School of MATHEMATICAL AND STATISTICAL SCIENCES Clemson University

JULIANNE (McKAY) BARNHART

phone: (360) 393 - 1963 | email: mckay6@clemson.edu

Research Statement

1 Overview

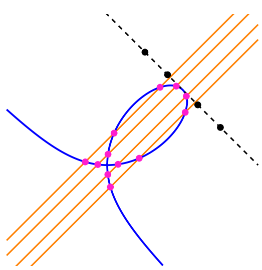
As a researcher in applied algebraic geometry, **I turn theory from algebraic geometry into effective computational tools for investigating polynomial systems and their solutions.** I proved a fact about the structure of solutions of a sparse polynomial system, which completed the sparse trace test. This work provided a new tool to computationally investigate sparse polynomial systems and other enumerative problems. As I continue to work on questions arising from applications of polynomial systems, I plan to bring students along through projects emphasizing experimentation, coding, and other applied methods.

2 Motivating Example: Classical Trace Test

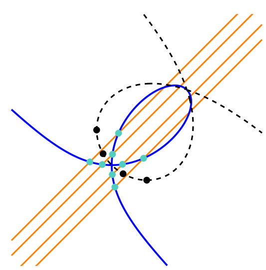
Consider $f(x, y) = x^3 + y^3 - 3xy$, and let $\mathcal{V}(f) = \{(x, y) : f(x, y) = 0\}$ be the variety of f, known as the folium of Descartes. Suppose we use a computer algorithm to compute the intersection points of $\mathcal{V}(f)$ with a line L.

Question: How can we be sure that the computer found all intersection points of $\mathcal{S} = \mathcal{V}(f) \cap L$? In other words, is the found solution set $\mathcal{S}$ *complete*?

The trace test verifies completeness of a solution set by checking for linearity of the *trace*, or sum of solutions, as the line L moves through a parallel family of lines. The trace is linear if and only if the solution set is complete. In figure 1a, there are three points in the intersection of $\mathcal{V}(f)$ with a line L, and their trace is given by one of the black points. As we move the line L through a family of parallel lines, the trace moves along the dotted black line. Therefore, a complete solution set implies a linear trace. On the other hand, in figure 1b, we have an incomplete solution set where only two points have been found. As we move the line L through a parallel family of lines, the trace of the two points moves non-linearly, as shown by the dotted black curve. Therefore, an incomplete solution set implies a non-linear trace.



(a) The trace of the complete solution set of $\mathcal{V}(f) \cap L$ moves linearly.



(b) The trace of the incomplete solution set of $\mathcal{V}(f) \cap L$ moves non-linearly.

Figure 1: The folium of Descartes $\mathcal{V}(f)$ in blue intersected with a family of parallel lines L in orange.

A key component of proving the classical trace test is knowing that the solutions within a set can freely permute with each other. In other words, it is necessary that the monodromy group is the full symmetric group. This fact is foundational to the converse part of the proof, where we begin with an incomplete solution set [3].

Computer algorithms which efficiently solve for solutions sets do not have a built-in way to check for completeness. Thus, the trace test is a vital computational tool for situations where completeness is important. My work contributes to the so-called sparse trace test [1], which is an analogue to the classical trace test for the setting of sparse polynomial systems.

3 Traces of Sparse Systems

A sparse polynomial system is a system where each polynomial has a defined set of exponents that are allowed to appear, called the *support* for that polynomial. When we consider the coefficients of the system as parameters, the trace is a function of the coefficients. In the sparse trace test, it is necessary to know a set of coefficients for which the trace is a linear function. In my master's thesis, I developed a method to classify the effect that each coefficient (or support element) has on the trace, an effect which I call *agency*.

Example 3.1. Let $\mathcal{A} = \{0, 1, 2\}$ be the support of a single-variable polynomial. The parametrized polynomial supported on $\mathcal{A}$ is $f = a_0 + a_1 x + a_2 x^2$. The quadratic formula gives two solutions x to f and reveals how the solutions depend on the chosen coefficients. The trace $\Sigma(f)$ is the sum of these two solutions, and so the trace also depends on the choice of coefficients.

$$\Sigma(f) = \frac{-a_1 + \sqrt{a_1^2 - 4a_2a_0}}{2a_2} + \frac{-a_1 - \sqrt{a_1^2 - 4a_2a_0}}{2a_2} = \frac{-a_1}{a_2}$$

The coefficient a_1 has a linear effect on the trace (*linear agency*), while the coefficient a_2 in the denominator has *non-linear agency*. Since the coefficient a_0 does not appear in the formula for the trace, it has *independent agency*.

I developed a method using homotopy continuation to quickly and accurately identify the agency of all coeffcients. Homotopy continuation is a foundational tool in applied algebraic geometry which allows the continual tracking of solutions as elements of a system change. My method formed the last piece of the puzzle for one direction of the sparse trace test: given a support meeting certain structural conditions, if the trace is a linear function of some set of coefficients, then the solution set is complete.

4 Restricted Monodromy Groups

Similarly to the proof of the classical trace test, the converse direction of the sparse trace test relies on knowing that the monodromy group in the sparse setting is the full symmetric group. Specifically, since the sparse trace test considers the trace as a function of some subset of the coefficients, I proved that the monodromy group restricted to one coefficient is the full symmetric group. The proof relies on group-theoretic characterizations, and so applies to many enumerative settings, not just sparse polynomial systems.

4.1 Monodromy

Monodromy occurs on a *branched cover* $\pi : X \to Y$. Without stating technical conditions, a branched cover is a map where a generic point c in the image space Y always has the same number of preimages in $\pi^{-1}(c)$, called a *fibre*, as any other generic point. The number of preimages above a generic point is called the *degree* of π. For sake of simplicity, I use a degree of d. The collection of points in the image space which do not have d points in their fibres is called the *branch locus* Δ.

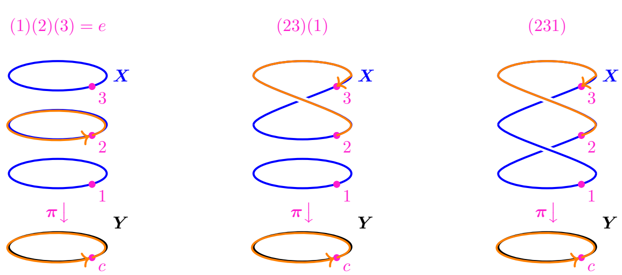


Figure 2: An example of monodromy on three different covering spaces. A loop from Y lifted to X in each picture corresponds to a different permutation of the fibre: $(1)(2)(3)$ in the first, $(32)(1)$ in the second, and (312) in the third.

Then, the monodromy action is a group action on the d points in a single fibre. It can be understood as follows (see figure 2): draw a loop from a generic point c in the coefficient space, and then lift the loop to the fibre $\pi^{-1}(c)$. The loop becomes many paths beginning and ending at the various points in $\pi^{-1}(c)$. Tracking the paths results in a permutation of the d points, and this permutation is an element of the symmetric group S_d. The monodromy group G_π of the branched cover is the group generated by all such permutations resulting from loops. It turns out that non-identity permutations are those resulting from loops which go around a point in the branch locus, so to compute a monodromy group, it is sufficient to draw all loops around Δ.

4.2 Monodromy for Sparse Systems

I consider a sparse polynomial system supported on $\mathcal{A}$ as a branched cover $\pi_\mathcal{A} : X_\mathcal{A} \to \mathbb{C}^\mathcal{A}$. In the sparse setting, the image space is the coefficient space $\mathbb{C}^\mathcal{A}$ consisting of tuples c representing a choice of coefficients for the polynomial system. Each fibre $\pi^{-1}(c)$ consists of c together with each solution to that particular system, forming the *incidence variety* $X_\mathcal{A}$. The support $\mathcal{A}$ determines the number of solutions to a generic choice of a polynomial system, so it makes sense to call this a branched cover because fibres above generic points all contain the same number of solutions.

It is known that the monodromy group $G_{\pi_\mathcal{A}}$ is S_d when $\mathcal{A}$ is non-lacunary, non-triangular, and when loops are drawn freely using all coefficients [2]. I introduce a restricted monodromy group G_{π_ℓ}, where loops can only be drawn within a complex line $\ell := te_\beta + c$ for some support element β. This means that all coefficients are fixed, and loops are drawn by changing one coefficient in the system.

Question: Let π_ℓ be the branched cover of a sparse polynomial system supported on $\mathcal{A}$ restricted to the line $\ell := te_\beta + c$. Under what conditions is $G_{\pi_\ell} = S_d$?

I prove a local-to-global property: if the restricted monodromy group G_{π_ℓ} contains a transposition, then G_{π_ℓ} along with the full monodromy group $G_{\pi_\mathcal{A}}$ are either both full symmetric or are both imprimitive. Furthermore, since I work with sparse polynomial systems which are non-lacunary and non-triangular, I know $G_{\pi_\mathcal{A}} = S_d$, so therefore if G_{π_ℓ} contains a transposition, it is S_d as well.

4.3 Components of the Branch Locus

The only loops which contribute non-trivial monodromy are those which go around a point in the branch locus Δ. I investigated the coefficient space the different ways that Δ can affect monodromy. I introduced a three-part decomposition of Δ which can be applied in many parametrized polynomial system settings. In the sparse polynomial setting Δ consists of choices for coefficients p which result in systems that do no have the d expected number of solutions. Loops around fibres which do not have d points can cause non-identity permutations.

Question: How can changing one coefficient of a sparse polynomial system affect the number of solutions?

There are three ways that changing a coefficient can affect monodromy, each giving a component of Δ.

1. **Discriminant locus Δ_μ: points p whose corresponding systems have solutions with multiplicity.**
   Changing one coefficient can cause a system to have a double solution, as in figure 3. A choice of coefficients $p \in \Delta$ which causes a solution of multiplicity two is called a *simply ramified point*. Simply ramified points cause transpositions in the monodromy group, since two solutions come together and swap.

2. **Boundary locus Δ_∂: points p whose corresponding systems lose solutions to infinity.**
   The number of solutions cannot decrease to zero, but can decrease by varying amounts, causing many different types of permutations in the monodromy group.

3. **Infinity locus Δ_∞: points p whose corresponding systems have a one-dimensional component.**
   Changing some coefficients could cause one polynomial to become a linear combination of other polynomials in the system, resulting in an underdetermined system with a one-dimensional solution component.

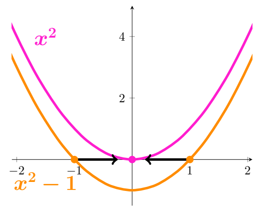


Figure 3: When the coefficient on the constant term is changed from -1 to 0, the two solutions of $x = -1, 1$ come together to be a solutions with multiplicity two at $x = 0$.

5 Future Work

As I pursue projects that bridge the gap between theory and computational tools in algebraic geometry, I plan to bring my own students along in this work. Applied algebraic geometry is a young and quickly growing field, meaning it is ripe with questions and applications that can be explored by undergraduate students. I often begin a project by experimentation in a coding language such as Python or Macaulay2. Setting up and running examples would be an accessible task from which undergraduate researchers could gain insight and confidence while contributing to new ideas.

For example, a project I am currently working on is about building neural networks to learn the real solutions of polynomial systems. Polynomial systems arise in many areas of engineering and science such as robotics, computer vision, and chemical reaction networks. While techniques from applied algebraic geometry find information about the set of all solutions, engineers and scientists are typically only interested in the real set of solutions. Often the set of real solutions is significantly smaller than the full set, meaning there is a gap in computational efficiency when we solve for all solutions. This project aims to utilize the power of machine learning to find only the real set of solutions for a polynomial system. The initial explorations for this project involved building small neural networks using PyTorch, the type of experimentation that would make for an excellent undergraduate research project.

I also plan to explore many questions relating to my work on trace tests and restricted monodromy groups. One question I am currently investigating is the structure of permutations that can result from loops around Δ_∂. For some permutations that result, I can point to aspects of the support to explain the type of permutation. For others, the reason is unknown. I am currently working with an undergraduate student in my research group to run experiments with the goal of understanding this aspect further. Other projects I intend to pursue are more open-ended, like extending the trace test to work in a more general setting, for an arbitrary variety and an arbitrary intersecting set.

Theory originating in algebraic geometry provides powerful tools to investigate computational and applied problems. I am excited to continue building and utilizing such tools as I pursue more research questions. I am prepared not only to continue my own work, but also to support undergraduate research in this field through hands-on experimentation and projects.

References

[1] T. Brysiewicz and M. A. Burr. Sparse trace tests. *Math Comp.*, 92:2893–2922, 2023.

[2] A. Esterov. Galois theory for general systems of polynomial equations. *Compositio Mathematica*, 155(2):229–245, February 2019.

[3] A. Leykin, J. I. Rodriguez, and F. Sottile. Trace test. *Arnold Mathematical Journal*, 4:113–125, 2018.