Functional Graphs of Polynomials on $\mathbb{Z}/p\mathbb{Z}$

Daniel Keliher, Lukas WinklerPrins

May 14, 2015

Abstract

We discuss some analytics of the associated graphs of polynomials on finite, prime-order fields and also give some graph statistics for those graphs for $p < 251$, and give our computational methods. We conclude with a brief literature review in an effort to summarize the current state of this problem. This was done as part of an undergraduate Group Indpendent Study Project (GISP) at Brown University with advisor Björn Sandstede.

Contents

1 Introduction

The problem of associated graphs on finite fields is motivated by arithmetic dynamics on finite fields. In general, arithmetic dynamics is concerned with dynamical systems consisting of a set S and map, f, from S to itself:

$$f : S \longrightarrow S$$

We're are also interested in what happens to some $x \in S$ under iterations of f. We call the set of points consisting of iterations of x under f the *orbit* of x:

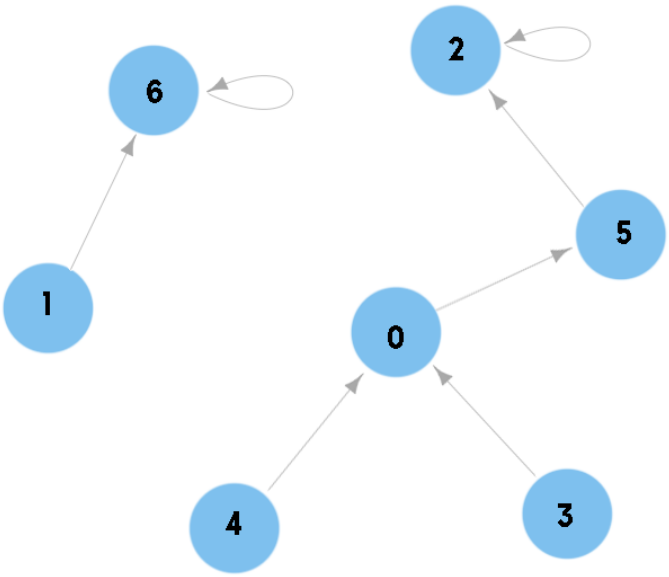
$$\text{Orbit}(x) = \{x, f(x), f^2(x), ...\}$$

Any $x \in S$ can be classified in four ways based on its orbit:

- x is periodic if Orbit(x) is finite and $f^n(x) = x$ for some n, and we say $x \in Per(\phi(x))$ (if $n = 1$, then x is $fixed$);
- x is preperiodic if Orbit(x) is finite and $f^{m+n}(x) = f^m(x)$ for some m, n, and we say $x \in Preper(\phi(x))$;
- If S is infinite, a point x can be wandering if Orbit(x) is infinite.

We are interested in graphs formed between the elements of prime order fields, $\mathbb{Z}/p\mathbb{Z}$ and polynomials $\phi(x) \in \mathbb{Z}/p\mathbb{Z}$, i.e. $\phi(x) \equiv \sum_{i=0}^{p-1} a_i x^i \pmod{p}$ (were each $a_i \in \mathbb{Z}/p\mathbb{Z}$). The associated graph for some ϕ over $\mathbb{Z}/p\mathbb{Z}$, will have the elements of the field as nodes, with a node u being connected to a node v if and only if $\phi(u) = v$.

We denote the associated graph of ϕ over $\mathbb{Z}/p\mathbb{Z}$ as $\mathcal{G}(\phi, \mathbb{Z}/p\mathbb{Z})$. When there is no risk of ambiguity, we may omit the field or polynomial. We also frequently omit the modulus by abuse of notation but it should be assumed at all times.



Associated graph for $p(x) = x^2 + 5$ on $\mathbb{Z}/7\mathbb{Z}$

Figure 1: Functional Graph for $p = 7$, $\phi = x^2 + 5$.

2 Analytic Results

Note that for ϕ of degree d on $\mathbb{Z}/p\mathbb{Z}$, there are $(p-1)p^d$ possible polynomials to construct.

2.1 Linear Classification

We proceed by thinking about moving from node to node as applying a polynomial, and generating the whole graph using iterations of that same polynomial. While slightly less direct, we feel it is more elucidating.

Lemma 2.1.1. *If $\phi(x) = ax + b$ where $a, b \in \mathbb{Z}/p\mathbb{Z}$, then $PrePer(\phi(x)) = \emptyset$.*

Proof. Let $\phi(x) = ax + b \in \mathbb{Z}/p\mathbb{Z}$. We seek to show there exists an n such that $\phi^n(x) = x$, or that all points are cyclic, so that $PrePer(\phi(x)) = \emptyset$. Then we find $\phi \circ \phi(x) = \phi^2(x) = a^2x + ab + b$, and through induction that

$$\phi^n(x) = a^n x + b\sum_{i=0}^{n-1} a^i \tag{2.1}$$

We find that $\forall x, a, b, p$, $\exists n$ such that $\phi^n(x) = x$, or that every point is periodic. The geometric series, when $a \neq 1$, in our general form can be written as

$$\sum_{i=0}^{n-1} a^i = \frac{1-a^n}{1-a}$$

We now find 2.1 is of the form

$$\phi^n(x) = a^n x + b\frac{1-a^n}{1-a}$$

and we find with $n = p - 1$ that

$$\phi^{p-1}(x) = a^{p-1}x + b\frac{1-a^{p-1}}{1-a} = x$$

by Fermat's Little Theorem when $a \neq 1$. In the case that $a = 1$ however, we note that ϕ is of the form $\phi(x) = x + b$, thus

$$\phi^n(x) = x + nb$$

Letting $n = p$ we find

$$\phi^p(x) = x$$

□

Corollary 2.1.2. *From the lemma, if $a = 1$, the length of a cycle of $\mathcal{G}$ is a divisor of p and if $a \neq 1$, the length of a cycle divides $p - 1$.*

The proof of this is clear from Lemma 2.1.

Corollary 2.1.3. *If $\phi(x) = ax + b$ where $a, b \in \mathbb{Z}/p\mathbb{Z}$ and $a > 1$, then $\mathcal{G}(\phi(x))$ has 2 or more components.*

Proof. For $a > 1$ we have demonstrated that $\phi^{p-1}(x) = x$, thus the length of any cycle must be a divisor of $p - 1$. As all points are in $Per(\phi(x))$, there must be at least two components. □

Corollary 2.1.4. *If $a = 1$, then $\mathcal{G}(\phi)$ has either 1 or p components. In particular there are p components when $b = 0$, and 1 component otherwise.*

Proof. Since the length of any cycle of $\mathcal{G}$ must be a divisor of p, the length is either 1 or p. However if $\phi(x) = x$ then clearly there are p components. In any other case ($b \neq 0$), there is one component since clearly $x \neq x + b$ for at least one x. □

Observation 2.1.5. $\mathcal{G}(\phi)$ for $\phi(x) = ax + b, a \neq 1$ over $\mathbb{Z}/p\mathbb{Z}$ always has at a fixed point.

Clearly there is always a solution in $\mathbb{Z}/p\mathbb{Z}$ to $x = ax + b \implies x(1 - a) = b$.

Lemma 2.1.6. *If $\phi_b(x) = ax + b$ and $\phi_c(x) = ax + c$, then $\mathcal{G}(\phi_b) \cong \mathcal{G}(\phi_c)$.*

Proof. A constant shift in value adds nothing structurally to a graph; it permutes all labels by a fixed amount in $\mathbb{Z}/p\mathbb{Z}$. □

2.2 Graph Analysis

In practice we used a standard directed adjacency matrix. The first row corresponds to the 0 node, the second row to the 1 node, and so forth for both rows and columns. With this, typical graph measurements such as degree, number of components, and connectivity can be studied.

It should be intuitive that, given our polynomial method for graph construction, each node has outdegree 1 and indegree of at maximum d, where d is the degree of our polynomial $\phi(x)$.

Lemma 2.2.1. *For any polynomial $\phi(x)$ acting on $\mathbb{Z}/p\mathbb{Z}$, the generated functional graph is planar.*

Proof. Let $\mathcal{G}(\phi(x), \mathbb{Z}/p\mathbb{Z})$ be the graph generated by $\phi(x)$ on $\mathbb{Z}/p\mathbb{Z}$. We seek to show for any polynomial and field our graph will be planar, which, by Kuratowski's theorem, means that neither K_5 nor $K_{3,3}$ exists as a subgraph of $\mathcal{G}$.

When $p < 5$ the generated graph is trivially planar as $|\mathcal{G}| = p$. For $p \geq 5$, we risk the possibility of a forbidden graph. However, K_5 requires 25 edges connected between its 5 nodes and $K_{3,3}$ requires 18 between its 6 nodes. In other words, K_5 and $K_{3,3}$ both require more edges between their nodes than they have edges. In a functional graph there are only as many edges between nodes as there are nodes.

Thus neither K_5 nor $K_{3,3}$ can exist as a subgraph of $\mathcal{G}$ and by Kuratowski's theorem, the functional graphs are planar. □

2.3 Graph Isomorphism

We consider two graphs isomorphic if they are structurally identical up to labeling.

Definition 2.3.1. With V_G as the set of nodes of a graph G, any bijective function $f : V_A \to V_B$ defines an **isomorphism** between graphs A and B.

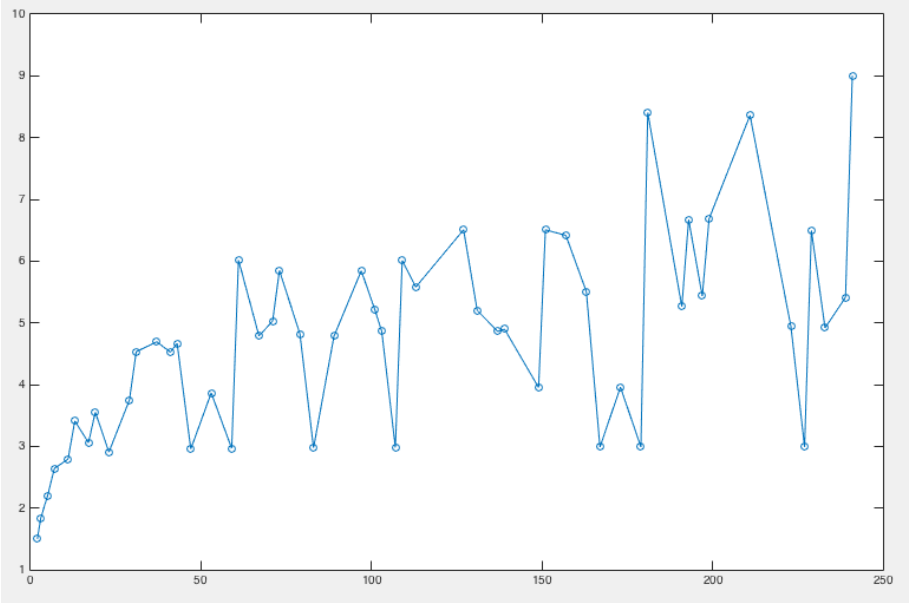
Konyagin et. al. [5] found that the number of distinct non-isomorphic) graphs for a given (p, d) polynomial graph family, $N_d(p) = O(p^{d-1})$.

3 Computational Results

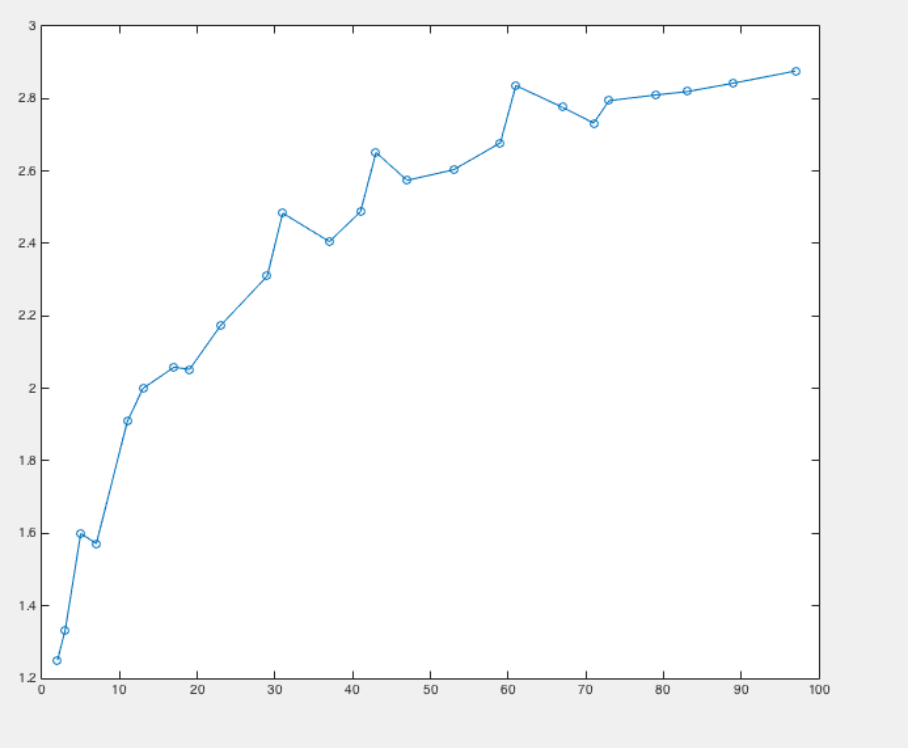
For implementation methods and out MATLAB code, please see Section 6.

3.1 Average Number of Components

A question of particular interest to the authors is this: for some $\phi(x)$ and $\mathbb{Z}/p\mathbb{Z}$, how many components will $\phi, \mathbb{Z}/\sqrt{\ }\mathbb{Z}$ have? It should be noted that some existing work seeks to answer this question (see 4: Literature Review). We have created a computational framework (see 6: Code) to begin thinking about this question, in which we compute the *average* number of components of n^{th}-degree polynomials over a fixed $\mathbb{Z}/p\mathbb{Z}$. The following two rough plots showing average number of compoenents vs. size (p) of field for polynomials of a given degree.



Average number of components for all linear polynomials on $\mathbb{Z}/p\mathbb{Z}$, $p \leq 251$



Average number of components for all quadratic polynomials on $\mathbb{Z}/p\mathbb{Z}$*,* $p \leq 97$

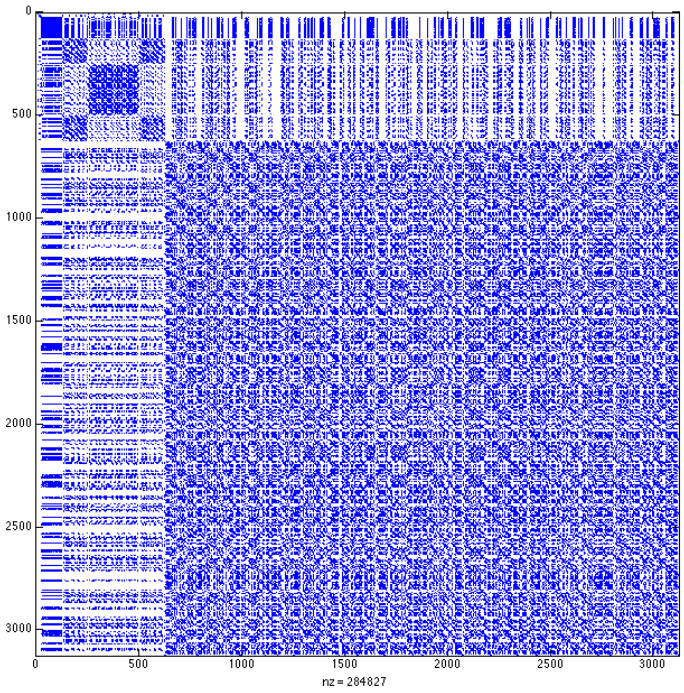
Unfortunately, computational limitations (lack of computing power and time) have made these the only computations that we can do that offer any insight.

3.2 Observations, Patterns, and all things Emergent

A natural question when considering polynomials and their functional graphs is this: When do two polynomials generate isomorphic graphs? Further, can two polynomials ever have *identical* functional graphs?

A component of our code seeks to look into this questions (See Appendix for code). We generate all possible polynomials over a particular prime field ordered by leading coefficient (the subsequently sub-ordered by the next coefficient), and consider each graph. The output is $p^p \times p^p$ matrix where the (i, j) entry is a 1 if the i^{th} polynomial's functional graph is isomorphic to the j^{th} polynomial's functional graph, and a 0 if the two polynomials'

functional graphs are not isomorphic. Clearly such matrices will have diagonals populated by 1s and will by symmetric about the diagonal. Below we show the spy diagrams (blue dots for 1s) of some sample calculations:



The same computational experiment, checking *all* polynomials of some prime order field, was done but this time checking for *identical* graphs. In each case ($p \leq 7$), we simply generate a $p^p \times p^p$ identity matrix. That is, there are no polynomials in these fields that generate identical functional graphs.

This results seem to, at least loosely, support the following:

Conjecture: *There is a one-to-one relation between the p^p possible polynomials on $\mathbb{Z}/p\mathbb{Z}$ and the p^p possible functional graphs.*

$$\{\phi(x) \in \mathbb{Z}/p\mathbb{Z}[x]\} \overset{\sim}{\longleftrightarrow} \{\mathcal{G}(\phi) \text{ s.t. } \phi(x) \in \mathbb{Z}/p\mathbb{Z}[x]\}$$

4 Literature Review

Here we review some relevant and sopporting papers dealing with associated graphs of mappings on finite sets, particularly finite fields. There are a number of excellent introductions to the field of Arithmetic Dynamics, namely Joseph H. Silverman's *The Arithmetic of Dynamical Systems*.

4.1 Reviews

Periods of Rational Maps Modulo Primes (Benedetto et. al.) [1] Using deep number theory, this paper focuses on analytical results regarding the size and type of preperiodic and periodic sets of rational fields and rational maps. In the latter parts of the paper, they find a closed form for the probability of a cycle of a particular length. Then, using ramification theory, Benedetto et. al. draw statements regarding graph statistics comparisons to random maps.

Random Mapping Statistics (Flajolet and Odlyzko) [2] Flajolet and Odlyzko asymptotically evaluate graph statistics for random mappings (dynamics within finite fields that are uniformly randomly created, rather than owing to a determined polynomial). As we have found, closed forms for iterations of ϕ become unusable for $d > 1$ with exceptions for a few special cases. Thus they turn to analysis of random mappings following the postulate that a polynomial behaves like a random mapping, investigating the number of components, number of cyclic nodes, and number of terminal nodes (those with no preimage). The paper also provides clear definitions for tail-length λ, cycle-length μ, and rho-length $\rho = \lambda + \mu$, which we find interesting as statistics.

Graph Components and Dynamics over Finite Fields (Flynn and Garton) [3] Flynn and Garton deal exactly with the problem in questiongraph-theoretic questions about finite fieldsvia some combinatorial methods. They find bounds for the average number of components for the associated graphs of polynomials and rational maps of fixed degree over finite fields. Further, they connect the problem to the more well understood behavior of randoms

maps on finite sets, particularly in the quadratic polynomial case. Improvements to be made would ideally be removing the constraints that have many of these results contingent on the degree of the polynomial being very large relative to the order of the field.

Efficient Algorithms for Graph Manipulations by (Hopcroft and Tarjan) [4] Hopcroft and Tarjan famously introduce their efficient method for computing the number of components in a graph in pseudocode in this paper.

Functional Graphs of Polynomials Over Finite Fields (Konyagin et. al.) [5] This paper takes interest in questions similar to our own, and takes special interest in the computational side of graph investigation and use in random processes. Allowing $N_d(q)$ to be the number of distinct (non-isomorphic) graphs generated by all polynomials of degree d on $\mathbb{F}_q$, they find some asymptotic bounds for $N_d(q)$ using combinatorial techniques and frequent assumptions of $\gcd(d, q-1) \geq 2$. They show that computationally using an adjacency list instead of adjacency matrix should give $O(n \log n)$ rather than $O(n^2)$ when testing isomorphism.

Expected Number of Components Under a Random Mapping Function (Kruskal) [6] Kruskal gives an analytic argument for expected number of graph components. The analytic argument given is interesting though perhaps not enlightening. The end result used [5]. I dont think we have follow a similar argument ourselves since their is an implicit assumption of randomness, while well be dealing with specific maps. Maybe the probabilistic method and explicit method can interface.

Eigenvalues of the Laplacian and their Relationship to the Connectedness of a Graph (Marsden) [7] This REU paper is an elementary summary of properties of the Laplacian, which are more rigorously investigated by Mohar. Walks through how the multiplicity of the zero eigenvalue is the number of components in a graph using a block-matrix method. Also introduced is Cheegers Inequality, which relates the expansion of a set to the

second-largest eigenvalue of the Laplacian.

The Laplacian Spectrum of Graphs (Mohar) [8] This is a non-journal survey of relationships between graph structures and spectrum of Laplacian matrix, with emphasis on connectivity of the graph. It provides statements of theorems and corollaries regarding the relationships (mostly via inequalities) of the second eigenvalues, λ_2, between graphs, their complements, spanning subgraphs, and more. Numerous references to Miroslav Fielders work on the second eigenvalue, which he calls the algebraic connectivity. This value can be used in many relationships connecting it to computational complexity and values of various graph metrics, such as diameter.

5 Further Explorations

There are numerous paths yet unexplored in this field. We seek to build upon these findings in the future. Some of our recommended directions include:

- Explore the degree distributions of graphs within a certain (p, d) regime.
- Seek an algorithm for constructing the graphs from polynomials created by products of linear terms.
- Investigate the near-randomness of $d \geq 4$.
- Choose to compare other statistics, such as size of preperiodic sets, component diameters, and cycle lengths, more fully.

6 Code

To access out code, please visit: `https://github.com/ltwp/KW-AD`.

The rest of this section serves as a ReadMe for the code. There are 8 files to our MATLAB codebase and we recommend running all in the same directory. All have some commenting

- `adjacency_list.m` takes an adjacency matrix and returns an adjacency list in the form of a vector. The i^{th} member of the vector is the index of the node that node i points to.

- `eval_polynomial.m` Takes a vector of coefficients (output by `get_graphs.m`), an x value, and a p value (for $\mathbb{Z}/p\mathbb{Z}$), and evaluates the corresponding polynomial at x in $\mathbb{Z}/p\mathbb{Z}$ and returns that as output.

- `get_graphs.m` is one of the most important parts. This function takes arguments d and p and generates all polynomials within this family. It returns both a cell array of all the matrices, and a large matrix of all coefficients, where each row of the matrix corresponds to a polynomial. For more documentation see the code. There is a line early in the program that can allow for the degree of the polynomial to be $\leq d$, rather than strictly $= d$.

- `graphs_isomorphic.m` takes two adjacency matrices and an optional third argument of 1 or 0. It returns true if the two corresponding graphs are isomorphic, false otherwise. If the third argument is turned on, the function will only return true if the graphs are exactly equal.

  This is the most computationally difficult part of this codebase. The question of isomorphic graphs is NP but not known to be NP-complete or in polynomial time.

  Our method performs four quick checks: are the graphs the same size? Do they have the same number of components? Do they have components of the same size? And does each component have cycles of the same length? These questions are all computable in sub-$O(n^2)$ time. If the matrices pass all of these checks, all that is currently implemented is a brute-force method for constructing all bijections to relabel adjacency lists. This can be done better, and implementation details are discussed in [5].

  It uses `adjacency_list.m`, `loop_lengths.m`, and `n_comps.m`, so they must share a directory.

- `loop_lengths.m` takes an adjacency matrix as well as a component labeling vector (the first output from `n_comps.m`) and returns a vector containing the lengths of the cycles in each component. The ordering of the vector is arbitrary.

- `make_adjacency.m` takes an adjacency list and returns an adjacency matrix (effectively the inverse of `adjacency_list.m`). It is not used anywhere in the code base.

- `n_comps.m` takes an adjacency matrix, A, and returns a list (as a vector), where the i^{th} entry contains the index of the component node i belongs to. In other words, the maximum value from this list gives you the total number of components, but the list also tells you which nodes are interconnected. An optional second output returns the size of the i^{th} component in the i^{th} position.

  `n_comps.m` and `loop_lengths.m` behave similarly and could be combined to save computation time if need be. The graph is navigated (easy as outdegree is always 1) from the first node and a component is labeled once the cycle is found (by returning to a previously visited node).

- `progress_bar.m` is a support program to help visualize the progress. Helpful when patience is thin. It has already been inserted into some code.

  Other files are support files for computing statistics and isomorphism checks of our choosing.

References

[1] Benedetto, R., Ghioca, D., Hutz, B., Kurlberg, P., Scanlon, T., and Tucker, T. (n.d.). Periods of rational maps modulo primes. *Mathematische Annalen*, 637-660.

[2] Flajolet, P., Odlyzko, A.M. Random Mapping Statistics. *Advances in Cryptology*, Ed. J.J. Quisquater, J. Vandewalle, Springer-Verlag Berlin Heidelberg, 1990, pp. 329-354.

[3] Flynn, R., and Garton, D. (n.d.). Graph Components And Dynamics Over Finite Fields. *International Journal of Number Theory*, 779-792.

[4] Hopcroft, J., and Tarjan, R. (1973). Algorithm 447: Efficient algorithms for graph manipulation. *Communications of the ACM*, 372-378.

[5] Konyagin, S., Luca, F., Mans, B., Mathieson, L., and Shparlinski, I. (2013). Functional Graphs of Polynomials Over Finite Fields. *ArXiv, 1307.2718v2.*

[6] Kruskal, M. (n.d.). The Expected Number of Components Under a Random Mapping Function. *The American Mathematical Monthly*, 61(6), 392-397.

[7] Marsden, A. (n.d.). Eigenvalues of the Laplacian and Their Relationship to the Connectedness of a Graph. (REU.)

[8] Mohar, B., The Laplacian Spectrum of Graphs, in *Graph Theory, Combinatorics, and Applications*, Vol. 2, Ed. Y. Alavi, G. Chartrand, O. R. Oellermann, A. J. Schwenk, Wiley, 1991, pp. 871898.