CS 710: Complexity Theory Date: Spring 2021

Final Project: A Report on Entropy Waves, the Graph Zig-Zag Product, and New Constant-Degree Expanders

Instructor: Jin-Yi Cai Report Author: Juan Rios

1 Introduction

This report is about the 2002 paper "Entropy Waves, the Zig-Zag Graph Product, and New Constant-Degree Expanders" by Omer Reingold, Salil Vadhan, and Avi Wigderson [1]. The *zig-zag product* is an operation that takes a large and a small graph, resulting in a new graph. This new graph inherits roughly the same size as the large parent, the degree of the small parent, and the expansion properties of both. One important concept is to view expanders as propagators of "entropy waves". These expanders transform high entropy concentration distribution in an area, to a distribution of a more "dissipated" concentration.

The paper focuses on defining the zig-zag product, along with the construction and properties of expanders. Additionally, an overview of two zig-zag product variants will be given. This report will summarize the paper to show the understanding gained by the report's author.

2 Background

Before the main findings and analyses of the paper are presented, some of the fundamental concepts and definitions needed to understand the paper are explored in this section. Recall that a graph $G(V, E)$ is a construct that represents a set of nodes or vertices V, where each vertex may or may not be related to other vertices in V. These relations to other vertices can be represented by the set of edges E. In this paper, these graphs are taken to be undirected, meaning all edges are bidirectional. These edges may be parallel, so that two edges can be incident upon two shared vertices, labeling G as a *multi-graph*. Lastly, G can contain loops, meaning that a vertex can connect to itself with an edge.

2.1 Expanders

Consider the concept of an expander graph. An expander graph is sparse. This means that there are few edges compared to the total number of edges possible. These graphs are nevertheless highly connected, meaning that a large number of elements (vertices and edges) need to be removed to separate the graph into two isolated graphs. Additionally the paper provides some informal definitions of these expander graphs:

- Every set of vertices has many neighbors.
- Every cut across the graph will cut through many edges.
- A random walk on the graph will quickly converge to a stationary distribution.

- The graph has strong isoperimetric inequality. This inequality is the relationship between the surface area of a set and its volume. Think of a sphere as a 3D representation of this set. The sphere encloses the maximum volume possible with its surface area. A graph with a strong isoperimetric inequality will have a smaller volume enclosed by its surface area.

Why are these expander graphs important? The paper provides examples of these graphs and their application in computer science: from network design, to coding theory, to cryptography. However, explicitly creating these graphs is a challenge, thus it is the aim of researchers to formulate new ways to create these graphs efficiently. Define an infinite family of expander graphs with the following properties: (1) The family consists of D-regular graphs, meaning that each vertex has D number of connections. D is the degree of a vertex, which is the number of edges incident on this vertex, and (2) for every graph in this family, the eigenvalue of the adjacency matrix of the graph with the second largest absolute value, λ, is bounded by D such that $\lambda < D$.

To expand this concept, the adjacency matrix M of a graph $G(V, E)$ with N vertices, is an N-by-N matrix, where a row index u represents a vertex, and column index v represents a second vertex, and the $(u, v)^{th}$ element of M is the number of edges connecting u to v. For a D-regular graph, $\frac{M}{D}$ gives a normalized adjacency matrix, which is also the probability distribution of the graph if a token where to take a random walk on this graph. This normalized matrix's largest eigenvalue is 1, but the primary interest is finding the second largest eigenvalue λ, which plays a role in the convergence rate of the random walk, along with the expansion properties of the graph.

The paper references works by other researchers that consider algebraic construction of these expanders, where λ is estimated in a sophisticated way, making it not intuitive to understand why the graphs are expanders. However, given the name of a vertex, the neighbors can be computed in polynomial time. The paper then states a distinct path to build these graphs combinatorically, as proposed by Ajtai in 1994 [2]. These graphs are built by starting with an arbitrary cubic ($D = 3$) graph with N vertices and then a proceeding with asequence of polynomially-many operations that gradually turn the cubic graph into an expander. However, the resulting graphs are not simply described, and do not have the applicability of the algebraically-constructed graphs. The paper [1] then provides a combinatorial construction of these graphs, along with a simple analysis proving expansion. This construction is iterative, and needs a basic building block - a single expander of constant size. Operations applied to this graph generate another graph with increased size but unchanged degree and expansion properties. This operation is the zig-zag product.

2.2 Graphs and Rotations

Recall that the graphs relevant to this paper are undirected, thus the adjacency matrix is symmetric. Suppose that the edges leaving each vertex of a graph G are labeled from 1 to D, thus a vertex's i^{th} edge connects to the i^{th} neighbor. The *rotation map* of G, is defined as $\text{Rot}_G(v, i) = (w, j)$. This means that the i^{th} edge of v leads to vertex w, and this edge also happens to be w's j^{th} edge. A family of $\mathfrak{G}$ of graphs is explicit if for every $G \in \mathfrak{G}$, Rot_G is found in time poly(log N), where N is the number of vertices. The construction of the expanders are done iteratively by operations that build new graphs from previous ones. The definition of these new graphs will show that the Rot_G of a new graph G can be computed with access to an oracle. Recall that an algorithm with access to an oracle can use the oracle to evaluate a function f in one time step per evaluation.

[1] For abbreviation "the paper" will be used to refer to the Reingold, Vadhan, and Wigderson paper that this report focuses on.

The reason that rotation maps are used is because so far, these D-regular graphs are assumed to have D-coloring of the edges. Recall that this edge "coloring" is an assignment of colors to edge on a graph such that no two edges incident on the same vertex have the same color. The zig-zag product does not retain this property hence the need to describe maps with rotation maps.

2.3 Second-Largest Eigenvector

The second-largest eigenvector of a graph G is $\lambda(G)$:

$$\lambda(G) = \max_{\alpha \perp 1_N} \frac{||M\alpha||}{||\alpha||} \tag{1}$$

where α is a vector perpendicular to a constant vector of length N, 1_N, and α is the vector that maximizes (1). Take $\pi \in [0,1]^N$ to represent the probability distribution on the vertices of G. π can be broken down to the sum of $\frac{1_N}{N} + \pi^{\perp}$, and multiplying π by M yields $M\pi = \frac{1_N}{N} + M\pi^{\perp}$. $\frac{1_N}{N}$ is an uniform distribution, and an eigenvector of M and thus remains unchanged by M. $M\pi$ can be interpreted as the probability distribution of selecting a vertex v, and then moving to a uniformly-selected neighbor of v. using (1), once can express:

$$\lambda(G) \geq \frac{||M\pi^{\perp}||}{||\pi^{\perp}||} \tag{2}$$

From (2), one can see how $\lambda(G)$ is a measure of how quickly a random walk on G converges to a uniform distribution. The smaller that $\lambda(G)$ gets, the better the expansion property. Relating to the discussion on expanders, an expander is a graph whose $\lambda(G) \leq \lambda < 1$. Now one can summarize a D-regular undirected graph G on N vertices with $\lambda(G) \leq \lambda$ as an (N, D, λ)-graph.

2.4 Squaring and Tensoring

Two operations used in the construction of the expander are squaring and tensoring. These operations will be described in terms of the rotation map and their effects on the eigenvalues. Let G be a (N, D, λ)-graph with $\text{Rot}_G(v_{i-1,k_i}) = (v_i, l_i)$, and taking G to the t power gives G^t to be a (N, D^t, λ^t)-graph with $\text{Rot}_{G^t}(v_0, (k_1, k_2, ..., k_t)) = (v_t, (l_t, l_{t-1}, ..., l_1))$. Additionally, Rot_G^t is computable in time poly(log N, log D, t) with t oracle queries to Rot_G.

The tensor product of two vectors a of size $N_a$and b of size N_b is $a \bigotimes b$, and is a vector of size $N_a N_b$, where the elements of this tensor product are all the possible products of all the elements in a with all the elements in b. For a graph G_1 and graph G_2 which are (N_1, D_1, λ_1) and (N_2, D_2, λ_2) graphs respectively, and with $\text{Rot}_{G_1}(v, i) = (v', i')$ and $\text{Rot}_{G_2}(w, j) = (w', j')$ respectively, then $G_1 \bigotimes G_2$ is a $(N_1 N_2, D_1 D_2, max(\lambda_1 \lambda_2))$ graph with $\text{Rot}_{G_1 \bigotimes G_2}((v, w), (i, j)) = ((v', w'), (i', j'))$. $\text{Rot}_{G_1 \bigotimes G_2}$ is computable in time poly(log $N_1 N_2$, log $D_1 D_2$) with an oracle query to Rot_{G_1} and another oracle query to Rot_{G_2}.

3 Zig-Zag Product

The paper introduces the zig-zag product, an operation that takes a large graph and a small graph. The resulting graph inherits roughly the same size as the larger graph, the degree of the smaller one, and the expansion properties of both graphs. Let G_1 be a (N_1, D_1, λ_1) graph, and

G_2 an (N_2, D_2, λ_2) graph. The zig-zag product of these graphs is denoted as $G_1 \textcircled{z} G_2$ and it is an $(N_1 D_1, D_2^2, \lambda_1 + \lambda_2 + \lambda_2^2)$ graph. This product is useful in creating arbitrarily large graphs with a specified degree.

The paper describes how every vertex v of G_1 is "blown up to a cloud" of D_1 vertices, thus a vertex v is blown into a "cloud" or set of vertices of size D_1 such that this set contains vertices $(v, 1), (v, 2), ..., (v, k), ..., (v, D_1)$. These new vertices have two components: the first component is a vertex of G_1 and the second component is a vertex of G_2. Every edge e in G_1 is associated with two vertices of $G_1 \textcircled{z} G_2$, (v, k) and (w, l), and $\text{Rot}_{G_1}(v, k) = (w, l)$. The vertices in the cloud are connected to each other using the edges of G_2. Since the graph product has D_2^2 edges, these edges originating from G_2 are labeled in two components (i, j). These edges of $G_1 \textcircled{z} G_2$ are defined as a random walk that can be broken down into 3 sub-steps as follows:

1. Starting with a vertex (v, k), move to a neighboring vertex (v', k) within the cloud using an edge of G_2. This step is labeled "zig" and affects only the second component, according to i.
2. Using an edge of G_1, jump across clouds to get from (v', k) to (w, l').
3. Move to a neighboring vertex (w, l) within this new cloud using an edge of G_2. This step is described as "zag" and affects only the second component, according to j.

The discussion will now deviate and touch on the idea of the zig-zag product as a entropy wave propagator. Entropy in a graph is a measure of the information rate achievable through communication over a channel. A random step in an expander increases the entropy of a distribution of vertices as long as the entropy is not too close to uniform. Considering the distribution on the vertices of the $G_1 \textcircled{z} G_2$ graph (v, k). Recall that step 1 affects the second component of the vertex k. The paper presents two cases:

1. the distribution on k is not too uniform, then Step 1 "works" (makes the distribution more uniform). Step 2 is viewed as a permutation, and step 3, like step 1, is a random step on G_2. Thus these steps won't undo step 1's work - making the distribution more uniform.
2. the distribution on k is close to uniform, step 1's work is futile, but step 2 is viewed as a random step in G_1, increasing the entropy in component 1, but because step 2 is simply a permutation, the entropy decreases in component 2. Step 3, by expanding G_2, increases the entropy of component 2. The paper then continues and states step 2 facilitates entropy in either step 1 or 3, since step 2 does not introduce randomness to the distribution on vertices of the product.

The paper states that this intuition is followed by the proof of the zig-zag product described later in this report. Returning to the discussion of the mechanics of the zig-zag product, If the following rotation maps for G_1, G_2 are defined: $\text{Rot}_{G_1}(v, k')$, $\text{Rot}_{G_2}(k, i)$, and $\text{Rot}_{G_2}(l', k')$, then $\text{Rot}_{G_1 \textcircled{z} G_2}((v, k), (i, j)) = ((w, l)(j', i'))$, and can be computed in time poly($\log N_1$, $\log D_1$, $\log D_2$), with one oracle query to G_1, and two oracle queries to G_2.

3.1 Analysis of Zig-Zag Product

The paper provides two analyses: one where λ of the product is a basic but sub-optimal bound on the second eigenvalue, and a second tighter bound which is less intuitive. First, consider the basic

bound. The aim behind the analysis is to show that taking a random step on $G_1 ⓩ G_2$ results in a more uniform distribution. Let M be the normalized adjacency matrix of $G_1 ⓩ G_2$, let π be the probability distribution on the vertices of $G_1 ⓩ G_2$ such that:

$$\pi = u_{N_1,D_1} + \alpha \tag{3}$$

where u_{N_1,D_1} is just a uniform distribution on vertices of $G_1 ⓩ G_2$, and $\alpha \in \mathbb{R}^{N_1,D_1}$ is the non-uniform component of π such that $\alpha \perp u_{N_1 D_1}$. For every vertex $v \in N_1$, define a vector $\alpha_v \in R^{D_1}$. Recall how each v in G_1 was blown up to a cloud of D_1 vertices, then define π_v as a multiple of the conditional distribution on this v cloud. Furthermore, α_v can be decomposed as $\alpha_v^{\|} + \alpha_v^{\perp}$, the first term is the component parallel to 1_D, and the second term is the component orthogonal to 1_D, then a decomposition of α can be as follows:

$$\alpha = \sum_v e_v \bigotimes \alpha_v = \sum_v e_v \bigotimes \alpha_v^{\|} + \sum_v e_v \bigotimes \alpha_v^{\perp} = \alpha^{\|} + \alpha^{\perp} \tag{4}$$

where $e_v \in \mathbb{R}^{N_1}$ is the v^{th} standard basis vector. This decomposition corresponds to the two cases presented regarding the distribution on the vertices of $G_1 ⓩ G_2$. $\alpha^{\|}$ corresponds to the uniform distribution, and $\alpha^{\perp}$ corresponds to the non-uniform distribution. Now the interest is how M acts on these two vectors. Take A and B to be the normalized adjacency matrices of G_1 and G_2 respectively. Recalling the three steps of the zig-zag product, M can be decomposed into the product of three matrices so that each matrix represents a step. Denote $\tilde{B} = I_{N_1} \bigotimes B$, where I_{N_1} is the identity matrix of size N_1^2. This tensoring with B gives the adjacency matrix of the graph at the time that where each of the D_1 vertices in a cloud are connected by the G_2 edges. Additionally, take $\tilde{A}$ [1] to represent step 2, the permutation matrix corresponding to Rot_{G_1}. Now M can be defined as:

$$M = \tilde{B}\tilde{A}\tilde{B} \tag{5}$$

recalling equation (1) which bounds M, one can combine with (5) to obtain:

$$M\alpha \cdot \alpha = \tilde{B}\tilde{A}\tilde{B}\alpha \cdot \alpha = \tilde{A}\tilde{B}\alpha \cdot \tilde{B}\alpha \tag{6}$$

One can expand $\tilde{B}\alpha = \tilde{B}(\alpha^{\|} + \alpha^{\perp}) = \alpha^{\|} + \tilde{B}\alpha^{\perp}$. This can be done because $\alpha^{\|}$ corresponds to a case with uniform distribution as discussed earlier, and taking a G_2 step does not achieve anything. Taking this into account results in:

$$M\alpha \cdot \alpha = \tilde{A}(\alpha + \tilde{B}\alpha^{\perp}) \cdot (\alpha + \tilde{B}\alpha^{\perp}) \tag{7}$$

and by expansion one gets

$$|M\alpha \cdot \alpha| = |\tilde{A}\alpha^{\|} \cdot \alpha^{\|}| + 2||\alpha^{\|}|| \cdot ||\tilde{B}\alpha^{\perp}|| + ||\tilde{B}\alpha^{\perp}||^2 \tag{8}$$

The following two bounds to some of the terms in (8) are bounded: First, $\frac{||\tilde{B}\alpha^{\perp}||}{||\alpha^{\perp}||} \leq \lambda_2$. This bound follows from equation (1), where the max of the expression is the second-largest eigenvalue of G_2, λ_2. The intuition here is that in the case where the conditional distributions are not uniform, the distributions become more uniform when taking a random G_2 step. The second bound is

[1] The relationship between A and $\tilde{A}$ will be shown later.

$\frac{|\tilde{A}\alpha^{\|}\cdot\alpha^{\|}|}{\alpha^{\|}\cdot\alpha^{\|}} \leq \lambda_1$. This bound follows the intuition that when the conditional distribution within each cloud is uniform, then jumping between clouds in step 2 makes the marginal distribution on the clouds more uniform. Applying these bounds to (8) gives:

$$|M\alpha\cdot\alpha| \leq \lambda_1\cdot||\alpha^{\|}||^2 + 2\lambda_2\cdot||\alpha^{\|}||\cdot||\alpha^{\perp}|| + \lambda_2^2\cdot||\alpha^{\perp}||^2 \tag{9}$$

and by dividing both sides of the equation with $||\alpha||^2$, and taking $p = \frac{||\alpha^{\|}||}{||\alpha||}$ and $q = \frac{||\alpha^{\perp}||}{||\alpha||}$, and taking $p^2 + q^2 = 1$ gives the following:

$$\frac{|M\alpha\cdot\alpha|}{||\alpha^2||} \leq \lambda_1\cdot p^2 + 2\lambda_2\cdot pq + \lambda_2^2\cdot q^2 \leq \lambda_1 + \lambda_2 + \lambda_2^2 \tag{10}$$

This is the upper bound on the second-largest eigenvalue of $G_1 ⓩ G_2$. This upper bound can be taken to be less than one, as long as either $\lambda_1, \lambda_2 < 1$. The next point of the discussion is how the paper obtained an improved upper-bound on this second-largest eigenvalue. Take equation (7) and divide by $||\alpha^2||$ to express the following:

$$\frac{M\alpha\cdot\alpha}{||\alpha^2||} = \frac{\tilde{A}(\alpha^{\|} + \tilde{B}\alpha^{\perp})\cdot(\alpha^{\|} + \tilde{B}\alpha^{\perp})}{||\alpha^{\|} + \alpha^{\perp}||^2} \tag{11}$$

The claim given by the paper is that $\tilde{A}$ is a reflection through a linear subspace S of $\mathbb{R}^{N_1 D_1}$. Thus for any vector v, $\tilde{A}v\cdot v = (cos2\theta)\cdot||v||^2$, where θ is the angle between v and S. Take a moment to better detail the relationship between A and $\tilde{A}$. Let $e_v \in \mathbb{R}^{N_1}$ be the v^{th} standard basis vector, so that Ae_v is the uniform distribution over the neighbors of v. Ae_v can also be expressed as $Ae_v = C\tilde{A}(e_v \bigotimes u_{D_1})$. Here, u_{D_1} is the uniform distribution, C is a linear mapping from $\mathbb{R}^{N_1 D_1} \to \mathbb{R}^{D_1}$. For example, if π is the probability distribution over the vertices of $G_1 ⓩ G_2$, then $C\pi$ gives the marginal distribution for a set of N_1 clouds. The paper describes as the tensoring of e_v with u_{D_1} as taking the uniform distribution over the k^{th} edge of v, and C maps (w, l) to w, recall that $\text{Rot}_{G_1}(v, k) = (w, l)$. The relationship between A and $\tilde{A}$ works for all vectors $\beta \in \mathbb{R}^{N_1}$ giving $A\beta_v = C\tilde{A}(\beta_v \bigotimes u_{D_1})$. Returning to the discussion of equation (11) and applying the claim results in:

$$\frac{M\alpha\cdot\alpha}{||\alpha^2||} = |cos2\theta|\cdot\frac{||\alpha^2 + \tilde{B}\alpha^{\perp}||^2}{||\alpha^{\|} + \alpha^{\perp}||^2} = |cos2\theta|\cdot\frac{cos^2\phi}{cos^2\phi'} \tag{12}$$

where θ is the angle between $\alpha^{\|} + \tilde{B}\alpha^{\perp}$ and S, $\phi \in [0, \frac{\pi}{2}]$ is the angle between $\alpha^{\|}$ and α, and ϕ' is the angle between $\alpha^{\|}$, $\alpha^{\|} + \tilde{B}\alpha^{\perp}$, and ψ is the angle between $\alpha^{\|}$ and S. One seeks to maximize equation (12) subject to the following constrains:

1. ϕ, ϕ', and $\psi \in [0, \frac{\pi}{2}]$, where ψ is the angle between $\alpha^{\|}$ and S.
2. Since ϕ' is the angle between $\alpha^{\|}$ and $\alpha^{\|} + \tilde{B}\alpha^{\perp}$, and θ is between $\alpha^{\|} + \tilde{B}\alpha^{\perp}$ and S, then $\theta \in [\psi - \phi', \psi + \phi']$.
3. Take the first bound discussed for equation (8), so that $\frac{||\tilde{B}\alpha^{\perp}||}{||\alpha^{\perp}||} = \mu_2 \leq \lambda_2$.
4. Take the second bound discussed for (8), so that $|cos2\psi| = \mu_1 \leq \lambda_1$.

μ_1 and μ_2 are placeholders for simplicity, and these variables will be replaced in the final equation. Consider the case where $\phi' \leq \min(\psi, \frac{\pi}{2} - \psi)$, this gives:

$$|cos2\theta| = \max[|cos2(\psi + \phi')|, |cos2(\psi - \phi')|] = |cos2\psi \cdot cos2\phi'| + |sin2\psi \cdot sin2\phi'| \tag{13}$$

then by plugging (13) into (12), substituting the appropriate values with μ_1, μ_2, and performing trigonometric manipulations yields:

$$|cos2\theta| \cdot \frac{cos^2\phi}{cos^2\phi'} = \frac{1}{2}(1 - \mu_2^2)\mu_1 + \frac{1}{2}\sqrt{(1 + \mu_2^2)^2\mu_1^2 + 4\mu_2^2(1 - \mu_1^2)} \tag{14}$$

finally by substituting the appropriate bounds back into (14) gives that for G_1, a (N_1, D_1, λ_1) graph, and G_2, a (D_1, D_2, λ_2) graph, then $G_1 \textcircled{z} G_2$ is a $(N_1 D_1, D_2^2, f(_2, \lambda_2))$ graph, resulting in:

$$f(\lambda_1, \lambda_2) = \frac{1}{2}(1 - \lambda_2^2)\lambda_1 + \frac{1}{2}\sqrt{(1 + \lambda_2^2)^2\lambda_1^2 + 4\lambda_2^2(1 - \lambda_1^2)} \tag{15}$$

The paper describes the properties of (15) as follows:

1. $f(\lambda, 0) = f(,) = \lambda$ and $f(\lambda, 1) = f(,) = 1$ for $\lambda \in [0, 1]$.
2. if neither λ_1 or λ_2 is one, then this function is increasing for both.
3. if both $f(\lambda_1) < 1$ and $f(\lambda_2) < 1$, $f(\lambda_1, \lambda_2) < 1$.
4. $f(\lambda_1, \lambda_2) \leq \lambda_1 + \lambda_2$ for $\lambda \in [0, 1]$

There is also the case where $\phi' > \min(\psi, \frac{\pi}{2} - \psi)$, however the bound found for this case always less than or equal to (15), thus it is not as useful as (15) when establishing the eigenvalue bound of $G_1 \textcircled{z} G_2$.

4 Expander Construction

Now that insight on the zig-zag product has been provided, this section explains how the zig-zag product is used to iteratively build an expander. Start with a H as any $(D4, D, \frac{1}{5})$ graph, which is the building block of the construction. A simple example of the iteration is as follows:

1. $G_1 = H^2$.
2. for every G_i, find $G_{i+1} = G_i^2 \textcircled{Z} H$.

G_i is a $(N, D^2, \frac{2}{5})$ graph with $N_i = D^{4i}$. This iteration will be refined later, since currently, computing neighborhoods in G_i is in time poly(N_i) rather than poly(logN_i). An overview of this refinement is given as follows:

- To resolve the poly(N_i) computation of G_i, the iteration will include taking the tensor product of the adjacency matrix. This procedure will be described in detail later on.

- The current iteration yields a graph with a reasonably small degree[1]. However, the refinement will show how to change the iteration so that an infinite family of explicit expanders with $D = 4$.
- Instead of using exhaustive search to find the required H, two elementary explicit constructions will be used.
- The paper describes *Ramanujan* graphs as graphs whose second eigenvalue is $2\sqrt{D - \frac{1}{D}}$, these types of graphs will be used in the construction.
- The paper suggests using the *replacement* product which offers a simple way to yield a product. So far, G_2 size is the same as G_1 degree, and this simpler operation instead replaces every vertex in G_1 with a copy of G_2. This product has similar properties to the zig-zag product. A better overview of this variant to the zig-zag product will be given in section 5.

A mentioned in the overview, the construction time can go from poly(N_i) to poly(logN_i) by introducing tensoring to reduce the length of the iteration. Consider H to be a (D^8, D, λ) graph. The iteration is as follows:

1. $G_1 = H^2$.
2. for every G_t, find $G_t = (G_{\frac{t-1}{2}} \bigotimes G_{\frac{t-1}{2}})^2 \textcircled{z} H$

where G_t is a (D^{8t}, D^2, λ_t) graph, with $\lambda_t = \lambda + O(\lambda^2)$, and Rot_{G_1} can be found in time poly(t, log$D$) with poly(t) oracle queries to Rot_H. To guarantee that G_t are expanders, H eigenvalue needs to be bounded such that $\lambda_h \leq \frac{1}{5}$, forcing the degree of H and the expander family rather large. This is because there is a trade off between degree and eigenvalue, because one desires a large eigenvalue for better expansion, but a small degree is also desired. However, a small degree forces a small eigenvalue, thus the need to optimize this trade off . There is a technique to zig-zag G_t with a cycle C one can obtain a family of 4-degree expanders. In fact, this technique can be used to convert any family of odd-degree expanders into a family of 4-degree expanders. If G is an (N, D, λ) graph with odd D and $\lambda < 1$, then $G \textcircled{Z} C$ is an $(ND, 4, \lambda')$ where $\lambda' < 1$.

4.1 The Base Graph

To build the infinity family of expanders one requires a base graph H to start with. H is a (D^8, D, λ) with $\lambda \leq \frac{1}{5}$. Now the focus is to describe two known simple constructions for H, although one can perform exhaustive search. The first one is based off of the projective plane construction work by Alon [3]. The idea is to use the *affine* plane to make $N = D^2$, then using the zig-zag product to obtain a graph with $N = D^8$. Consider q to be a prime power of p such that $q = p^t$, and let $\mathbb{F}_q$ be a finite field of size q. Define a graph AP_q where the set of vertices is $\mathbb{F}_q^2$ and a set of edges $[(a, b), (c, d) : ac = b + d]$. This means that vertex (a, b) is connected to all points on a line $y = ax - b$. Denote this line as $L_{a,b}$. This results in AP_q as an $(q^2, q, \frac{1}{\sqrt{q}})$ graph, with a rotation map computable in poly(log$q$) given $\mathbb{F}_q$.

[1] by small degree this usually means $D \leq 1000$

The square of AP_q is almost a complete graph, meaning almost every pair of vertices is connected by a unique edge. This is because all pairs of lines in the $\mathbb{F}_q^2$ plane intersect. M is the adjacency matrix of AP_q and is of size q^2xq^2, and M^2 is calculated as follows:

$$M^2 = \frac{I_q \bigotimes qI_q + (J_q - I_q \bigotimes J_q)}{q^2} \tag{16}$$

The element in the (a, b) row and (a', b') column represents the number of common neighbors between (a, b) and (a', b'). These elements are divided by q^2. I_q is the q x q identity matrix, and J_q is a q x q matrix of constant 1s. When $a = a'$ then the lines $L_{a,b}$ and $L_{a',b'}$ have exactly one intersection. If $a = a'$ but $b \neq b'$ then their intersection is empty. Lastly, if both $a = a'$ and $b = b'$ then the intersection is of size q. The rotation map for AP_q is as follows:

$$Rot_q((a, b), t) = \begin{cases} ((\frac{t}{a}, t - b), t) & a \neq 0 \text{ and } t \neq 0 \\ ((t, -b), a) & a = 0 \text{ or } t = 0 \end{cases}$$

(17)

where $a, b, t \in \mathbb{F}_q$. Define the following graphs as:

$$AP_q^1 = AP_q \bigotimes AP_q \tag{18}$$

$$AP_q^{i+1} = AP_q^i \circledz AP_q \tag{19}$$

where AP_q^i is a $(q^{2(i+1)}, q^2, O(\frac{i}{\sqrt{q}}))$ graph, and $\text{Rot}_{AP_q^i}$ can be computed in poly(i, log$q$) given $\mathbb{F}_q$. The paper states that an $i = 7$ and large enough q gives a suitable H. A second construction for H relies on Cayley graphs derived from the generator matrix of an error-correcting code. The generator matrix rows form the basis for a linear code. Consider a prime power q and $d \in \mathbb{N}$. Define a graph $LD_{q,d}$ with a set of vertices $\mathbb{F}_q^{d+1}$, and $D = q^2$. A vector $a \in \mathbb{F}_q^{d+1}$, and $(x, y) \in \mathbb{F}_q$, then the $(x, y)^{\text{th}}$ neighbor of a is $a + (y, yx, yx^2, ..., yx^d)$. This gives that $LD_{q,d}$ is a $(q^{d+1}, q^2, \frac{d}{q})$ graph whose rotation map can be computed in time poly(logq, d) given $\mathbb{F}_{\text{ıı}}$. taking $d = 7$ and large enough q gives a decent H.

Let M be the normalized q^{d+1} x q^{d+1} adjacency matrix of $LD_{q,d}$. Let p be a characteristic of $\mathbb{F}_q$, $\zeta = e^{\frac{2\pi i}{p}}$ a primitive p^{th} root of 1. and L a map that takes $\mathbb{F}_q \to \mathbb{F}_p$. L is the identity map when $p = q$. Take a function $\chi_a(b) = \zeta^{L(\Sigma a_i b_i)}$. Here, $\chi_a(b + c) = \chi_a(b)\chi_a(c)$, where $b, c \in \mathbb{F}^{d+1}$. If a is a sequence so that $a = (a_{0,...,a_d}) \in \mathbb{F}^{d+1}$ if it can be show that for any instance of a, χ_a is an eigenvector of M, then all a are eigenvectors of M. This is shown as follows:

$$(M_{\chi_a})(b) = \frac{1}{q^2} \sum_{c \in \mathbb{F}^{d+1}} M_{bc} \cdot \chi_a = \frac{1}{q^2} \sum_{x,y \in \mathbb{F}_q} \chi_a(b + (y, yx, ..., yx^d)) \tag{20}$$

$$(M_{\chi_a})(b) = (\frac{\sum_{x,y \in \mathbb{F}_q}}{\chi_a} (y, yx, ..., yx^d)q^2) \cdot \chi_a(b) = \lambda_a \cdot \chi_a(b) \tag{21}$$

Here, χ_a is an eigenvector of M with eigenvalue λ_a. The following equation shows that $|\lambda_a| \leq \frac{d}{q}$ for all but one $a \in \mathbb{F}^{d+1}$:

$$\lambda_a = \frac{1}{q^2} \sum_{x,y\mathbb{F}_q} \chi_a((y, yx, ..., yx^d)) = \frac{1}{q^2} \sum_{x,y\in\mathbb{F}_q} \zeta^{L(yp_a(x))} \tag{22}$$

where $p_a(x)$ is the polynomial $a_0 + a_1x + ... + a_d(x)^d$. If x is a root of p_a, then $\zeta^{L(yp_a)} = 1$ for every y allowing x to contribute $\frac{q}{q^2} = \frac{1}{q}$ to λ. If it is not the case that x is a root of $p_a(x)$, then $yp_a(x)$ takes on every value of ${}_{Fq}$ as y changes. This gives that $\zeta^{L(yp_a(x))}$ varies uniformly for every p^{th} of 1. Because the sum of all these p^{th} is 0, none of the x's contribute to λ_a. If $p \neq 0$, then P_a has at most d roots, this gives $|\lambda_a| \leq \frac{d}{q}$.

5 Zig-Zag Variants and Final Remarks

The paper presents two variations to the zig-zag product, however this report mostly focuses on the original zig-zag product. This section provides a brief overview of these variations. The first variation seeks to achieve a better relationship between the degree and expansion of the product graph which uses a more efficient use of the expansion of the smaller graph. Instead of three steps, this variation has two "zig" and two "zag" moves, where the second "zig" and the first "zag" use the same random bits. By reusing bits, the degree is decreased. If G_1 is an (N_1, D_1, λ_1) graph, and G_2 is an (N_2, D_2, λ_2) graph, then $G_1 \textcircled{z'} G_2$ is a $(N_1D_1, D_2^3, \lambda_1 + 2\lambda_2^2)$ graph, and $\text{Rot}_{G_1 \textcircled{z'} G_2}$ is computable in poly($\log N_1$, $\log D_1$, $\log D_2$) with one oracle query to Rot_{G_1} and $D_2 + 2$ oracle queries to Rot_{G_2}. The second-largest value obtained is $O(\frac{1}{D^{\frac{1}{3}}})$

The second variant aims to simplify the product operation, but the eigenvalue-degree tradeoff is deteriorated. By taking the product of two expanders, the result is a large expander whose degree depends on the smaller graph. The idea is to place a copy of G_2 around every vertex of G_1 while maintaining both graph's edges. Every vertex is connected to all of its original neighbors in the cloud, along with one vertex in the neighboring cloud. An example takes G_1 as a n-dimension cube graph, and G_2 is the cycle on n vertices, then the result is a graph termed *cube connected cycle*. Basically, if you take a undirected cube graph, each vertex is replaced by a cycle. This architecture was popular for parallel computers. If G_1 is an (N_1, D_1, λ_1) graph, and G_2 is an (N_2, D_2, λ_2) graph, then $G_1 \textcircled{r} G_2$ is a $(N_1D_1, D_2+1, g(\lambda-1, \lambda_2, D_2))$ graph, where $g(\lambda_1, \lambda_2, D_2) \leq (p+(1-p)f(\lambda_1, \lambda_2))^{\frac{1}{3}}$, $p = \frac{D_2^2}{(D_2+1)^3}$, and $f(\lambda_1, \lambda_2)$ is as given in section 3.1.

This Report shows how the zig-zag product is defined, and how it is used to build expanders that inherit the expansion properties of the parents. This is important because many applications in computer science require such explicit constructions, and previous algebraic constructions are hindered by sophistic evaluations of the eigenvalues. This paper gives a combinatorical way to build these graphs, along with a simpler and more intuitive evaluation of the eigenvalues.

References

[1] O. Reingold, S. Vadhan, A. Wigderson. Entropy waves, the zig-zag graph product, and new constant-degree expanders. In *Annals of Mathematics*, 2002.

[2] M. Ajtai. Recursive construction for 3-regular expanders. In *Combinatorica*, 1994.

[3] N. Alon. Eigenvalues, geometric expanders, sorting in rounds, and Ramsey theory. In *Combinatorica*, 1986.