be used to model any chemical reaction. Below you will see what the graph looks like with our proteins being used in the reactions to model Phosphorylation:

$$S_0 + K \to S_0K \to S_1 + K \to S_1K \to S_2 + K \tag{2.1}$$

$$S_2 + P \to S_2P \to S_1 + P \to S_1P \to S_0 + P \tag{2.2}$$

Looking at this graph, there are three major components to take notes of. These components are reactions, complexes, and species

Definition 2.3. Reactions are represented by the arrows in the graph and labeled by the reaction rate constants. The reaction rate constants give the rate at which each reaction happens. We will label these in order by:

$$K_1, K_2, K_3, K_4, K_5, K_6, K_7, K_8 \tag{2.3}$$

Definition 2.4. Complexes are the vertices in the graph and the chemical compound formed by the union of species. These are defined by the following with X_i representing the concentration of each species:

$$S_0 + K,\ S_0K,\ S_1 + K,\ S_1K,\ S_2 + K,\ S_2 + P,\ S_2P,\ S_1 + P,\ S_1P,\ S_0 + P \tag{2.4}$$

Definition 2.5. Species are the molecules undergoing the chemical reactions that make up the components of the complexes. These are defined by:

$$X_1 := S_0,\ X_2 := S_1,\ X_3 := S_2,\ X_4 := S_0K,\ X_5 := S_1K,\ X_6 := S_1P,\ X_7 := S_2P,\ X_8 := K,\ X_9 := P \tag{2.5}$$

Now that we have clearly defined each piece of our graph, we can begin to use this to form our system of equations

2.3 Mass Action Kinetics

Mass Action Kinetics is the tool we use to find our equations.

Definition 2.6. Mass Action Kinetics tells us that the rate of an elementary reaction is proportional to the product of the concentrations of the species in the reactant. We use Mass Action Kinetics to obtain a system of Ordinary Differential Equations that model the Chemical Reaction Network.

Since we have nine species, we will be obtaining nine equations that represent rate of the reaction with that species. Now we must take a look at the graph and combine the rate of the reaction of an individual species each time that it goes through the reaction. These will be represented by $\dot{X}_i$ with i corresponding to its respected species.

$$S_0 + K \to S_0K \to S_1 + K \to S_1K \to S_2 + K \tag{2.6}$$

$$S_2 + P \to S_2P \to S_1 + P \to S_1P \to S_0 + P \tag{2.7}$$

To give a brief example of how this works, lets begin with finding $\dot{X}_1$. We are going to look at the graph and mathematically model each time the species represented by X_1 goes through a reaction. As stated earlier, X_1 represents the species S_0. In the graph, S_0 goes through a reaction twice. In the first case, the product of the concentrations is shown through the variables corresponding the species S_0, K, andS_0K. Since S_0 is being lost as its own species in the reaction, the reaction rate becomes negative. This results in the first case of X_1 to look like:

$$-K_1 \cdot X_1 \cdot X_8 \tag{2.8}$$

This process is trivial. After following the same process for the second case, the reaction rate equation for X_1 turns out to be:

$$\dot{X}_1 = -K_1 \cdot X_1 \cdot X_8 + K_8 \cdot X_6 \tag{2.9}$$

Once this process is repeated for all nine species, a system of equations will be obtained. This system looks like:

$$\begin{aligned}
\dot{X}_1 &= -K_1X_1X_8 + K_8X_6 \\
\dot{X}_2 &= K_2X_4 - K_3X_2X_8 + K_6X_7 - K_7X_2X_9 \\
\dot{X}_3 &= K_4X_5 - K_5X_3X_9 \\
\dot{X}_4 &= K_1X_1X_8 - K_2X_4 \\
\dot{X}_5 &= K_3X_2X_8 - K_4X_5 \\
\dot{X}_6 &= K_7X_2X_9 - K_8X_6 \\
\dot{X}_7 &= -K_5X_3X_9 - K_6X_7 \\
\dot{X}_8 &= -K_1X_1X_8 + K_2X_4 - K_3X_2X_8 + K_4X_5 \\
\dot{X}_9 &= -K_5X_3X_9 + K_6X_7 - K_7X_2X_9 + K_8X_6
\end{aligned}$$

2.4 Conservation Laws

The last major idea to look at before beginning to solve for the roots of our equation is to apply the conservation laws. The conservation laws are equations that represent the total amount of each species. These are important because the amount of species remains constant throughout the graph. The concentration are as follows, with S_{TOT} representing the total amount of Substrate, K_{TOT} representing the total amount of Kinase, and P_{TOT} representing the total amount of Phosphatase:

$$\begin{aligned}
&S_{TOT} = X_1 + X_2 + X_3 + X_4 + X_5 + X_6 + X_7 \\
&K_{TOT} = X_4 + X_5 + X_8 \\
&P_{TOT} = X_6 + X_7 + X_9
\end{aligned}$$

Now that we have these three new equations, we are forced to reduce out system to at least a 9×9. Notice that when we take the derivative of each side, we are left with a sum of our reaction rate equations:

$$\begin{aligned}
&0 = \dot{X}_1 + \dot{X}_2 + \dot{X}_3 + \dot{X}_4 + \dot{X}_5 + \dot{X}_6 + \dot{X}_7 \\
&0 = \dot{X}_4 + \dot{X}_5 + X_8 \\
&0 = \dot{X}_6 + \dot{X}_7 + \dot{X}_9
\end{aligned}$$

We can now algebraically simplify and reduce this. After reducing the system of equations as much as possible, we are left with the following 2×2 Quadratic Pentanomial System.

$$\begin{aligned}
c_1X_8^2 + c_2X_8X_9 + c_3X_8 + c_4X_9 + c_5 = 0 \\
c_6X_9^2 + c_7X_8X_9 + c_8X_8 + c_9X_9 + c_{10} = 0
\end{aligned}$$

We now have a system of equations that will be much easier to work with.

3 A-Discriminant Varieties

While we do ultimately want to solve for the roots of our 9×9 system of polynomial equations, first we must ask ourselves how many $\mathbb{R}$ roots are there, if any at all? In order to do this we will use something called the ***A-Discriminant Variety***.

Definition 3.1. n-dimensional Projective Space (over a field K), denoted by $\mathbb{P}^n_K$

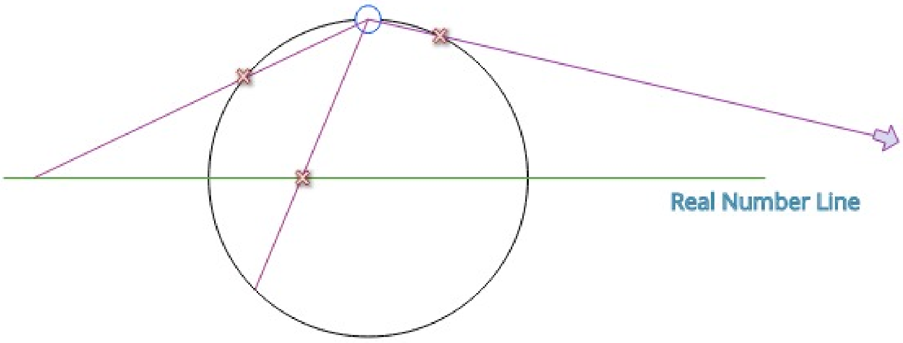
$$\{[x_0 : \cdots : x_n] \mid x_i \in K\} \; / \; \{[x_0 : \cdots : x_n] = [\lambda x_0 : \cdots : \lambda x_n] \; \forall \lambda \in K \setminus \{0\}\}$$

This is an equivalence class in which numbers that are scalar multiples of one another are placed in the same equivalence class. To further illustrate this concept, let's look at $\mathbb{P}^1_{\mathbb{R}}$

$\mathbb{P}^1_{\mathbb{R}}$ is referred to as *The Real Projective Line* and is defined as follows:

$$\mathbb{P}^1_{\mathbb{R}} := \{[x_0 : x_1] \mid x_0, x_1 \in \mathbb{R}\}/\{[x_0 : x_1] = [\lambda x_0 : \lambda x_1] \; \forall \lambda \in \mathbb{R} \setminus \{0\}\}$$

One way to visualize this space is the set of all lines that pass through the origin, with numbers that lie on the same line to be part of the same equivalence class. Another way to imagine this is through the following diagram:
Each point on the $\mathbb{R}$ number line is assigned to a point on the circle. The arrow pointing off the paper is meant to exemplify the fact that no matter how far off we go on the number line, the secant line that intersects the circle at any particular point will never be tangent to the circle. For this reason, we go ahead and assign $\{\infty\}$ to the North Pole!



We're particularly interested in $\mathbb{P}^n_{\mathbb{C}}$ because it becomes a much nicer way to parametrize polynomials since

$$c_1x^{a_1} + \cdots + c_dx^{a_d}$$

always has the same roots as

$$\lambda c_1x^{a_1} + \cdots + \lambda c_dx^{a_d} \quad (\forall \lambda \neq 0)$$

Definition 3.2. If $f(x) = c_1x^{a_1} + \cdots + c_dx^{a_d}$ and $A = \left[a_1, \ldots, a_d\right] \in \mathbb{Z}^{1\times d}$ (a_i are distinct) The ***A-Discriminant Variety***, denoted by ∇_A, is defined as:

$$\nabla_A := \{[c_1 : \cdots : c_d] \in \mathbb{P}^{d-1}_{\mathbb{C}} \mid f \text{ has a degenerate root in } \mathbb{C} \setminus \{0\}\}$$

where f has a degenerate root in $\mathbb{C} \setminus \{0\}\}$ means that at the root, the function has horizontal tangency.

Example 3.3. $f(x) = c_0 + c_1x + c_2x^2$

In this case, ∇_A is as follows:

$$\nabla_A = \{[c_0 : c_1 : c_2] \in \mathbb{P}^1_{\mathbb{C}} \mid \{ \begin{array}{l} c_0 + c_1x + c_2x^2 = 0 \\ c_1 + 2c_2x = 0 \end{array} \} \text{ has a root}\}$$

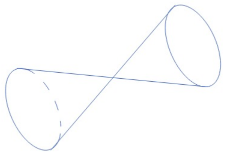
Turns out that if you play around with this system of equations, you end up with the following:

$$\nabla_A = \{[c_0 : c_1 : c_2] \in \mathbb{P}^1_{\mathbb{C}} \mid c_1^2 - 4c_0c_2 = 0\}$$

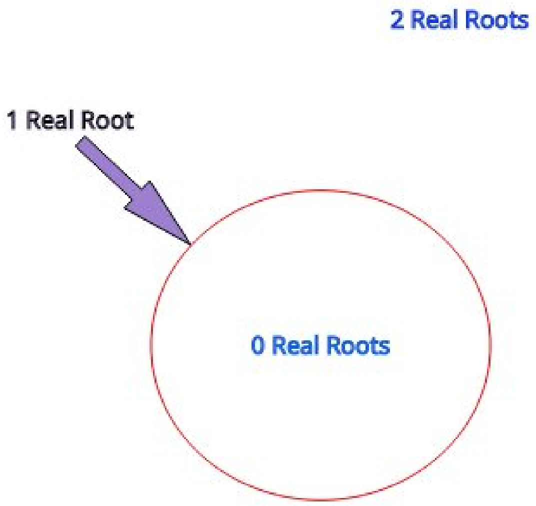
Recall from high school algebra, when $c_1^2 - 4c_0c_2 > 0$, $c_1^2 - 4c_0c_2 = 0$, and $c_1^2 - 4c_0c_2 < 0$, $f(x)$ has 2, 1, and 0 $\mathbb{R}$ roots, respectively. We call $c_1^2 - 4c_0c_2$ the ***A-discriminant polynomial***, which we denote by $\varDelta_A$. In general, ∇_A is the zero set of $\varDelta_A$!

Discriminants are important because they divide the coefficient space into regions with constant topology!

For this example, the zero set of $\varDelta_A$ is a double cone



If we look at the $\mathbb{R}$ part of $\nabla_A \cap \mathbb{P}^1_{\mathbb{R}}$ we get a simpler picture:



In $\mathbb{P}^1_{\mathbb{R}}$, our discriminant variety divides our coefficient space into 3 regions, with every point that lies on the circle representing the set of coefficients that gives us 1 $\mathbb{R}$ root, with every set of coefficients

The $\mathbb{R}$ part of $\nabla_A \cap \mathbb{P}^{d-1}_{\mathbb{R}}$ divides $\mathbb{P}^{d-1}_{\mathbb{R}}$ into regions with constant topology, just like the discriminant, but is much easier to work with than our typical A-discriminant polynomial, $\varDelta_A$!

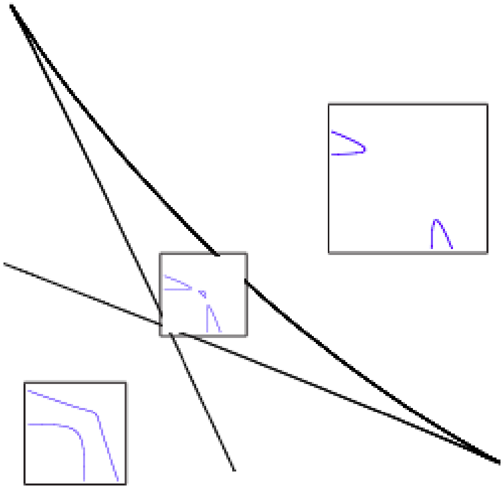
Example 3.4. $f(x, y) = c_0 + c_1x + c_2y + c_3x^4y + c_4xy^4$

Since we can always multiply by some scalar without changing our roots, we can always write a polynomial of this form into the following

$$f(x, y) = 1 - \alpha x - \beta y + xy(x^3 + y^3)$$

The A-Discriminant Variety for this example looks like this

The coefficient space is divided up in such a way in which the zero sets are either two connected pieces or two connected pieces and a point!



Extending our definition of ∇_A to more than one variable comes naturally, so finally, we can say:

If we have $f(x_1, ..., x_n)$ and $A = \left[a_1, \ldots, a_d\right] \in \mathbb{Z}^{n \times d}$

$$\nabla_A := \{[c_1 : \cdots : c_d] \in \mathbb{P}_{\mathbb{C}}^{d-1} \mid f = \tfrac{\partial f}{\partial x_1} = \tfrac{\partial f}{\partial x_2} = \cdots = \tfrac{\partial f}{\partial x_n} = 0\}$$

Now we can answer the question of which orthants our ∇_A touches. If we let $C :=$ cone over∇_A, then in the coefficient space, $\mathbb{R}^t \backslash C$ is a union of connected regions where the number of positive roots of our system is constant. Then we know that if ∇_A does not intersect with a particular orthant, then the number of positive roots is constant on the whole orthant. If ∇_A does intersect a particular orthant, then the number of positive roots can vary.

3.1 Horn-Kapranov Uniformization

Horn-Kapranov is going to assist us in seeing how the roots behave by simplifying how find the Discriminant.

Definition 3.5. Given $A = \{a_1, \cdots, a_m\} \subset \mathbb{Z}^N$ with ∇_A a hypersurface, the discriminant locus ∇_A is the closure of:

$$\{[u_1\lambda^{a_1} : \cdots : u_m\lambda^{a_m}] \mid u \in \mathbb{C}^m, Au = 0, \sum_{i=1}^{m} u_i = 0, \lambda \in (\mathbb{C}^*)^n\}$$

Thus, the null space of a $(n+1) \times m$ matrix, $\hat{A}$, provides a parametrization of ∇_A

Horn-Kapranov allows us to take seemingly complicated polynomials and rescale them by either a polynomial, x, or y. We can reduce the total number of parametrics and ultimately have less unknown coefficients we have to account for.

Theorem 3.6. The number of connected components of the real zero set of $f(x) = \sum_{i=1}^{T} c_i x^{a_i}$ (suitably compactified) is constant when $[c_1 : \cdots : c_T]$ ranges over a fixed connected component of $\mathbb{P}_{\mathbb{R}}^{T-1} \backslash \nabla_A$

This theorem is important because now we know that if we are evaulating the connected components and the sign of our discriminant polynomial is constant, then we can respectively read off the number of possible roots straight from the given ∇_A

3.2 Reduced A-Discriminant Contour

We are able to take a reduced version of our Discriminant Varieties and find a more efficient way to see the behavior of our coefficients.

Definition 3.7. The Reduced A-Discriminant Contour can be defined as
$C := \{(log|[\lambda_1 : \lambda_2]|B^T)B \subset \mathbb{R}^2\}$

Notice this is not the same as the previously defined A-Discriminant Variety and this definition loses information because of the absolute value signs. We are losing information about the signs of the coefficients which will play a crucial role in how our roots behave.

3.3 Signed Reduced A-Discriminant Contour

We can fix the signs of $\sigma = (\text{sign}(c_1), \cdots, \text{sign}(c_4))$ to get C_σ = a piece of C corresponding to $\nabla_A \cap \mathbb{P}^{t-1}_{\mathbb{R}}$ with $[c_1 : \cdot : c_4]$ having sign $\neq \sigma$.

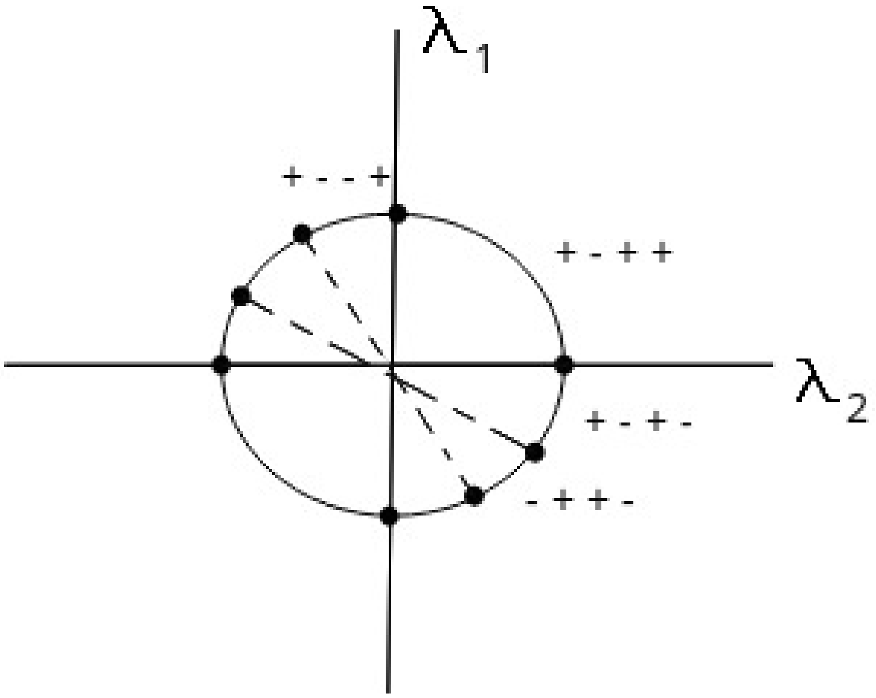
Using Horn-Kapranov, we can find the signs of our λ_1 and λ_2 by finding the right null space of the given A. We can then find what the signs of the coefficients will be in the projective space according to which regions.

Example 3.8. $A = [0, 1, 2, 3] \Rightarrow f(x) = c_1 + c_2x + c_3x^2 + c_4x^3$

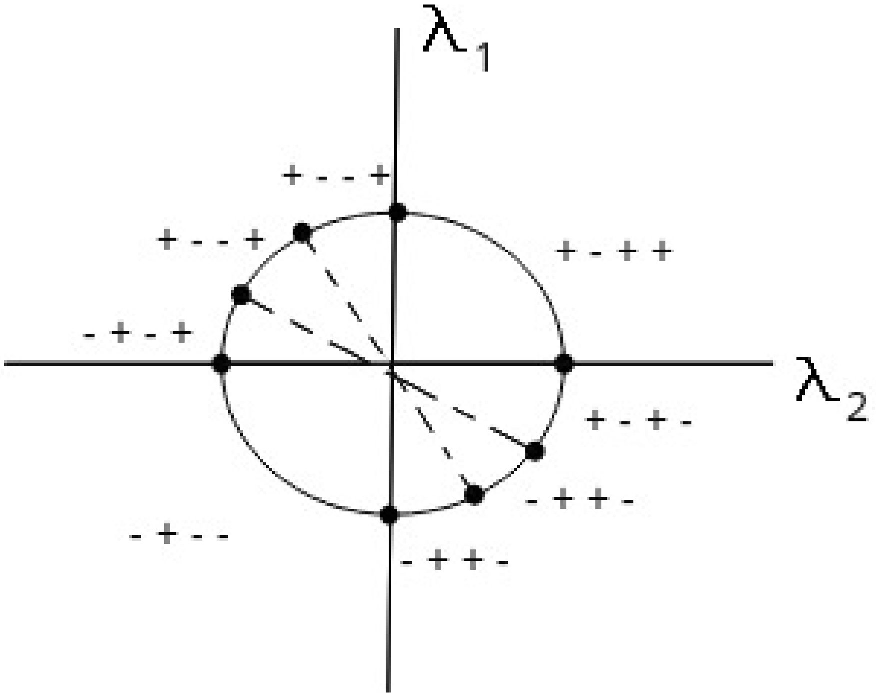
$$\hat{A} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 \end{bmatrix} \qquad \Rightarrow \qquad B = \begin{bmatrix} 1 & 2 \\ -2 & -3 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\begin{aligned} [c_1 : c_2 : c_3 : c_4] \in \mathbb{P}^3_{\mathbb{R}} \Rightarrow\ & \text{sign}(\lambda_1 + 2\lambda_2) = \sigma_1 \text{ or } -\sigma_1 \\ & \text{sign}(-2\lambda_1 - 3\lambda_2) = \sigma_2 \text{ or } -\sigma_2 \\ & \text{sign}(\lambda_1) = \sigma_3 \text{ or } -\sigma_3 \\ & \text{sign}(\lambda_2) = \sigma_4 \text{ or } -\sigma_4 \end{aligned}$$

Looking at this example, we have our given A matrix, we can solve for a corresponding B matrix, and then we are able to use Horn-Kapranov to solve for the signs of the lambdas. We can then set each sign equation equal to zero which will result in four lines on our λ_1 and λ_2 coordinate plane in the projective space.



As we expected, we can see that the four lines we obtained divide our projective space into four regions where we have specific sign possibilities.



Notice that the opposite region gives you the exact opposite signs if we are to fill projective space all the way around the circle. Also, notice that not all sign possibilities are accounted for. The sign possibilities not accounted for give us sign possibilities that will vary on the amount of roots obtained.

4 Linear Programming

Now we are able to see how the roots behave depending on the coefficients using our A-Discriminant Varieties. We have also seen that we are able to get a more accurate version of this if we pay attention to the signs of our coefficients. To extend more on this idea, we know that the different combination of signs will tell us where the orthants of constant $\mathbb{R}$ roots are. Our 2×2 system has 1024 sign combinations, so we can use Linear Programming to quickly run through all 1024 sign vectors to see which ones will give us a region with constant roots.

By Horn-Kapranov, the $\mathbb{P}^{t-1}_{\mathbb{R}} \cap \nabla_A$ is given by: $((\text{vector of linear forms})^{\text{rational powers}} \cdots)$

So the possible signs of the coordinates of ∇_A are described by: $(sign(\beta_1 {\cdot} \lambda), \cdots, sign(\beta_t {\cdot} \lambda))$ where the null space B looks like:

$$B = \begin{bmatrix} \beta_1 \\ \cdot \\ \cdot \\ \cdot \\ \beta_T \end{bmatrix}$$

In other words,
A choice of sign $\sigma := (\sigma_1, \cdots, \sigma_t) \in \{\pm 1\}^t$ occurs $\Leftrightarrow \exists \lambda = (\lambda_1, \cdots, \lambda_{t-n-1}) \in \mathbb{R}^{t-n-1}$ with $sign(\lambda \cdot \beta_1) = \sigma_1 \cdots sign(\lambda \cdot \beta_t) = \sigma_t$.

So, for LP feasibility, we want to see if $\lambda \cdot \beta_1 + \cdots + \lambda_{t-n-1} \cdot \beta_{t-n-1} \gtrless 0$.

4.1 Linear Programming - Step One

The first thing we must do is find the Cayley Embedding of our 2 × 2 system.

$$c_1X_8^2 + c_2X_8X_9 + c_3X_8 + c_4X_9 + c_5 = 0$$
$$c_6X_9^2 + c_7X_8X_9 + c_8X_8 + c_9X_9 + c_{10} = 0$$

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

The above A matrix will represent our Cayley Embedding. The first row in the matrix represents the ones we see after parameterizing with Horn-Kapranov. The second row represents the exponents on the X's in both polynomials. The third row represents the exponents on the Y's in both polynomials. Finally, the last row is a block of zeros and ones that are necessary by the Cayley Trick for this embedding to function properly.

4.2 Linear Programming - Step Two

We must now find the corresponding B matrix which is just the right null space of the previously obtained A matrix. This will result in the following matrix:

$$B = \begin{bmatrix} 1 & 1 & -1 & -1 & 0 & 0 \\ -1 & 0 & 1 & 2 & 1 & 2 \\ -1 & -2 & 0 & -1 & -1 & -2 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & -1 & -1 & -1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

4.3 Linear Programming - Step Three

The final step of Linear Programming is to solve the 1024 Linear Programming feasibility problems. Standard Form of Linear Programming we will be using a program that will maximize $c \cdot x$ such that $Ax = b, x \geq 0$. Therefore, we will be placing our A and B matrix in to this program and ultimately solving the following question:

Does there exist λ such that

$$\lambda_1 \cdot \beta_1 + \cdots + \lambda_{t-n-1} \cdot \beta_{1,t-n-1} \overset{>}{\underset{<}{}} 0$$

$$\vdots$$

$$\lambda_1 \cdot \beta_t + \cdots + \lambda_{t-n-1} \cdot \beta_{t,t-n-1} \overset{>}{\underset{<}{}} 0$$

$\lambda\beta^t \gtrless 0$?

4.4 Results

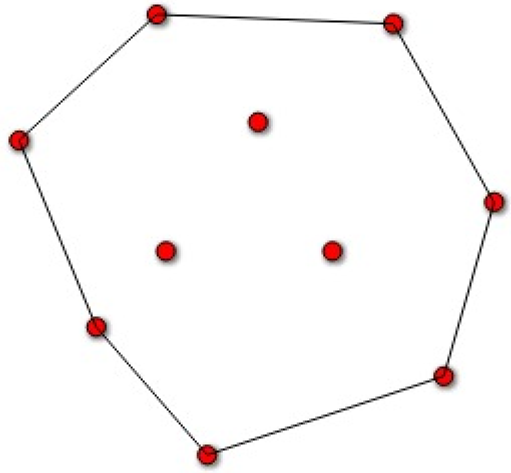
When we first ran this program 1024 times, we found that there were 674 sign combinations where the inequalities are feasible. When then realized that we can force c_5 and c_{10} to be positive because we can always multiply these two coefficients by a number to force them to be positive without altering the geometry of the roots in the projective space. Once we did this, we lowered the amount of sign possibilities to 256 and found that there are 151 sign combinations where the inequalities are feasible. A region lying within the 151 sign combinations where the inequalities are feasible will result in an area with varying root amounts. The other 105 sign combinations tell us where the regions of constant roots are. We could go even further to find exactly which signs give us which number of roots. We will see error drawing these roots out when our coefficients end up having a sign combination where the inequalities are feasible.

5 Convex Geometry

Solving for the $\mathbb{R}$ roots of polynomial systems becomes more challenging as we add more terms and variables. However, we can use convex geometry to find metric estimates of real roots in a fraction of the time! Here we will develop some of the tools necessary to build up to our ultimate goal – the Archimedean Tropical Variety, a construction which splits up the domain of a function into more specific regions of interest.

Definition 5.1. A ***convex set*** is a set of points such that, given any two points P, Q, then the line segment PQ is also in the set

Definition 5.2. For any $S \subset \mathbb{R}^n$, the ***convex hull*** of S, denoted conv$\{S\}$ is the *smallest* convex set containing S.
For a set of points sitting in $\mathbb{R}^2$, this looks like the following:



We will use the convex hull function in our first construction, the Newton Polytope.

Definition 5.3. If $f(x) = \sum_{i=1}^{t} c_i x^{a_i}$, where $x^{a_i} = x_1^{a_{1,i}} x_2^{a_{2,i}} \cdots x_n^{a_{n,i}}$, the ***Newton Polytope*** of f, denoted by Newt(f), is defined as follows:

$$\text{Newt}(f) := \text{conv}\{a_i \mid c_i \neq 0\}$$

The volume of the Newton polytope can be used to compute the degree of the corresponding hypersurface, and via mixed volumes, the number of roots of systems of equations!

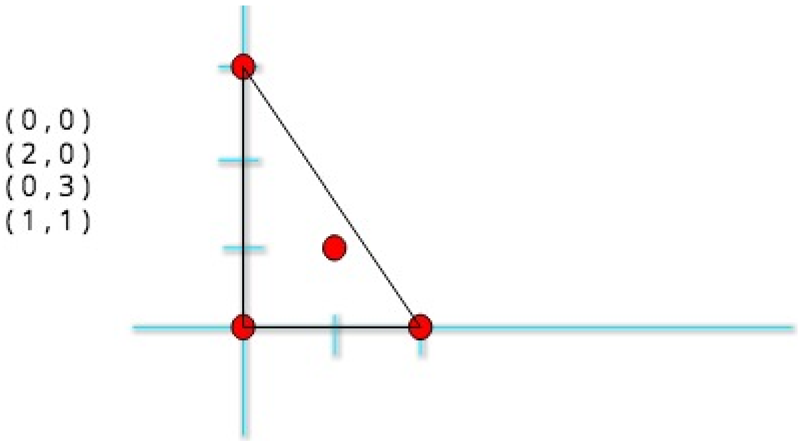
As with most things, this construction and the proceeding ones are best illustrated with an example.

Example 5.4. $f(x, y) = 1 + x^2 + y^3 - 100xy$

If we write this out in the following way, although a bit redundant, we can easily read off the points that we will then take the convex hull of

$f(x, y) = 1 * x^0 * y^0 + 1 * x^2 * y^0 + 1 * x^0 * y^3 - 100 * x^1 * y^1$

$\Rightarrow \text{Newt}(f) = \text{conv}\{(0, 0), (2, 0), (0, 3), (1, 1)\}$



From here we can construct the Archimedean Newton Polytope, which builds on our already established Newton Polytope by taking into account the coefficient of each term, and assigning to each point what can be though of as a weight in the negative z direction.

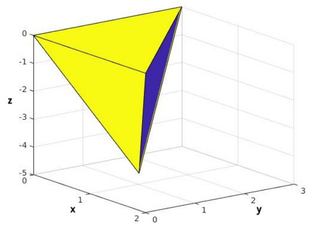
Definition 5.5. If $f(x) = \sum_{i=1}^{t} c_i x^{a_i}$, where $x^{a_i} = x_1^{a_{1,i}} x_2^{a_{2,i}} \cdots x_n^{a_{n,i}}$, the ***Archimedean Newton Polytope*** of f, denoted by ArchNewt(f), is defined as follows:

$$\text{ArchNewt}(f) := \text{conv}\{(a_i, -\text{Log}|c_i|) \mid i \in \{1, \ldots, t\}, c_i \neq 0\}$$

Example 5.6. $f(x, y) = 1 + x^2 + y^3 - 100xy = 1 * x^0 * y^0 + 1 * x^2 * y^0 + 1 * x^0 * y^3 - 100 * x^1 * y^1$

Writing it out in this form again makes it rather easy to see that

$$\begin{aligned}\text{ArchNewt}(f) &= \text{conv}\{(0, 0, -\text{Log}(1)), (2, 0, -\text{Log}(1)), (0, 3, -\text{Log}(1)), (1, 1, -\text{Log}(100))\}\\ &= \text{conv}\{(0, 0, 0), (2, 0, 0), (0, 3, 0), (1, 1, -\text{Log}(100))\}\end{aligned}$$



Using the Newton Polytope and the Archimedean Newton Polytope, we can directly construct what is called the Archimedean Tropical Variety. However, first we will begin with a definition of the Archimedean Tropical Variety that is independent of the previous two constructions, in order to elaborate more on the information this tropical variety can tell us about a function.

Definition 5.7. If $f(x) = \sum_{i=1}^{t} c_i x^{a_i}$, where $x^{a_i} = x_1^{a_{1,i}} x_2^{a_{2,i}} \cdots x_n^{a_{n,i}}$, the ***Archimedean Tropical Variety*** of f, denoted by ArchTrop(f), is defined as follows:

$$\text{ArchTrop}(f) := \{w \in \mathbb{R}^n \mid \underset{i}{max} |c_i e^{w \cdot a_i}| \text{ at } \geq 2 \text{ distinct } i\}$$

where $w = (w_1, \ldots, w_n)$

Let's proceed naively using our familiar example

Example 5.8. $f(x, y) = 1 * x^0 * y^0 + 1 * x^2 * y^0 + 1 * x^0 * y^3 - 100 * x^1 * y^1$

$|c_1 e^{w \cdot a_1}| = |1 * e^{(w_1, w_2) \cdot (0,0)}| = 1$

$|c_2 e^{w \cdot a_2}| = |1 * e^{(w_1, w_2) \cdot (2,0)}| = e^{2w_1}$

$|c_3 e^{w \cdot a_3}| = |1 * e^{(w_1, w_2) \cdot (0,3)}| = e^{3w_2}$

$|c_4 e^{w \cdot a_4}| = |-100 * e^{(w_1, w_2) \cdot (1,1)}| = 100 e^{w_1 + w_2}$

In order to draw ArchTrop(f), let's look at when each pair of terms is maximal and equal in norm!

For the 1st and 2nd terms:

$1 = e^{2w_1} \Rightarrow w_1 = 0$

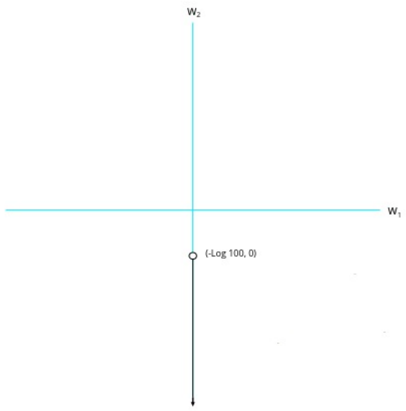
$1 > e^{3w_2} \Rightarrow 0 > w_2$

$1 > 100 e^{w_1 + w_2} \Rightarrow -\text{Log}100 > w_2$

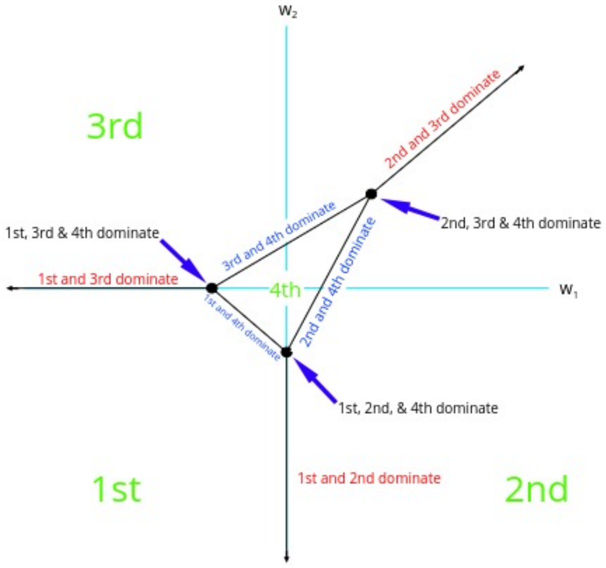
In summary, the 1st and 2nd terms *dominate* when:

$w_1 = 0$
$w_2 < -\text{Log}100$

For ArchTrop(f), this gives us the following ray:



Repeating the process with the 1st and 3rd terms, 2nd and 3rd terms, etc, yields us the following graph for ArchTrop(f):



Notice, ArchTrop(f) has divided the domain of our function into pieces where different terms dominate over the others. For example, on the ray that points to the Northeast, the 2nd and 3rd terms dominate, the point where this ray originates from, the 2nd,3rd, and 4th terms dominate, and the area captured within the triangle represents where the 4th dominates all on its own.

Furthermore, ArchTrop(f) gives us metric information about the roots and areas where we can find constant isotopy types.Much like how the quadratic discriminant $c_1^2 - 4c_0c_2$ gives us information about the number of roots, ArchTrop(f) can do this for more general curves!

Example 5.9. $f(x, y) = 1 + x^2 + y^3 - cxy$ $(c > 0)$ (generalized form of our familiar example)

Turns out the zero set of $f(x, y)$ is either $\varnothing$, a point, or an oval, and this occurs when $c < \frac{6}{2^{\frac{1}{3}}3^{\frac{1}{2}}}$, $c = \frac{6}{2^{\frac{1}{3}}3^{\frac{1}{2}}}$, $c > \frac{6}{2^{\frac{1}{3}}3^{\frac{1}{2}}}$, respectively.

Notice, in the case we've been working with $c = 100 > \frac{6}{2^{\frac{1}{3}}3^{\frac{1}{2}}}$, which is reflected in our picture of ArchTrop(f).

Theorem 5.10. Let $Z_{\mathbb{C}}(f) :=$ the Complex zero set of f
For any pentanomial f in $\mathbb{C}[x_1, \ldots, x_n]$, any point of $\mathrm{Log}|Z_{\mathbb{C}}(f)|$ is within distance $\log(4)$ of some point of ArchTrop(f).

The proof of this theorem is a direct consequence of a more general theorem:

Theorem 5.11. *For any $f \in \mathbb{C}[x_1^{\pm 1}, \cdots, x_n^{\pm 1} =]$ with exactly $t \geq 2$ monomial terms and* Newt(f) *of dimension k we have:*

(1) *(a)* $\mathrm{Amoeba}(f) \subseteq \overline{\mathrm{ArchTrop}(f)}_{\log(t-1)}$ *and, for $k = 1$,* $\mathrm{Amoeba}(f) \subsetneq \mathrm{ArchTrop}(f)_{\log 3}$.
(b) $\mathrm{ArchTrop}(f) \subseteq \mathrm{Amoeba}(f)_{\varepsilon k,t}$ *where* $\varepsilon_{1,t} := (\log 9)t - \log\frac{81}{2} < 2.2t - 3.7$,
$\varepsilon_{2,t} := \sqrt{2}(t-2)((\log 9)t - \log\frac{81}{2}) < (t-2)(3.11t - 5.23)$, *and*
$\varepsilon_{k,t} := \sqrt{k}[\frac{1}{4}t(t-1)]((\log 9)t - \log\frac{81}{2})$ *for* $k \geq 3$.

In particular, $\varepsilon_{k,t} < \frac{3}{5}t^{3/2}(t-1)^2$ *for all* $k \geq 1$ *and* $t \geq 2$.

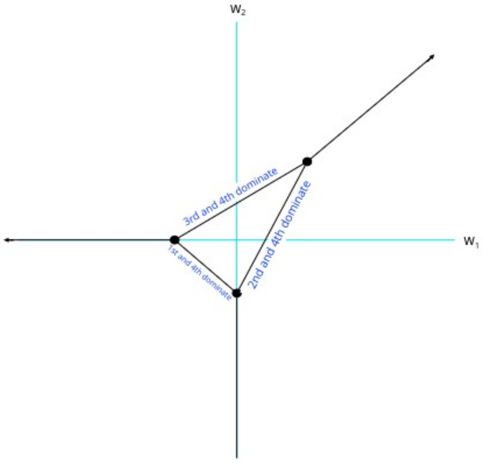
(2) *Let* $\phi(x) := 1 + x_1 + \cdots + x_{t-1}$ *and* $\psi(x) := (x_1+1)^{t-k} + x_2 + \cdots x_k$*. Then*
(a) Amoeba(ϕ) *contains a point at distance* $\log(t-1)$ *from* ArchTrop(ϕ) *and*
(b) ArchTrop(ψ) *contains points approaching distance* $\log(t-k)$ *from* Amoeba(ψ).

Theorem 3.4, Assertion 1a, from
"Metric Estimates and Membership Complexity for Archimedean Amoebae and Tropical Hypersurfaces," (by Martin Avendano, Roman Kogan, Mounir Nisse, and J. Maurice Rojas), presented at MEGA 2013, Journal of Complexity, Vol. 46, June 2018, pp. 45-65. QED

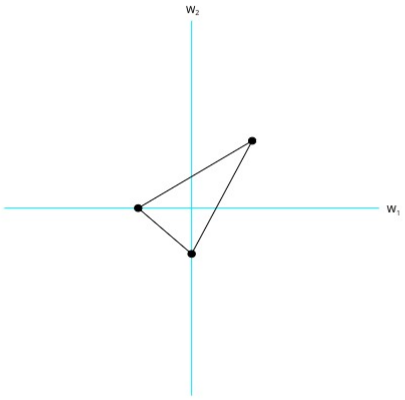
While this is useful for more general situations, in the case of our 9 × 9 system, since there is a biological process underlying our equations, it doesn't make sense for us to look at negative roots, let alone complex ones! This motivates us to look towards another construction, but fortunately, it is formed similarly as ArchTrop(f).

Definition 5.12. ArchTrop$_+$ $(f) := \{w \in \mathbb{R}^n \mid \underset{i}{max}|c_i e^{w\cdot a_i}| \text{ w/ } c_j c_{j'} < 0\}$
Notice ArchTrop$_+(f) \subset$ ArchTrop(f)

Example 5.13. $f(x,y) = 1 + x^2 + y^3 - 100xy$
The coefficient pairs that multiply to a negative product are the 1st and 4th, the 2nd and 4th, and the 3rd and 4th! Recall that these pairs of terms dominate here:



Which leaves us with the following picture for ArchTrop$_+(f)$



$\mathrm{ArchTrop}_+(f)$ gives us a piecewise linear function that resembles the set of positive roots! More formally,

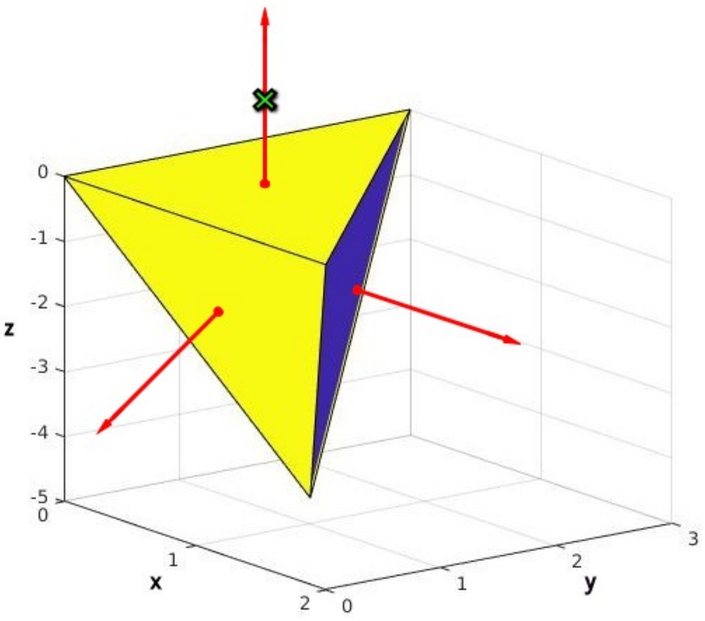
Theorem 5.14. Let $Z_+(f) :=$ the positive zero set of f
For any pentanomial f in $\mathbb{C}[x_1, \ldots, x_n]$, any point of $\mathrm{Log}|Z_{\mathbb{C}}(f)|$ is within distance $\log(4)$ of some point of $\mathrm{ArchTrop}_+(f)$.

Now let's go over the method of constructing ArchTrop(f) that requires both Newt(f) and ArchNewt(f).

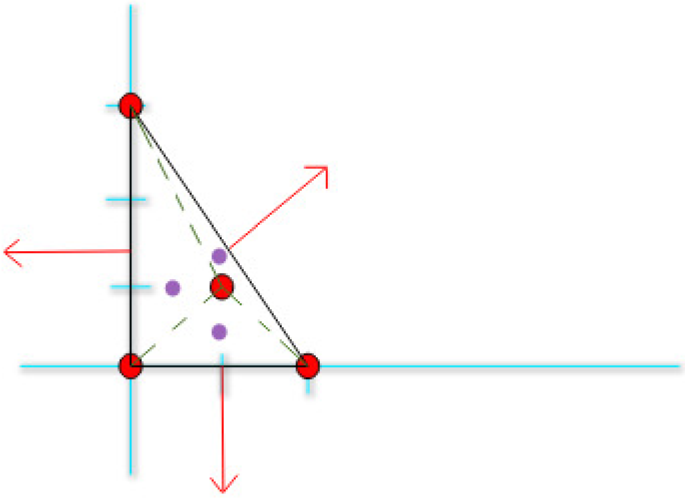
From ArchNewt(f) we will need the outer normals of the lower hull of our convex hull. From there, we will project the lower faces of ArchNewt(f) unto the plane where Newt(f) lies, forming a triangulation of our Newton Polytope.

From Newt(f), we will need the outer normals of the edges of our convex hull

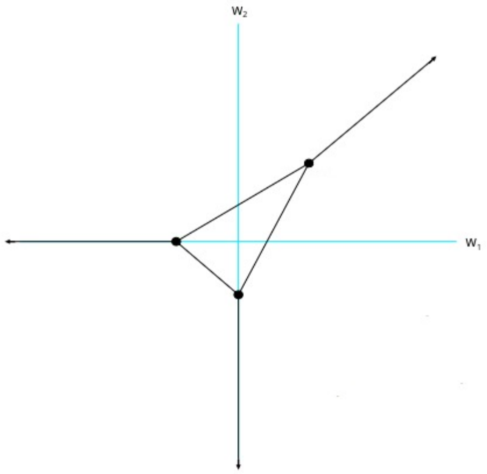
Referring back to our familiar example, that looks like the following:



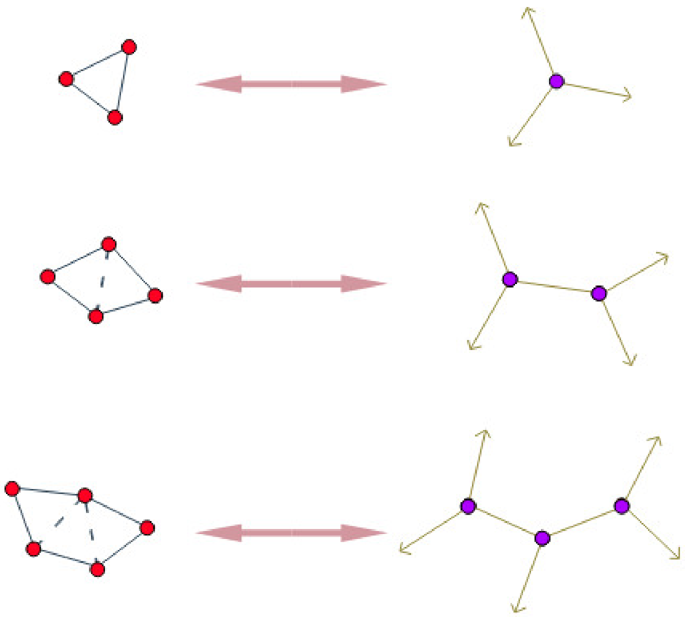
Once we have these outer normals, we need to normalize them into the form $(w, -1)$, where we then take w to be the vertices of our ArchNewt(f). Now we will project the edges of our lower hull onto the plane in which Newt(f) lies.



Notice that this induces a triangulation of our Newton Polytope. The dots in these triangles roughly correspond to the respective outer normals of the lower edges of ArchNewt(f). Finally, we take the outer normals of the edges of our Newton Polytope, and now ArchTrop(f) looks like this:

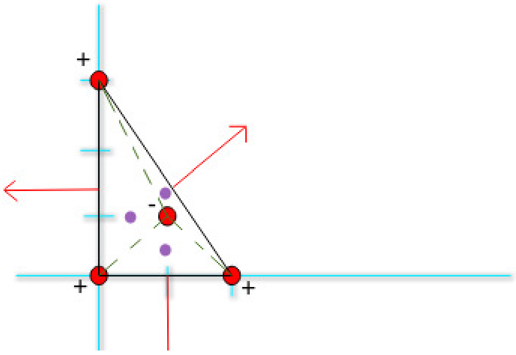


Just as we had before! Now, to elaborate, the vertices of ArchTrop(f) are *dual* to the triangulation of Newt(f) induced by the lower faces of ArchNewt(f) as are the rays are *dual* to the edges of Newt(f)



By dual, we mean there exists a bijective order-reversing mapping from our Newton Polytope on the left to our Archimedean Tropical Variety on the right. Each triangle in our triangulated Newton Polytope corresponds to a point of ArchTrop(f), and each outer edge of our triangle corresponds to a ray of ArchTrop(f). In the cases where the Newton Polytope is triangulated with adjacent triangles, there is a line segment in ArchTrop(f) that connects of the points that were respectively assigned to.

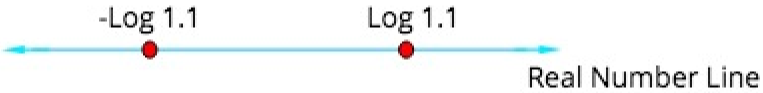
In order to construct ArchTrop$_+(f)$ after constructing ArchTrop(f) like this, we look again for alternating signs. We do this by assigning to each of our exponent vectors a corresponding sign. Then, we look for pieces which containing the same signs, and in a sense "turn off" that piece of ArchTrop$_+(f)$. Our Newton Polytope now looks like this:



Notice that the ray pointing to the Northeast lies on the exterior edge of our Newton Polytope connected by two positive points, so the corresponding ray of ArchTrop(f) will not appear in ArchTrop$_+(f)$!

Before we go on, a small discrepancy must be pointed out with ArchTrop$_+(f)$, which is best illustrated with an example.

Example 5.15. $f(x) = 1 - 1.1x + x^2$
ArchTrop$_+(f)$ looks like



ArchTrop$_+(f)$ is telling us that two points make up the positive zero set of our function, but if you look at the discriminant $\Rightarrow 1.1^2 - 4 < 0 \Rightarrow f$ has two non-$\mathbb{R}$ roots! So this is a case in which ArchTrop$_+(f)$ is not giving us accurate information. On the other hand, if you look at $\{c \in \mathbb{R}_+ \mid \text{connected zero set of } (1 - cx + x^2) \neq \text{ArchTrop}_+(f)\} = (0, 2)$.

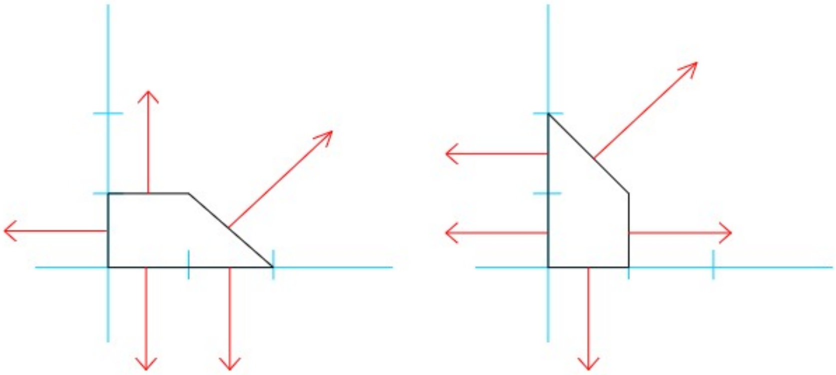
This isn't too bad when compared to the entirety of $\mathbb{R}$, so it does not negate our study of ArchTrop$_+(f)$

6 Our Research

In this section, we will be applying these constructions to our reduced 2×2 quadratic pentanomial system, while motivating how we wrote our code throughout.

$$f_1(x_8, x_9) = c_1x_8^2 + c_2x_8x_9 + c_3x_8 + c_4x_9 + c_5$$
$$f_2(x_8, x_9) = c_6x_9^2 + c_7x_8x_9 + c_8x_8 + c_9x_9 + c_{10}$$

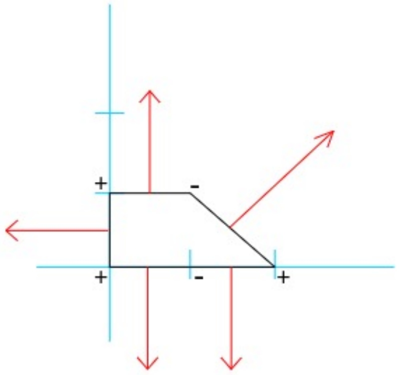
Constructing their respective Newton Polytopes results in the following convex hulls, along with their corresponding outer normals, yields us:



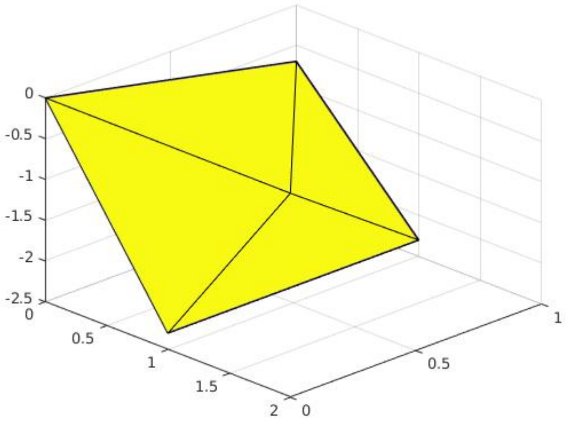
For the moment, we will focus on f_1, since all of the constructions are similar for both functions.

Example 6.1. Suppose $c_1 = 1, c_2 = -10, c_3 = -10, c_4 = 2, c_5 = 1$

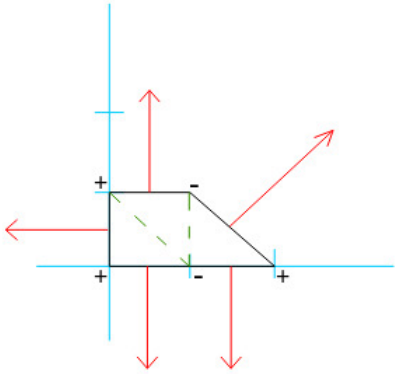
This results in the following combination of signs:



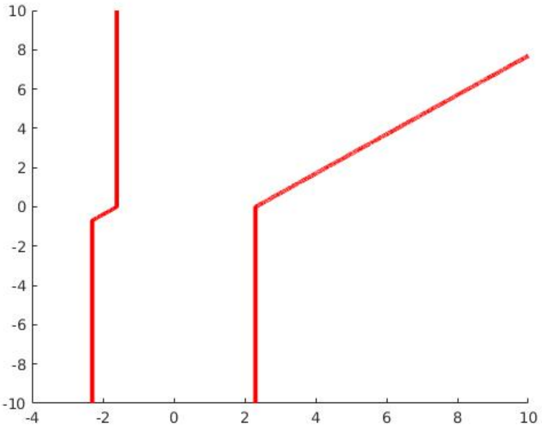
With *these* coefficients, we end up with the following for $\mathrm{ArchNewt}(f_1)$:



From there we obtain the following triangulation of $\mathrm{Newt}(f_1)$:



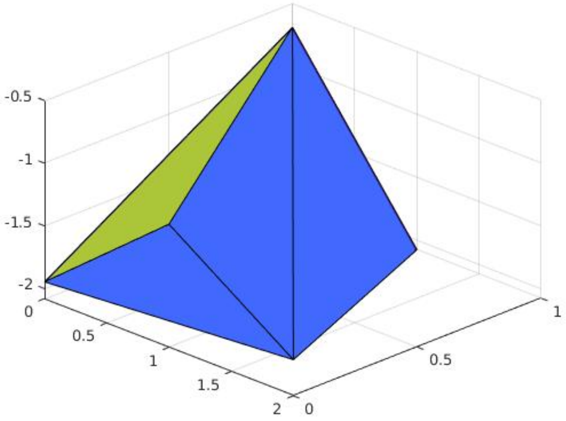
Hence, the following for $\mathrm{ArchTrop}_+(f_1)$:



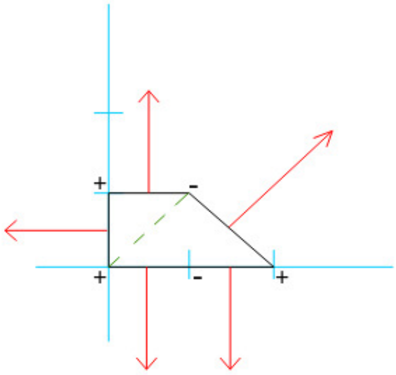
We used Matlab to produce the above image. At first our code was hyper-specific to the aforementioned triangulation. We started to use Matlab's random integer generator for the interval $[-10, 10]\backslash\{0\}$ (using a uniform distribution), while keeping the signs the same. However, when we found the following coefficients,

$$c_1 = 6, c_2 = -8, c_3 = -3, c_4 = 2, c_5 = 7$$

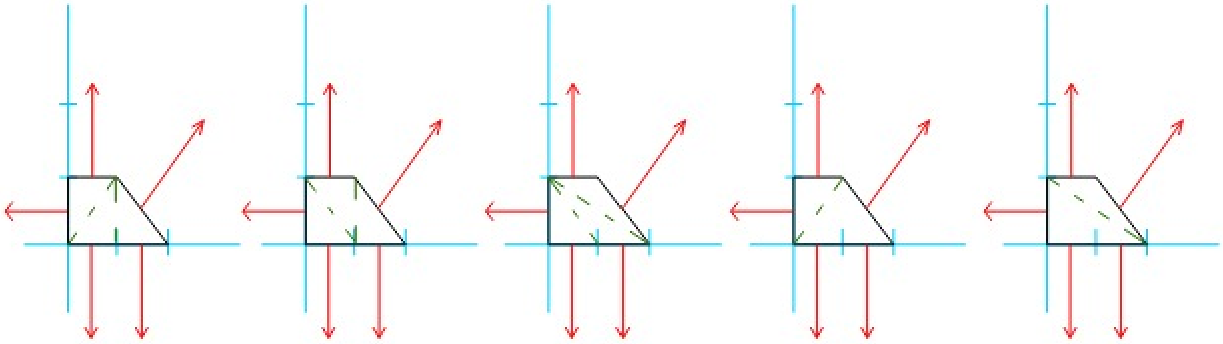
$\mathrm{ArchTrop}_+(f_1)$ was not plotting as we were expecting it to. This is because we didn't realize that we obtain the following picture for $\mathrm{ArchNewt}(f_1)$:



and subsequently the following picture for $\mathrm{Newt}(f_1)$:



This triangulation of our Newt(f_1) consists of two triangles, hence our ArchTrop$_+(f_1)$ would only contain two vertices! After some careful analysis, we soon came to find out, that actually there were a total of 5 possible triangulations for f_1.

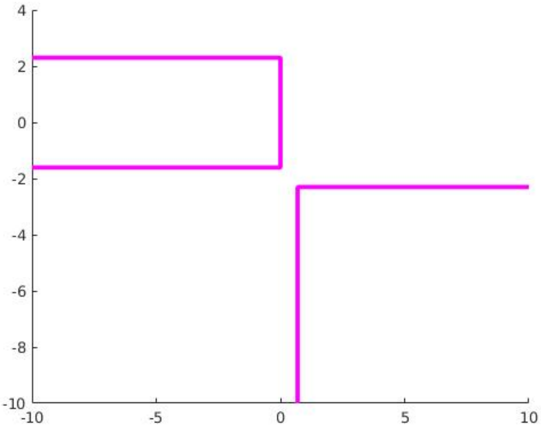


Again, since Newt(f_2) is merely a rotation of Newt(f_1), it too has only 5 possible triangulation for f_2! Once we moved passed that, we able to begin to successfully draw pictures for ArchTrop$_+(f_1)$ and ArchTrop$_+(f_2)$:

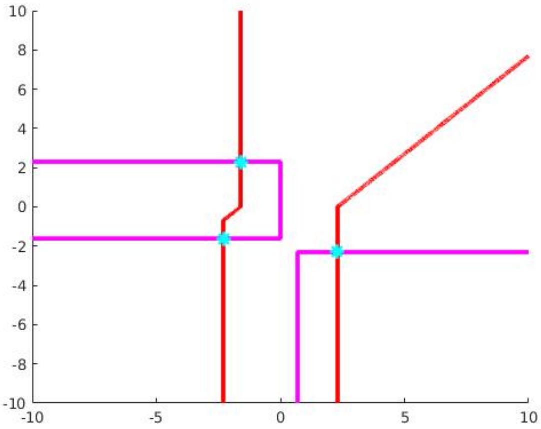
Example 6.2. Suppose for f_2 we had the following coefficients:

$$c_6 = 1, c_7 = -10, c_8 = -10, c_9 = 2, c_1 0 = 1$$

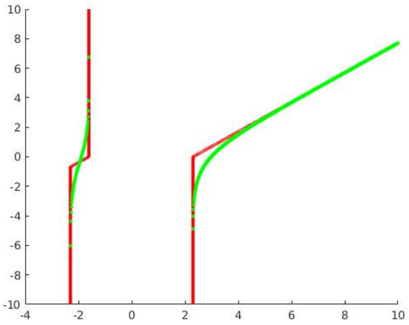
Following the same steps yields us the following for ArchTrop$_+(f_2)$:

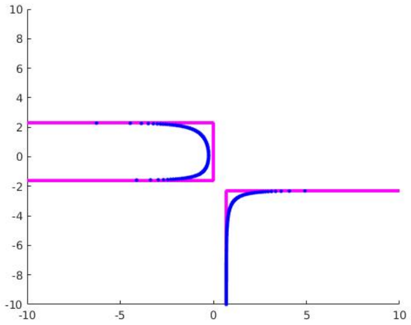


Something interesting happens when we plot both on the same set of axes!

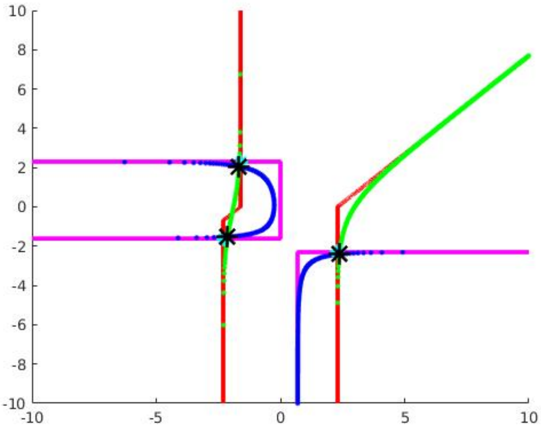


Notice that $\mathrm{ArchTrop}_+(f_1)$ and $\mathrm{ArchTrop}_+(f_2)$ intersect 3 times (represented by the cyan points)! This will come in handy in just a moment. Recalling Theorem 5.10, notice what happens when we plot $\mathrm{Log}|Z_+(f_1)|$ and $\mathrm{Log}|Z_+(f_2)|$ on the same set of axes as their corresponding $\mathrm{ArchTrop}_+(f_i)$.





Notice what happens when we plot all of these together on the same set of axes:



The positive zero sets have the same number of intersections (represented by the black points) as $\mathrm{ArchTrop}_+(f_1) \cap \mathrm{ArchTrop}_+(f_2)$! It is important to note that they do not line up perfectly, but we'll get to that shortly.In the meantime, we have a theorem to go along with this phenomenon:

Theorem 6.3. *If F is a random real 2×2 quadratic pentanomial system with supports having Cayley embedding*

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

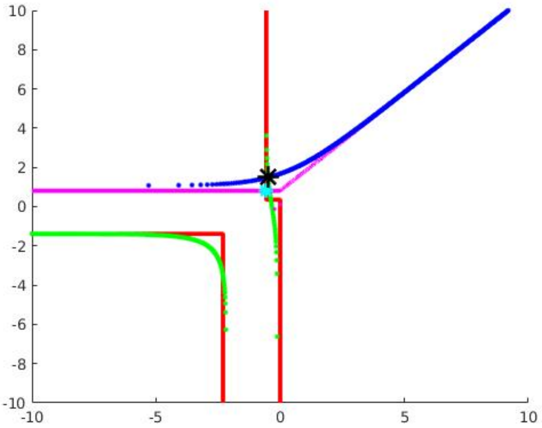
such that the coefficient vector $(c_1, \dots, c_{10})$ has each c_i with mean 0, then with probability at least 41%, F has the same number of positive roots as the cardinality of $\mathrm{ArchTrop}(f_1) \cap \mathrm{ArchTrop}(f_2)$.

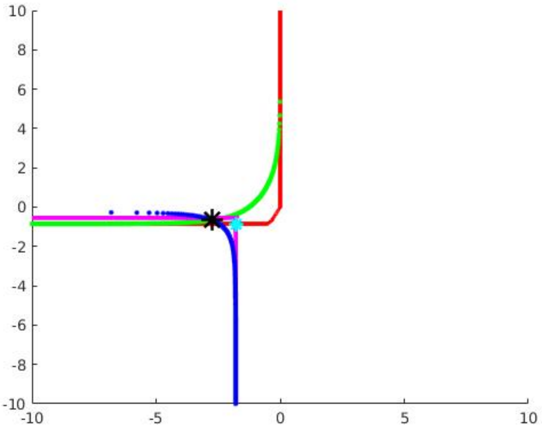
The proof follows from our analysis of sign conditions:

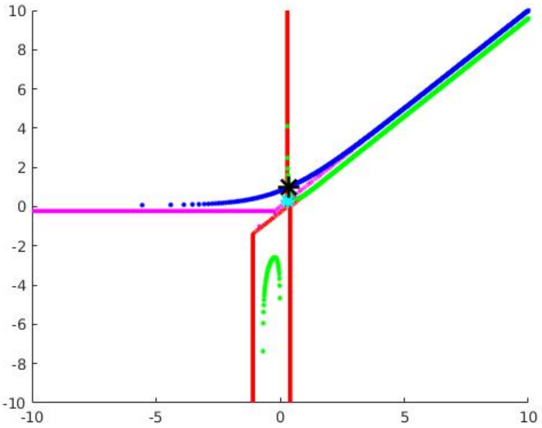
Proof. Assuming the constant terms of both f_1 and f_2 are normalized to be 1 (dividing each equation $f_i = 0$ by a suitable constant), there are a total of $2^4 * 2^4 = 256$ possibilities for the remaining signs. Via linear programming, we found that $\nabla_{\mathcal{A}}$ intersect exactly 151 of these orthants. Assuming each coefficient is a real random variable with mean 0, this means that positive and negative sign are equally likely, and occur respectively with probability 50%. So by our later theorem on signed chambers, the cardinality of the $\mathrm{ArchTrop}(f_i)$ intersections and the intersections of the $\mathrm{Log}\,|Z_+(f_i)|$ are the same on exactly $256 - 151 = 105$ out of 256 of the possible orthants. So, with probability at least $105/256 > 0.41$, these intersection cardinalities are the same and we are done. QED □

While 41% may not seem like the most appealing number, we will see soon that it turns out that this number appears to be closer to 75%!

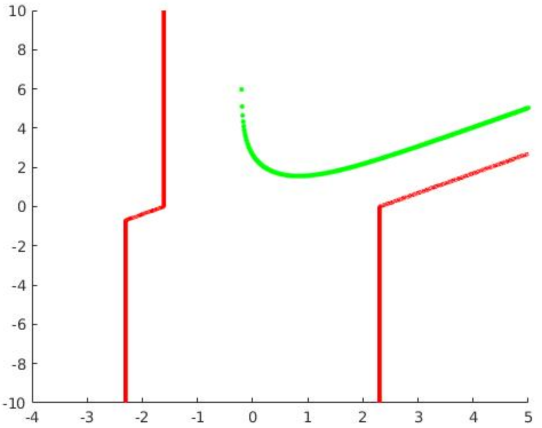
In order to test this theorem, we used Matlab's random number generator, which relies on the Uniform distribution. We specified our numbers to the interval $[-10, 10]$ because we figured since the coefficients of these numbers are derived from chemical reactions taking place in cells, their magnitudes probably won't blow up to high. Here are a few successes:

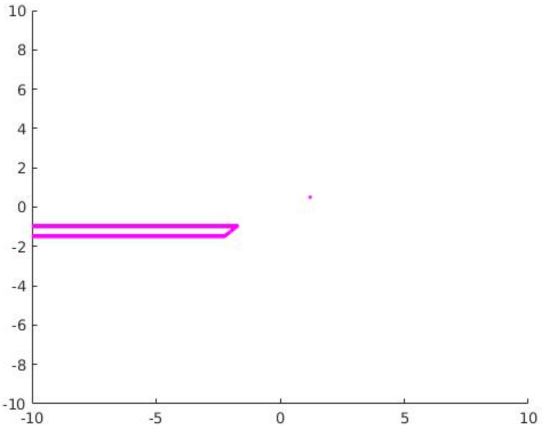


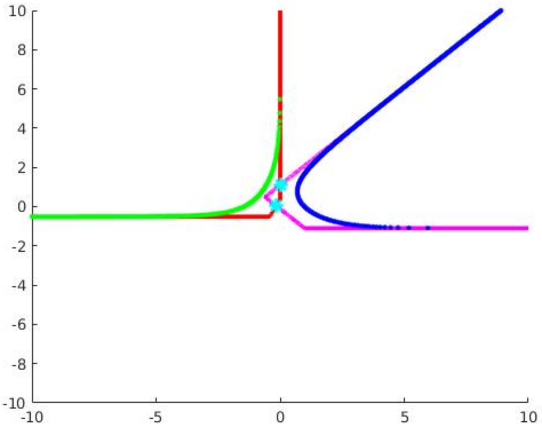




For all of these examples, the number of intersections of the zero sets and $\mathrm{ArchTrop}_+(f_1) \cap \mathrm{ArchTrop}_+(f_2)$ are the same! Now for a few failures...

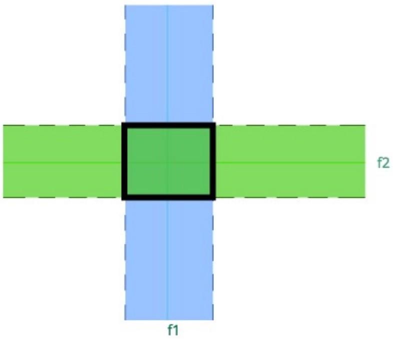




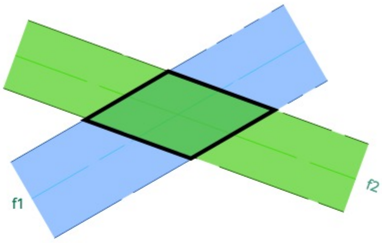


The first two failures relate to the fact that sometimes the number of connected components of a function's positive zero set does not correspond to the number of connected components of that function's $\mathrm{ArchTrop}_+$ construction. The final failure does not deal with this, however, notice that $\mathrm{ArchTrop}_+(f_1)$ and $\mathrm{ArchTrop}_+(f_2)$ intersect twice, while the $\mathrm{Log}|Z_+(f_1)|$ and $\mathrm{Log}|Z_+(f_2)|$ never intersect.

Intuitively, this happens because of the fact that $\mathrm{Log}|Z_+(f_i)|$ is within a distance of $\mathrm{Log}(4)$ of $\mathrm{ArchTrop}_+(f_i)$. When $\mathrm{ArchTrop}_+(f_1)$ and $\mathrm{ArchTrop}_+(f_2)$ intersect perpendicularly, the margin of error looks something like this:



However if $\mathrm{ArchTrop}_+(f_1)$ and $\mathrm{ArchTrop}_+(f_2)$ intersect at different angles, the margin of error increases:

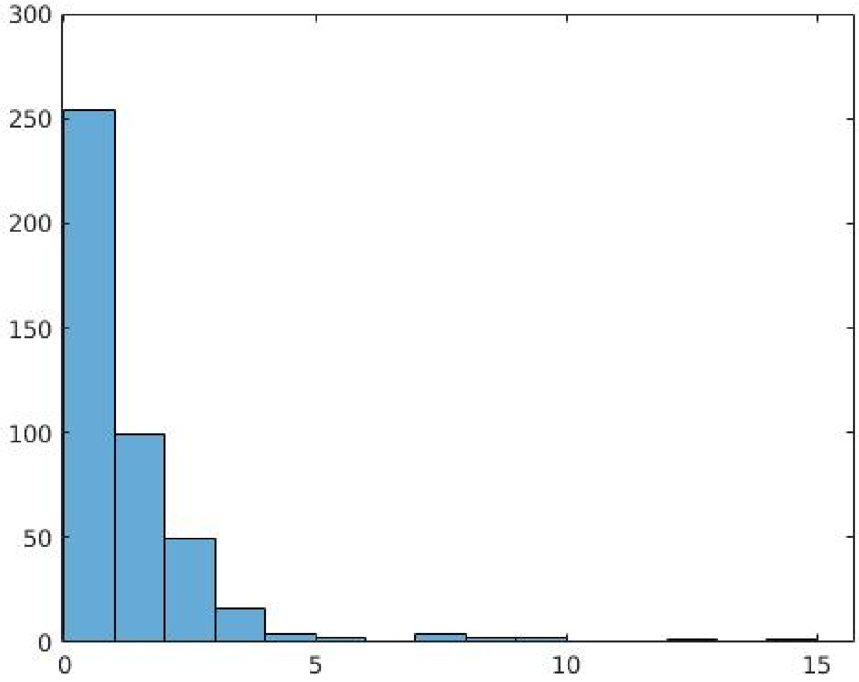


We, of course, have a theorem for that:

Theorem 6.4. For any 2×2 polynomial system non-degenerate F with supports having Cayley embedding A, the number of nonzero real roots of F depends only on the completed signed A-discriminant chamber containing F.

Essentially, if you are "deep" in a signed chamber of the A-Discriminant Variety, then the intersections are more likely to match up. This is because if you jiggle the coefficients, then you will still very clearly remain in the same signed chamber of the discriminant variety. However, if you are closer to the A-Discriminant Variety, then this is more likely to fail. This is because if you jiggle the coefficients, you may actually end up in a different signed chamber, since, after all, the A-discriminant variety tells us regions in coefficient space where the signs change.

Putting this aside for the moment, we ran through 1000 different examples using random coefficients found between $[-10, 10] \setminus 0$, and computed the Hausdorff distance between $\text{ArchTrop}(f_1) \cap \text{ArchTrop}(f_2)$ and $\text{Log}|Z_+(f_1)| \cap \text{Log}|Z_+(f_2)|$, to obtain the following histogram:



This shows that our constructions are more likely to work closer to 75% of the time rather than the initial 41% of the time.

7 Future Research

7.1 Generalizing our code

As we mentioned before, in order to develop our code, we took advantage of the fact that there were only 5 possible triangulations for each of our given functions. While this worked out nicely in this case, we would like to someday generalize our code in such a way that we don't need to write things in a case by case situation, seeing as the possibilities for triangulations can very well increase the more terms are added to a function.

7.2 Finding a condition number

If we let $h_0(\cdot) :=$ th number of connected components of '$\cdot$', then we are particularly interested in finding when the following holds:

$$h_0(Z_+(f)) = h_0(\text{ArchTrop}_+(f))$$

Although we have some idea of how probable it is that our constructions will work, it would be even better to find out which set of coefficients will change.

7.3 Stability and the Jacobian

Although we didn't get to this this summer, it is very much worth it to note that our systems, at the end of the day, are differential equations. While we do have a fairly reliable way of finding the positive roots, there still is room for things to be said about whether or not these solutions are stable, and we can figure that out by examining the eigenvalues of the corresponding Jacobian Matrix!