Polytopes with Everywhere Positive Curvature

By

JENNA RASHKOVSKY

SENIOR THESIS

Submitted in partial satisfaction of the requirements for Highest Honors for the degree of

BACHELOR OF SCIENCE

in

APPLIED MATHEMATICS

in the

COLLEGE OF LETTERS AND SCIENCE

of the

UNIVERSITY OF CALIFORNIA,

DAVIS

Approved:

Jesús A. De Loera

March 2025

ABSTRACT

This senior thesis investigates polytopes with positive notions of curvature in 3-dimensional space. In this thesis, we introduce two notions of curvature: Forman Ricci curvature and Effective Resistance curvature. We discuss the methods for calculating both and provide examples. Additionally, we explain the processes we undertook to gather a diverse database of polytopes. Finally, we present our findings of polytopes, identifying the ones that exhibit everywhere positive curvature for both notions.

ACKNOWLEDGMENTS.

First, I would like to thank Professor Jesús De Loera for giving me the opportunity to write this senior thesis with him.

Moreover, I would like to offer my special thanks to Jillian Eddy, a Ph.D student in the Department of Mathematics at UC Davis, for her generous support and help throughout the project.

Lastly, I would like to thank the National Science Foundation for support provided via the NSF grant 2348578 for Professor De Loera.

Contents

CHAPTER 1

Introduction: Discrete Notions of Curvature

Discrete curvature notions are designed to reflect the properties of different notions of curvature for Riemannian manifolds. Similar to the results of Bonnet-Myers [**8**], discrete curvatures can be used to bound the diameter of a polytope. The diameter is the maximum distance between any two vertices of a polytope, where distance is the minimum number of edges required to travel from one vertex to another. This is useful for bounding the computational cost of search algorithms that operate on the polytope. However, just as in the continuous setting, current diameter bounds using notions of discrete curvature require that the curvature of the polytope is everywhere positive. Thus, we are curious about characterizing polytopes which have everywhere positive discrete curvature under various curvature definitions.

1.1. Preliminary Definitions and Lemmas

In order to provide a solid foundation for our discussion, we begin by introducing some key definitions and lemmas that will guide and inform our work. These preliminary concepts are essential for understanding the structure and properties of the objects we will study. To start, we consider *polytopes*, which are higher-dimensional abstractions of polygons. While polytopes can be quite complex, in this paper we focus specifically on combinatorial polytopes. Our primary interest lies not in the precise geometric details, such as the specific positions of vertices or hyperplanes, but rather in their combinatorial geometry.

A particular type of polytope we explore is the *simplicial polytope*, which is defined by the property that all its faces are simplices. In the case of 3-dimensional polytopes, this means that all the faces are triangles. Furthermore, we introduce the concept of a *simple polytope*. A d-dimensional polytope is considered simple if each of its vertices is adjacent to exactly d edges. For 3-dimensional polytopes, this implies that each vertex is connected to exactly three edges. Finally, we discuss the *dual* of a polytope, which is formed by swapping the roles of the polytope's vertices and facets: the facets become vertices, and the vertices become facets.

Next, we will state two lemmas that will play key roles in our paper.

LEMMA 1 (Derived from Euler's Formula [**6**]). *Let P be a 3-dimensional polytope and let $F = p_3 + p_4 + \cdots$ be the number of faces of P, where p_i is the number of i-sided faces* *in P. Then,*

$$\sum_{i\geq 3}(6-i)p_i \geq 12.$$

If P is simple, then

$$\sum_{i\geq 3}(6-i)p_i = 12.$$

PROOF. By Euler's formula for polyhedra [**6**], $V - E + F = 2$, where V is the number of vertices, E is the number of edges, and F is the number of faces. For the polytope P, each i-sided face contributes i edges, counted twice.

Since each edge belongs to two faces, $\sum_{i\geq 3} ip_i = 2E$.

Now, let's substitute $F = \sum_{i\geq 3} p_i$ and use $2E = \sum_{i\geq 3} ip_i$ and $V - E + F = 2$.

Then we consider the inequality:

$$\sum_{i\geq 3}(6-i)p_i = 6F - \sum_{i\geq 3} ip_i.$$

Rewriting using E:

$$\sum_{i\geq 3}(6-i)p_i = 6F - 2E.$$

We can also rearrange the equation $V - E + F = 2$, to get $E = V + F - 2$. Using this, we get:

$$\sum_{i\geq 3}(6-i)p_i = 6F - 2(V + F - 2) = 6F - 2V - 2F + 4 = 4F - 2V + 4.$$

Given $E \geq 3V - 6$ for 3-polyhedra:

$$V - 2E + 4F \geq 12.$$

And, in particular, if P is simple, $2E = 3V$ so

$$V - 2E + 4F = 12.$$

□

LEMMA 2 (Degree-Diameter Bound). [**7**] *Let $G = (V, E)$ be a graph with maximum vertex degree k and diameter d. Then, the number of vertices $|V|$ is bounded by*

$$|V| \leq 1 + k\sum_{i=0}^{d-1}(k-1)^i.$$

We also include the diameter bounds provided by these notions of curvature. Note that their hypotheses motivate our exploration of the space of everywhere-positive polytopes.

THEOREM 1 (Forman, 2003). *Let P be a combinatorial polytope. Suppose there exists $c > 0$ such that for all edges $e \in P$, the Forman Ricci curvature $\mathcal{F}(e) \geq c$. Then,*

$$diam(P) \leq \frac{2}{c}\left(\max_{e\in P} \#\{f^{(2)} > e\} + 1\right).$$

THEOREM 2 ([**2**]). *Let $G = (V, E)$ be a simple graph with vertices indexed by $1, \ldots, |V|$). Suppose that for each vertex $i \in G$ the Effective Resistance curvature $\kappa_i \geq c > 0$ for some constant c. Then,*

$$diam(P) \leq \lceil \frac{2}{c} \log |V| \rceil.$$

1.2. Forman Ricci Curvature

The first notion of curvature that we looked at is Forman Ricci curvature. Forman Ricci curvature is a combinatorial form of discrete curvature assigned to the edges (or 1-dimensional cells) of regular, quasiconvex CW complexes. Specifically, polytopes meet the conditions needed for this type of curvature. Introducing the notation of Forman [**4**] in our context, let P be a convex combinatorial polytope. We use the notation $f^{(2)}$ to refer a 2-dimensional face of P, and in general, the subscript $^{(i)}$ to indicate the dimension of the attributed face. Additionally, if $f^{(2)}$ is part of a 2-dimensional face, we write $e < f^{(2)}$ to indicate that an edge $e \in P$ is contained in the closure $\overline{f^{(2)}}$ of that 2-dimensional face.

DEFINITION 1. Let P be a polytope and e, e' edges of P. There are two ways in which e and e' are considered *parallel neighbors*:

i) 0-neighbors. There exists $v^{(0)} \in P$ such that $v^{(0)} < \overline{e}$ and $v^{(0)} < \overline{e'}$. There does *not* exist $f^{(2)} \in P$ such that $\overline{f^{(2)}} > e$ and $\overline{f^{(2)}} > e'$. This means that two edges share a vertex but do not belong to the same facet of the polytope.

ii) 2-neighbors. There exists $f^{(2)} \in P$ such that $\overline{f^{(2)}} > e$ and $\overline{f^{(2)}} > e'$. There does *not* exist $v^{(0)} \in P$ such that $v^{(0)} < \overline{e}$ and $v^{(0)} < \overline{e'}$. This means that two edges belong to the same facet of the polytope but do not share a vertex.

DEFINITION 2. Let P be a combinatorial polytope. The *Forman Ricci Curvature* of an edge $e \in P$ is given by

$$\mathcal{F}(e) = \#\{f^{(2)} > e\} + \#\{v^{(0)} < e\} - \#\{\text{parallel neighbors of } e\}.$$

In simpler terms, the Forman Ricci Curvature of an edge is the number of 2-dimensional faces that the edge is part of plus the number of vertices that make up the edge (typically 2) minus the number of parallel neighbors of the edge as described in Definition 1.

We will provide an example calculating the Forman Ricci curvature of an edge using Figure 1 below.

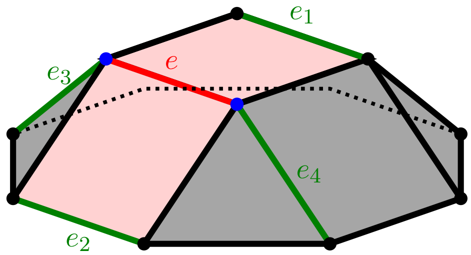


FIGURE 1. 3-dimensional polytope "Square Cupola" .

Within the example in Figure 1, we will show how to calculate the Forman Ricci curvature of the red edge labeled e. In order to do this, we must use Definition 2.

First, $\#\{f^{(2)} > e\}$ is the number of faces that the edge is part of. This can be seen by the two pink quadrilaterals, as edge e is part of 2 faces. This means $\#\{f^{(2)} > e\} = 2$.

Next, $\#\{v^{(0)} < e\}$ is the number of vertices that make up the edge. This can be seen by the two blue vertices, as the edge, e, is made up of them. Thus, $\#\{v^{(0)} < e\} = 2$.

Finally, $\#\{\text{parallel neighbors of } e\}$ are the parallel neighbors, which can be seen by the dark green edges. Edges e_3 and e_4 are the 0-neighbors of e, as they share a vertex with e but not a facet. Edges e_1 and e_2 are the 2-neighbors of e, as they share a face with e but not a vertex. Thus, this means $\#\{\text{parallel neighbors of } e\} = 4$.

Putting everything together, we get that

$$\mathcal{F}(e) = 2 + 2 - 4 = 0.$$

As a result, the red edge, e, has a Forman Ricci curvature of 0.

Using this process, we computed the Forman Ricci curvatures for each edge in the graph. The results are presented in Figure 2 below. In the figure, blue edges indicate negative curvature, red edges indicate positive curvature, and black edges indicate zero curvature.

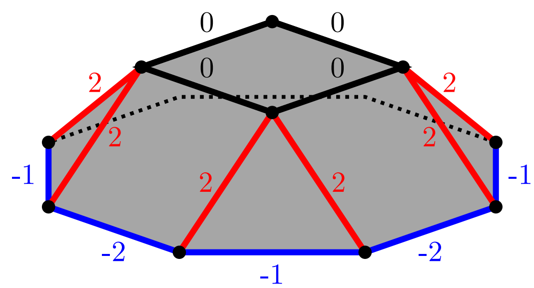


FIGURE 2. 3-dimensional polytope "Square Cupola" with edges labeled according to their Forman Ricci curvature.

1.2.1. Theoretical results. In order to determine which 3-polytopes have everywhere positive Forman Ricci curvature, we first observe some conditions that such polytopes must meet.

We first notice a special relationship between an everywhere positive 3-polytope and its dual polytope.

THEOREM 3. *A combinatorial 3-polytope P has everywhere positive Forman curvature if and only if its dual P^{Δ} has everywhere positive Forman curvature.*

PROOF. Since P is a 3-polytope, every edge $e \in P$ corresponds to a dual edge $e^{\Delta} \in P^{\Delta}$. When we take the dual, every vertex contained in e becomes a 2-face containing e^{Δ}. At the same time, every 2-face containing $e \in P$ is dualized to a vertex $v \in P^{\Delta}$ contained in e^{Δ}.

In this way, the set of 0-neighbors of e is dualized to the set of 2-neighbors of e^{Δ}, and the set of 2-neighbors of e is dualized to the set of 0-neighbors of e^{Δ}. □

By considering which attributes *cannot* appear in a 3-polytope with everywhere positive Forman curvature, we begin to characterize exactly which polytopes are everywhere positive.

THEOREM 4. *Let P be a 3-dimensional combinatorial polytope. If at least one of the following is true, then P does not have everywhere positive Forman Ricci curvature.*

i) *There exists a 2-dimensional face $f^{(2)} \in P$ such that $\#\{e^{(1)} < \overline{f^{(2)}}\} > 6$ (i.e. the number of sides of $f^{(2)}$ is more than 6).*
ii) *There exists a 0-dimensional vertex $v^{(0)} \in P$ such that $\#\{\overline{e^{(1)}} > v^{(0)}\} > 6$ (i.e. the degree of $v^{(0)}$ is more than 6).*

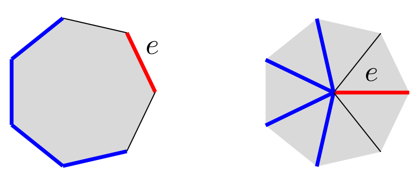


FIGURE 3. Parallel neighbors (blue) of an edge e (red) in the case where e is an edge of a heptagon (left), and in the case where e is adjacent to a vertex of degree 7 (right).

PROOF. Suppose the case of (i). Let f be a 2-dimensional face of P containing $k > 6$ edges, and let $e^{(1)} < f$ (i.e. e is one of the k sides of f). We aim to compute an upper bound on the Forman curvature of e. Note that e must contain 2 vertices and must be contained in two 2-dimensional faces, one of which being f. Since f is a k-gon with $k > 6$, the number of 2-neighbors to e with respect to f is $k - 3$. The total number of parallel neighbors of e then satisfies

$$\#\{\text{parallel neighbors of } e\} \geq k - 3.$$

So, computing the Forman curvature of e,

$$\mathcal{F}(e) = 2 + 2 - \#\{\text{parallel neighbors of } e\} \leq 7 - k \leq 0.$$

Thus there is a 1-dimensional face of P with non-positive Forman curvature.

Now suppose the case of (ii). Let v be a 0-dimensional face of P that is contained in $d > 6$ 1-dimensional faces (i.e. v is a vertex of degree $d > 6$). Let $e' > v$ be one of these 1-dimensional faces. Since e' is contained in two 2-dimensional faces, the number of 0-neighbors to e' with respect to v is $d - 3$. So,

$$\#\{\text{parallel neighbors of } e\} \geq d - 3.$$

Computing the Forman curvature of e,

$$\mathcal{F}(e) = 2 + 2 - \#\{\text{parallel neighbors of } e\} \leq 7 - d \leq 0.$$

Thus there is a 1-dimensional face of P with non-positive Forman curvature. □

These observations allow us to utilize graph-theoretical bounds on the number of vertices of a graph.

THEOREM 5. *There are finitely many 3-dimensional combinatorial polytopes with everywhere positive Forman Ricci curvature.*

PROOF. Let P be a 3-dimensional combinatorial polytope with everywhere positive Forman Ricci curvature and graph $G = (V, E)$. We know that the maximum diameter of P is 6. By Theorem 4, we know that the maximum degree of a vertex is 6. Then, computing the degree-diameter bound of P, we have

$$|V| \leq 1 + 6\sum_{i=0}^{5} 5^i = 23437$$

Thus with the number of vertices of P bounded from above, there are finitely many 3-dimensional polytopes (of combinatorial type) that have everywhere positive Forman Ricci curvature. □

Focusing specifically on simple 3-dimensional polytopes, we can further bound the number of vertices.

THEOREM 6. *Let P be a 3-dimensional simple polytope whose edges have been assigned uniform weight of 1. Suppose for all edges $e \in P$ we have $\mathcal{F}(e) > 0$. Then, P has at most 20 vertices.*

PROOF. Let n be the number of vertices of P, E the number of edges, and F the number of faces, where

$$F = F_3 + F_4 + \cdots$$

represents the number of faces, F_i being the number of i-sided faces.

Then, we can write Euler's formula as:

$$V - E + F = 2$$
$$n - E + F_3 + F_4 + F_5 + F_6 + \cdots = 2.$$

However, we know by Theorem 4 that, since P has everywhere positive Forman Ricci curvature, $F_i = 0$ for all $i \geq 7$. Additionally, if P contains a hexagon, it must be combinatorially equivalent to the hexagonal pyramid. Thus, if P contains a hexagon, P has 7 vertices. Otherwise, we move on with the following identity:

$$n - E + F_3 + F_4 + F_5 = 2. \tag{1.1}$$

Since $\deg(v) = 3$ for all vertices $v \in P$, we have

$$2E = \sum_{i=1}^{n} \deg(v_i) = 3n.$$

So, we can further substitute into 1.1 to get

$$n - \frac{3}{2}n + F_3 + F_4 + F_5 = 2.$$

$$\frac{n}{2} = F_3 + F_4 + F_5 - 2 \tag{1.2}$$

Now, by consequence of Euler's theorem, P is trivalent, so we have the following equality:

$$\sum (6 - i)F_i = 3F_3 + 2F_4 + F_5 = 12$$
$$\implies 12 - 3F_3 - 2F_4 - F_5 = 0.$$

So, we add 0 to formula 1.2 to achieve

$$\frac{n}{2} = F_3 + F_4 + F_5 - 2 + (12 - 3F_3 - 2F_4 - F_5)$$
$$\frac{n}{2} = 10 - 2F_3 - F_4$$
$$\leq 10,$$

giving the inequality $n \leq 20$. □

PROPOSITION 1 (Characterization). *Let P be a 3-dimensional polytope with everywhere positive Forman Ricci curvature. Let K and L be adjacent 2-dimensional faces of P, with k and ℓ sides respectively. Then,*

$$(k, \ell) \in \{(3, 3), (3, 4), (3, 5), (3, 6), (4, 4), (4, 5)\}.$$

PROOF. Let $e = \bar{K} \cap \bar{L}$. Then,

$$\mathcal{F}(e) = 2 + 2 - ((k - 3) + (\ell - 3)) - \#\{0\text{-neighbors of } e\} \leq 10 - (k + \ell),$$

which achieves equality when both vertices of e are degree 3.

Then, if $\mathcal{F}(e) > 0$ we must have $k + \ell < 10$. □

These results allow us to exhibit all everywhere-positive 3-polytopes under the Forman Ricci notion of curvature, see Section 2.3.1.

1.3. Resistance Curvature

The next notion of curvature that we will discuss in this paper is Effective Resistance curvature. This curvature is assigned to the vertices of the polytope rather than the edges [**3**]. Resistance curvature utilizes linear algebra and requires solving a system of linear equations.

We will provide some definitions of matrices involved in finding the Effective Resistance curvature.

Definition 3. The *adjacency matrix* of a graph is an $n \times n$ matrix where n is the number of vertices of the polytope [**5**]. Since we are looking at unweighted graphs, if $i \neq j$, then (i,j)-entry of the adjacency matrix is:

$$a_{ij} = \begin{cases} 0, & \text{if i and j do not share an edge,} \\ 1, & \text{if i and j share an edge.} \end{cases}$$

For the purpose of this paper, since we are looking at graphs without loops, when $i = j$, $a_{ij} = 0$. It is also important to note that since we will be observing undirected graphs, the adjacency matrices will be symmetric.

Definition 4. The *Laplacian matrix* of a graph, L, is defined by:

$$L = D - A$$

where D is the diagonal matrix of vertex degrees and A is the adjacency matrix of the graph [**5**].

Definition 5. Let G $= (V, E)$ be a graph whose vertices are indexed by $v_1, \ldots, v_n$. The *effective resistance curvature matrix*, Ω, of G has (i,j)-entries:

$$\omega_{ij} \coloneqq (\boldsymbol{e}_i - \boldsymbol{e}_j)^T L^+ (\boldsymbol{e}_i - \boldsymbol{e}_j)$$

where L^+ is the pseudoinverse of the Laplacian matrix of G and $\boldsymbol{e}_i$ is the ith unit vector.

1.3.1. Construction of Effective Resistance Matrix. We will walk through an example of constructing the effective resistance matrix. We will find the matrix for the polytope shown in Figure 4 below.

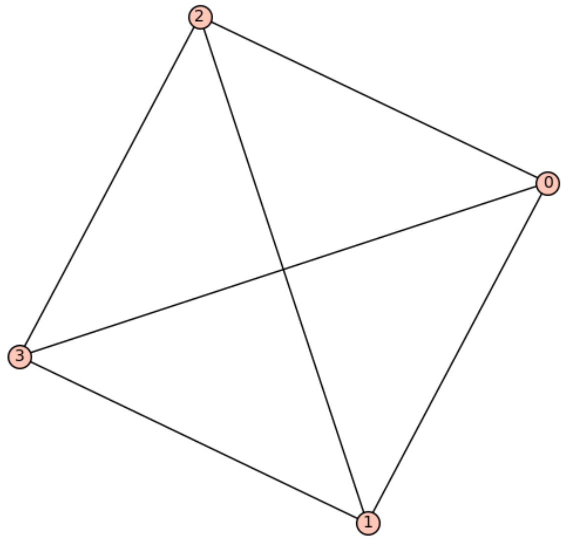


FIGURE 4. A simple polytope with four vertices each with degree 3.

First, we need to find the Laplacian matrix for this polytope. We will do this using Definition 4. Since each vertex has degree 3, this means that D has 3 on the diagonal entries. We also know that since each vertex shares an edge with every vertex except itself, the adjacency matrix, A, has a 1 everywhere except the diagonal. Thus, we get

$$L = \begin{bmatrix} 3 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 3 \end{bmatrix} - \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 3 & -1 & -1 & -1 \\ -1 & 3 & -1 & -1 \\ -1 & -1 & 3 & -1 \\ -1 & -1 & -1 & 3 \end{bmatrix}.$$

Next, in order to be able to invert the Laplacian matrix, L, we must perturb the matrix. We can do this by adding a small number to each entry of the matrix. We will call the perturbation matrix Γ.

$$\Gamma = L + \frac{1}{\text{\# of vertices}}$$

For our example, since there are 4 vertices, we would add $\frac{1}{4}$ to each entry of L, leaving us with

$$\Gamma = \begin{bmatrix} 3.25 & -0.75 & -0.75 & -0.75 \\ -0.75 & 3.25 & -0.75 & -0.75 \\ -0.75 & -0.75 & 3.25 & -0.75 \\ -0.75 & -0.75 & -0.75 & 3.25 \end{bmatrix}.$$

From this, we can compute the inverse using Python.

$$\Gamma^{-1} = \begin{bmatrix} 0.4375 & 0.1875 & 0.1875 & 0.1875 \\ 0.1875 & 0.4375 & 0.1875 & 0.1875 \\ 0.1875 & 0.1875 & 0.4375 & 0.1875 \\ 0.1875 & 0.1875 & 0.1875 & 0.4375 \end{bmatrix}$$

We can now find the effective resistance matrix by using the modified equation below.

$$\Omega_{ij} = \Gamma^{-1}_{ii} + \Gamma^{-1}_{jj} - 2\Gamma^{-1}_{ij} \tag{1.3}$$

Plugging everything into Equation 1.3 will give us our effective resistance matrix for the polytope in Figure 4,

$$\Omega = \begin{bmatrix} 0 & 0.5 & 0.5 & 0.5 \\ 0.5 & 0 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0 & 0.5 \\ 0.5 & 0.5 & 0.5 & 0 \end{bmatrix}.$$

1.3.2. Solving for Resistance Curvature. After finding the effective resistance curvature matrix, the next step is to solve for the effective resistance curvature of each vertex in the polytope.

DEFINITION 6. Let G $= (V, E)$ be a graph whose vertices are indexed by $v_1, \ldots, v_n$. The *effective resistance curvature* of a vertex v_i is the ith entry of the unique solution vector κ satisfying $\Omega\kappa = \mathbf{1}$.

Using Definition 6 on our example, we can set up our equation as

$$\begin{bmatrix} 0 & 0.5 & 0.5 & 0.5 \\ 0.5 & 0 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0 & 0.5 \\ 0.5 & 0.5 & 0.5 & 0 \end{bmatrix} \begin{bmatrix} \kappa_1 \\ \kappa_2 \\ \kappa_3 \\ \kappa_4 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}.$$

This therefore allows us to solve for the effective resistance curvature for each vertex. With our example, we have

$$\begin{bmatrix} \kappa_1 \\ \kappa_2 \\ \kappa_3 \\ \kappa_4 \end{bmatrix} = \begin{bmatrix} 0.666667 \\ 0.666667 \\ 0.666667 \\ 0.666667 \end{bmatrix}.$$

As a result this allows us to see that the polytope shown in Figure 4 has everywhere positive effective resistance curvature. We can also see that each vertex has the same resistance curvature.

1.4. Examples

In this section, we will show three different polytopes along with their Forman Ricci curvatures and effective resistance curvatures. First, we will show a polytope that has everywhere positive Forman Ricci curvature but does not have everywhere positive effective resistance curvature.

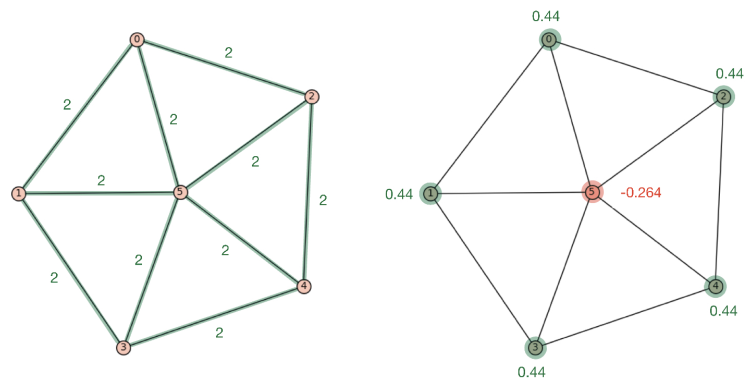


FIGURE 5. Example with everywhere positive Forman Ricci Curvature but not resistance curvature. The left shows the polytope's Forman Ricci Curvature while the right shows the polytope's effective resistance curvature.

The polytope in Figure 5 has everywhere positive Forman Ricci curvature (on the left), as each edge has a curvature of 2. The effective resistance curvature (on the right), however, is negative at one vertex.

Next, we will show a polytope that does not have everywhere positive Forman Ricci curvature but has everywhere positive effective resistance curvature.

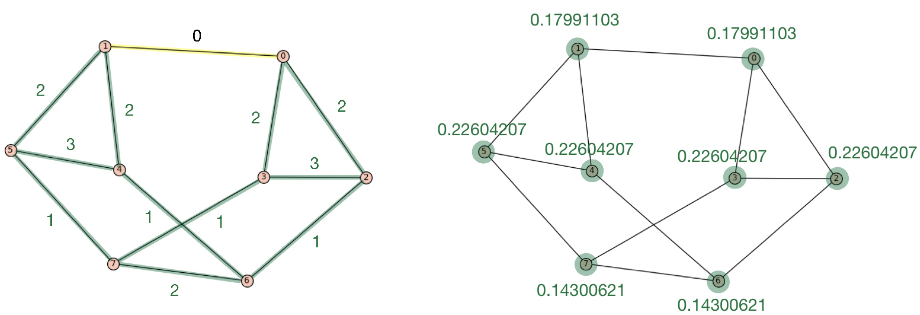


FIGURE 6. Example with everywhere positive effective resistance curvature but not Forman Ricci curvature. The left shows the polytope's Forman Ricci curvature while the right shows the polytope's effective resistance curvature.

The polytope in Figure 6 does not have everywhere positive Forman Ricci curvature since one edge has a curvature of 0. The effective resistance curvature, however, is positive for each vertex of this polytope.

Lastly, we will show a polytope that has everywhere positive Forman Ricci curvature and effective resistance curvature.

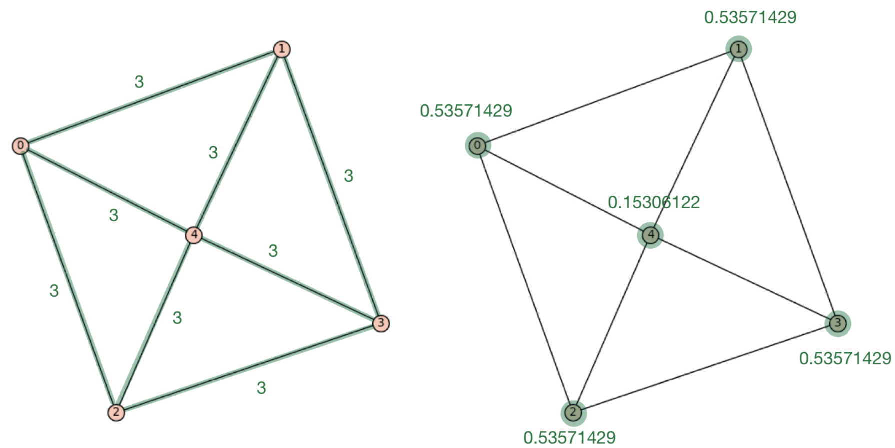


FIGURE 7. Example with everywhere positive Forman Ricci curvature and effective resistance curvature. The left shows the polytope's Forman Ricci curvature while the right shows the polytope's effective resistance curvature.

The polytope in Figure 7 has everywhere positive Forman Ricci curvature and effective resistance curvature. For the Forman Ricci curvature, each edge has the same calculation of 3. For the effective resistance curvature, each point has a positive calculation.

CHAPTER 2

Database of Polytopes

In this chapter, we will outline the various methods we used to create a database of polytopes. We will also discuss the dataset we gathered for each polytope. We will then present the results of our analysis, focusing on which polytopes had everywhere positive Forman Ricci curvature and effective resistance curvature.

2.1. Generation Process

To generate a diverse set of polytopes, we used five distinct methods. Each method provided unique advantages for our dataset. The following sections outline these methods in detail.

2.1.1. House of Graphs. Our initial dataset came from a database, *House of Graphs* [**1**]. By Steinitz' Theorem, graphs formed by the edges and vertices of 3-dimensional convex polyhedra are exactly 3-connected, planar graphs [**9**]. As a result, when we searched for graphs in the database, we filtered the search by looking for only planar graphs with a vertex connectivity of 3. This allowed us to gather 2438 graphs.

2.1.2. Random Matrices. The next method we used to generate polytopes for our dataset involved randomly generating matrices using Python in SageMath, a mathematics software system. We randomly constructed an $n \times 3$ matrix of integers, where n represented the number of vertices of the polytope and was randomly selected from a range, typically between 4 and 100. The reason we set the number of columns in the matrix to be 3 is because the polytopes we wanted to generate were in 3-dimensional space. As a result, each row of the matrix corresponded to a point in 3D space. After generating the matrix, we then used the Polyhedron command in SageMath to create a polytope with the vertices represented by the rows of the randomly generated matrix. Below, we will provide an example to illustrate the process.

As an illustration, here is a random matrix with 6 vertices that we generated using this method:

$$\begin{bmatrix} 6 & -1 & -1 \\ -2 & 0 & 3 \\ -1 & -1 & 41 \\ 157 & 5 & -1 \\ 1 & -1 & 1 \\ 1 & -17 & -7 \end{bmatrix}$$

With this matrix, we would then create a polytope that has vertices (6, -1, -1), (-2, 0, 3), (-1, -1, 41), (157, 5, -1), (1, -1, 1), and (1, -17, -7). By applying the Polyhedron function in SageMath, the resulting polytope is shown below, with both its 2D and 3D representations.

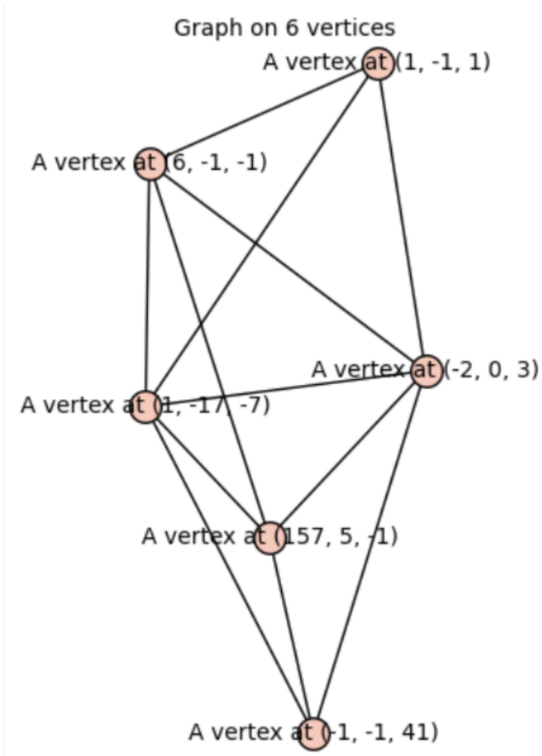


FIGURE 1. The sample polytope graph using the method of randomly generating a matrix.

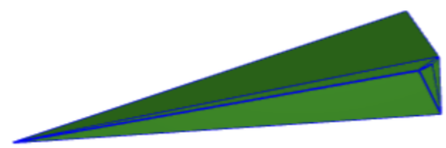


FIGURE 2. The 3D representation of the poltyope graph corresponding to the randomly generated matrix.

Figures 1 and 2 above allow us to see how a randomly generated matrix can be transformed into a polytope. Using this method, we were able to generate thousands of random polytopes.

However, one limitation of this approach is that it only generates simplicial polytopes. Since we are looking at 3-dimensional polytopes, this means that these polytopes are made up of only triangular faces. While this was useful for gathering a large amount of polytopes, it did not provide the level of diversity needed for a more well-rounded dataset. To address this limitation and enhance the variety of our dataset, we employed additional methods, which we will describe in the following sections.

2.1.3. Dual Graphs. The next step in our approach was to take each of the randomly generated polytopes and calculate their duals. As mentioned previously, the dual of a polytope is constructed by creating a vertex for each face of the original polytope. An edge is then placed between two vertices if their corresponding faces share an edge in the original polytope. This transformation allows us to generate new polytopes with a different combinatorial structure, providing a more diverse set of data. In order to generate the duals, we used a built-in dual function in SageMath, which computes the dual polytope efficiently. We will show an example of a dual polytope that we constructed below.

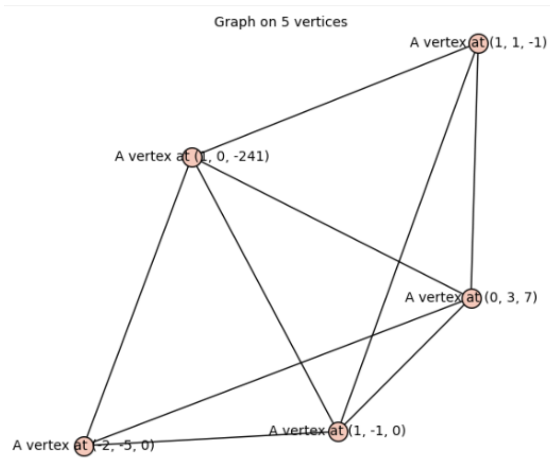


FIGURE 3. The original polytope randomly generated.

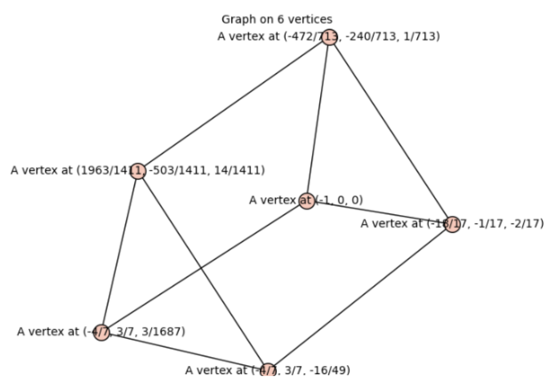


FIGURE 4. The dual of the randomly generated polytope on the left.

Figure 3 is a randomly generated polytope using the random matrix method explained in the previous section. Figure 4 illustrates the dual of the polytope shown in Figure 3. Using this approach, we generated the dual polytope for each of the randomly created simplicial polytopes. It is important to note that when we take the dual of a simplicial polytope, the result is a simple polytope.

By applying this process, we were able to generate new polytopes with a more diverse combinatorial structure. However, to achieve even more variety in our dataset, we needed to explore additional methods that would allow us to generate non-simplicial and non-simple polytopes, further expanding the range of our data.

2.1.4. Pyramiding. Another approach we used to generate polytopes for our dataset was pyramiding. This method involves taking previously generated polytopes from all the methods described earlier, and basically adding a pyramid to them through the addition of a new vertex.

The pyramiding process begins by selecting a polytope and identifying one of its faces, which can be a triangle, quadrilateral, pentagon, or any shape. The vertices that form this identified face are then tracked. Next, a new vertex is added to the polytope, and an edge is added to connect the new vertex to each of the vertices that make up the chosen face. We were able to write Python code to complete this process, enabling us to efficiently generate new polytopes with pyramiding. We will show an example using this method below.

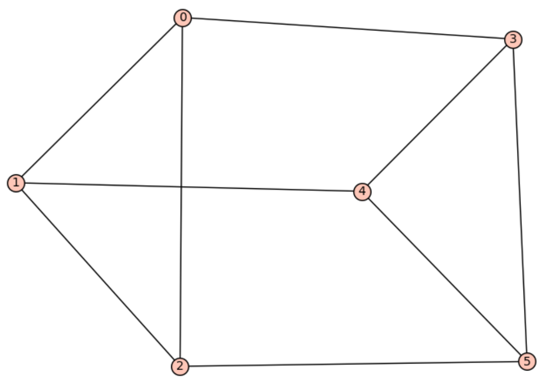


FIGURE 5. Original polytope from previous data.

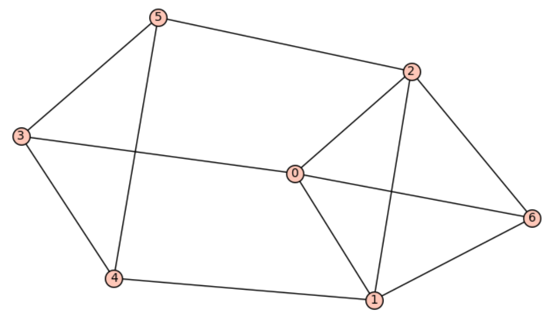


FIGURE 6. Polytope constructed using the pyramid method on the polytope to the left.

Figure 5 shows the original polytope, which was part of our initial dataset. In Figure 6, you can see the result of applying the pyramiding method. In this case, the chosen face was a triangle made up of vertices 0, 1, and 2. A new vertex, labeled 6, was then added to the polytope. Edges were created from the new vertex to the existing vertices 0, 1, and 2, resulting in three new edges and one new vertex being added to the polytope.

We applied this pyramiding method to our entire dataset, which allowed us to generate non-simplicial and non-simple polytopes. This helped increase the diversity of our dataset, adding more variety to the types of combinatorial structures we had collected.

2.1.5. Slicing Polytope. The final method we implemented for generating new polytopes was slicing a corner of the polytope, which involves a Y-Δ operation. In graph theory, this operation consists of slicing a vertex and creating a triangle, which results in two new vertices being added to the original graph. This is due to the fact that one vertex is eliminated but three are added at the same time.

The slicing operation allows for the 'shallow' chopping of one vertex at a time. It can be performed on vertices of any degree, and we do not need to worry about maintaining the polytope structure or the 3D embedding during the operation. It is also important to note that this method preserves the simplicity of the polytope, ensuring that the polytope remains simple after the slicing operation. We will now provide an example of this method below.

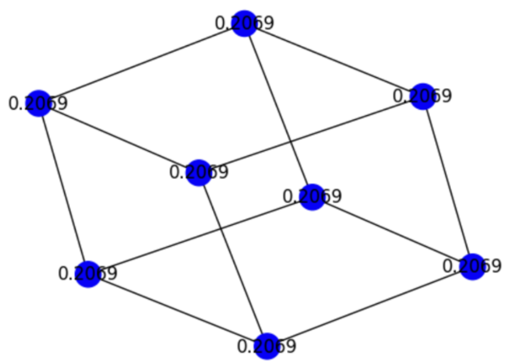


FIGURE 7. The graph of a cube with labeled effective resistance curvature.

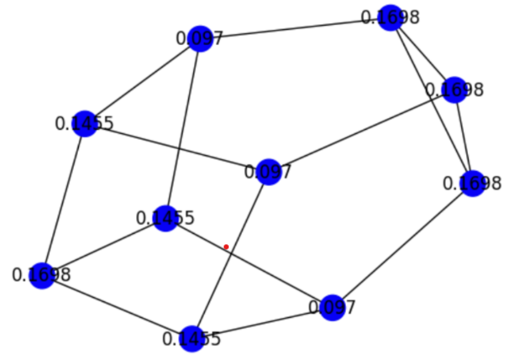


FIGURE 8. The graph resulted of slicing a corner of a cube with labeled effective resistance curvature

Figures 7 and 8 above show a cube and the result of slicing a corner of the cube to create a new polytope. The original cube has 6 quadrilateral faces. After applying the slicing operation, the new polytope has 3 quadrilateral faces, 2 pentagonal faces, and 1 triangular face. Both the original and new polytopes are simple polytopes, showing how this operation preserves simplicity. It is also interesting to observe how slicing the corner of a polytope with everywhere positive resistance curvature resulted in another polytope with every positive resistance curvature.

2.2. About our dataset

Using the five methods mentioned in the previous section, we gathered a diverse set of graphs and added them to a CSV file. After collecting the data, we wrote Python code to eliminate any duplicate graphs. This consisted of removing any graphs that had the same adjacency matrices. We also removed isomorphic graphs to ensure uniqueness. Isomorphic graphs are graphs that are structurally identical despite having different representations such as different vertex labels or edge configurations. In other words, one graph can be transformed into the other by relabeling the vertices while maintaining the same adjacency relationships. By removing duplicate and isomorphic graphs, we ensured that each graph in our dataset represented a unique structure.

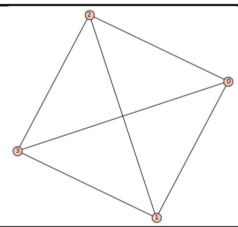
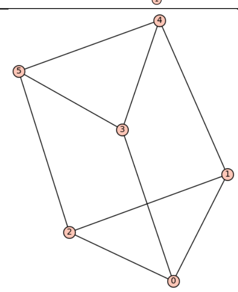
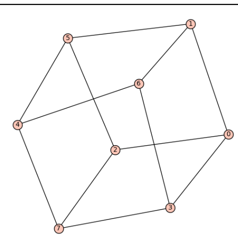
After completing the cleaning process, our final dataset contained 4,216 unique graphs. For each graph, we calculated both the Forman Ricci curvature and the Effective Resistance curvature. Since a polytope is characterized by a set of curvatures associated with its edges and vertices, we organized this data by categorizing it into three types: largest curvature, smallest curvature, and binary curvature. A polytope was assigned a binary curvature value of 1 if all the curvatures in the list were positive, and a value of 0 if any of the curvatures were 0 or negative. After completing the calculations, we found that there were

47 polytopes with everywhere positive Forman Ricci curvature and 299 polytopes with everywhere positive effective resistance ccurvature.

2.3. Everywhere positive polytopes

In this section, we will discuss our findings regarding the polytopes with everywhere positive curvatures. First, we will focus on Forman Ricci curvature as well as all the positive polytopes we found. Then, we will discuss the case for effective resistance curvature.

2.3.1. Forman. By Theorem 6, we know that there are finitely many simple polytopes with everywhere positive Forman Ricci curvature. Through our data, we were able to find four of them, which will be displayed in Table 1 below.

Polytope	Adjacency Matrix	# Faces	# Vertices	# Edges
	$\begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$	4	4	6
	$\begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix}$	5	6	9
	$\begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \end{bmatrix}$	6	8	12

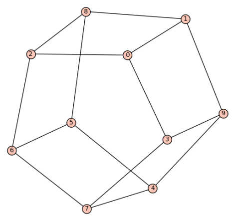
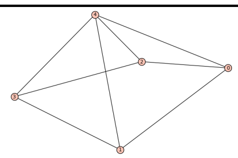
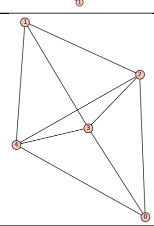
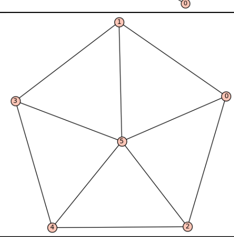
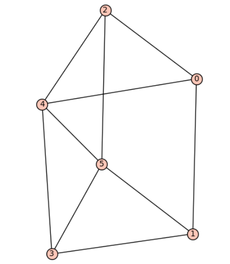
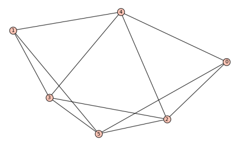
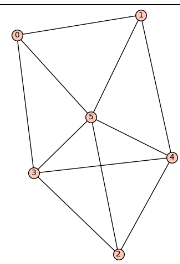
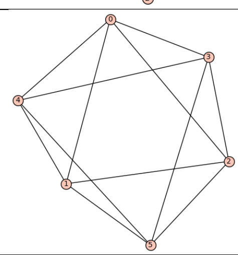
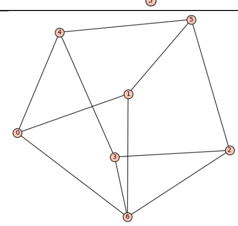
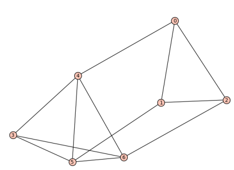
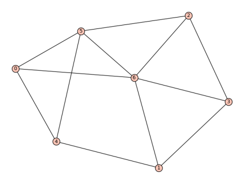
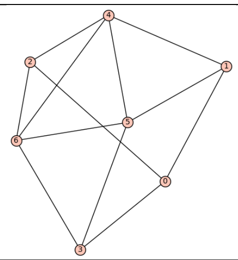
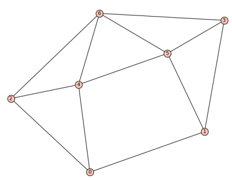
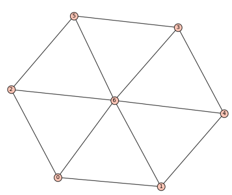
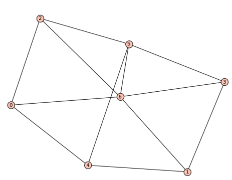
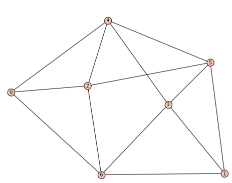
Polytope	Adjacency Matrix	# Faces	# Vertices	# Edges
	$\begin{bmatrix} 0&1&1&1&0&0&0&0&0&0\\ 1&0&0&0&0&0&0&0&1&1\\ 1&0&0&0&0&0&1&0&1&0\\ 1&0&0&0&0&0&0&1&0&1\\ 0&0&0&0&0&1&0&1&0&1\\ 0&0&0&0&1&0&1&0&1&0\\ 0&0&1&0&0&1&0&1&0&0\\ 0&0&0&1&1&0&1&0&0&0\\ 0&1&1&0&0&1&0&0&0&0\\ 0&1&0&1&1&0&0&0&0&0 \end{bmatrix}$	7	10	15

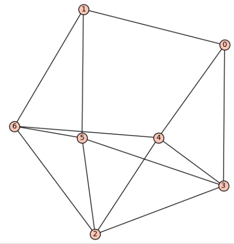
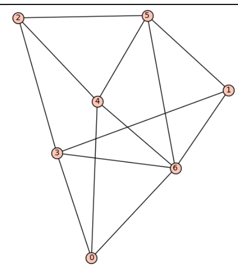
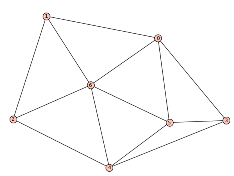
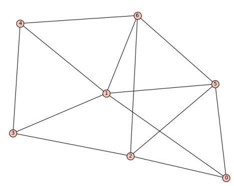
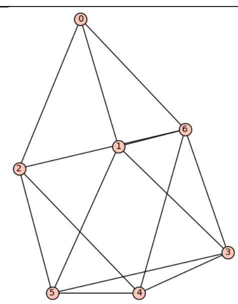
TABLE 1. Simple polytopes with everywhere positive Forman Ricci curvature

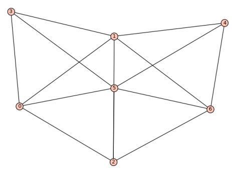
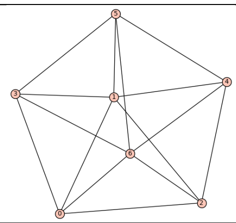
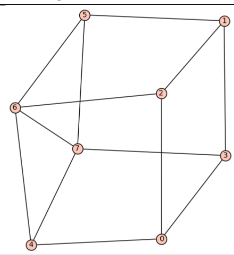
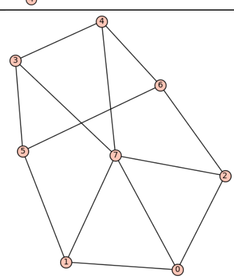
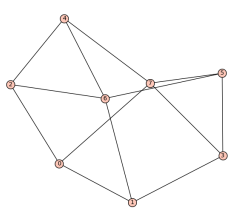
Although the four polytopes with everywhere positive Forman Ricci curvature listed above are the only simple ones with this property, we also identified 43 additional polytopes that are not simple but still exhibit positive curvature. According to Theorem 5, there are finitely many such polytopes. These results are displayed in Table 2 below.

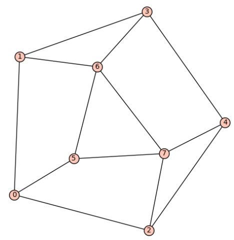
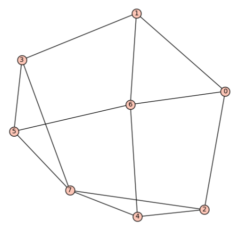
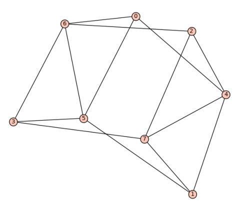
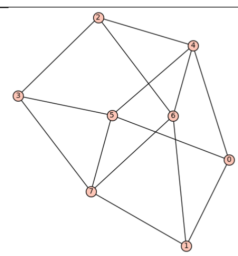
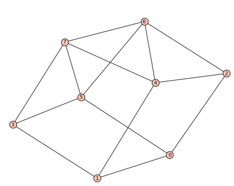
Polytope	Adjacency Matrix	# Faces	# Vertices	# Edges
	$\begin{bmatrix} 0&1&1&0&1\\ 1&0&0&1&1\\ 1&0&0&1&1\\ 0&1&1&0&1\\ 1&1&1&1&0 \end{bmatrix}$	5	5	8
	$\begin{bmatrix} 0&0&1&1&1\\ 0&0&1&1&1\\ 1&1&0&1&1\\ 1&1&1&0&1\\ 1&1&1&1&0 \end{bmatrix}$	6	5	9
	$\begin{bmatrix} 0&1&1&0&0&1\\ 1&0&0&1&0&1\\ 1&0&0&0&1&1\\ 0&1&0&0&1&1\\ 0&0&1&1&0&1\\ 1&1&1&1&1&0 \end{bmatrix}$	6	6	10

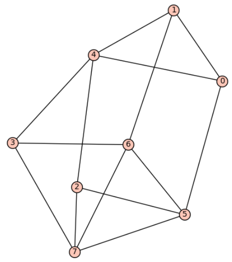
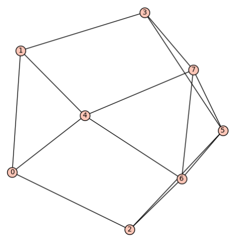
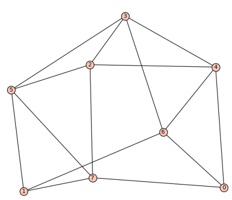
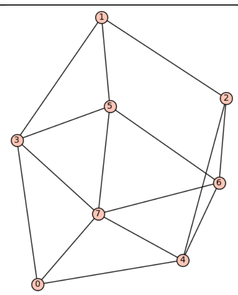
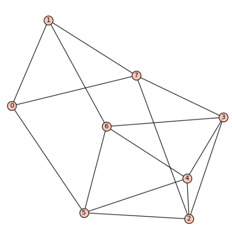
Polytope	Adjacency Matrix	# Faces	# Vertices	# Edges
	$\begin{bmatrix} 0&1&1&0&1&0\\ 1&0&0&1&0&1\\ 1&0&0&0&1&1\\ 0&1&0&0&1&1\\ 1&0&1&1&0&1\\ 0&1&1&1&1&0 \end{bmatrix}$	6	6	10
	$\begin{bmatrix} 0&0&1&0&1&1\\ 0&0&0&1&1&1\\ 1&0&0&1&1&1\\ 0&1&1&0&1&1\\ 1&1&1&1&0&0\\ 1&1&1&1&0&0 \end{bmatrix}$	7	6	11
	$\begin{bmatrix} 0&1&0&1&0&1\\ 1&0&0&0&1&1\\ 0&0&0&1&1&1\\ 1&0&1&0&1&1\\ 0&1&1&1&0&1\\ 1&1&1&1&1&0 \end{bmatrix}$	7	6	11
	$\begin{bmatrix} 0&1&1&1&1&0\\ 1&0&1&0&1&1\\ 1&1&0&1&0&1\\ 1&0&1&0&1&1\\ 1&1&0&1&0&1\\ 0&1&1&1&1&0 \end{bmatrix}$	8	6	12
	$\begin{bmatrix} 0&1&0&0&1&0&1\\ 1&0&0&0&0&1&1\\ 0&0&0&1&0&1&1\\ 0&0&1&0&1&0&1\\ 1&0&0&1&0&1&0\\ 0&1&1&0&1&0&0\\ 1&1&1&1&0&0&0 \end{bmatrix}$	6	7	11
	$\begin{bmatrix} 0&1&1&0&1&0&0\\ 1&0&1&0&0&1&0\\ 1&1&0&0&0&0&1\\ 0&0&0&0&1&1&1\\ 1&0&0&1&0&1&1\\ 0&1&0&1&1&0&1\\ 0&0&1&1&1&1&0 \end{bmatrix}$	7	7	12

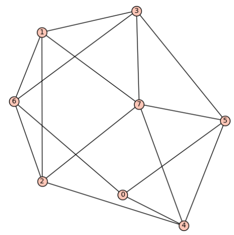
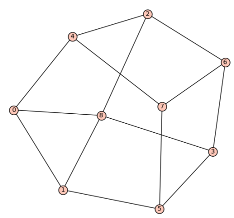
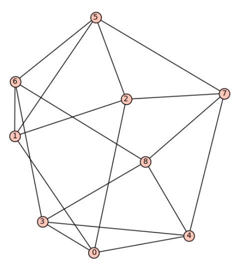
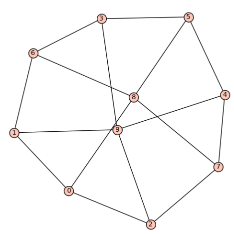
Polytope	**Adjacency Matrix**	**# Faces**	**# Vertices**	**# Edges**
	$\begin{bmatrix} 0&0&0&0&1&1&1\\ 0&0&0&1&1&0&1\\ 0&0&0&1&0&1&1\\ 0&1&1&0&0&0&1\\ 1&1&0&0&0&1&0\\ 1&0&1&0&1&0&1\\ 1&1&1&1&0&1&0 \end{bmatrix}$	7	7	12
	$\begin{bmatrix} 0&1&1&1&0&0&0\\ 1&0&0&0&1&1&0\\ 1&0&0&0&1&0&1\\ 1&0&0&0&0&1&1\\ 0&1&1&0&0&1&1\\ 0&1&0&1&1&0&1\\ 0&0&1&1&1&1&0 \end{bmatrix}$	7	7	12
	$\begin{bmatrix} 0&1&1&0&1&0&0\\ 1&0&0&1&0&1&0\\ 1&0&0&0&1&0&1\\ 0&1&0&0&0&1&1\\ 1&0&1&0&0&1&1\\ 0&1&0&1&1&0&1\\ 0&0&1&1&1&1&0 \end{bmatrix}$	7	7	12
	$\begin{bmatrix} 0&1&1&0&0&0&1\\ 1&0&0&0&1&0&1\\ 1&0&0&0&0&1&1\\ 0&0&0&0&1&1&1\\ 0&1&0&1&0&0&1\\ 0&0&1&1&0&0&1\\ 1&1&1&1&1&1&0 \end{bmatrix}$	7	7	12
	$\begin{bmatrix} 0&0&1&0&1&0&1\\ 0&0&0&1&1&0&1\\ 1&0&0&0&0&1&1\\ 0&1&0&0&0&1&1\\ 1&1&0&0&0&1&0\\ 0&0&1&1&1&0&1\\ 1&1&1&1&0&1&0 \end{bmatrix}$	7	7	12
	$\begin{bmatrix} 0&0&1&0&1&0&1\\ 0&0&0&1&0&1&1\\ 1&0&0&0&1&1&1\\ 0&1&0&0&1&1&1\\ 1&0&1&1&0&1&0\\ 0&1&1&1&1&0&0\\ 1&1&1&0&0&0&0 \end{bmatrix}$	8	7	13

Polytope	**Adjacency Matrix**	**# Faces**	**# Vertices**	**# Edges**
	$\begin{bmatrix} 0&1&0&1&1&0&0\\ 1&0&0&0&0&1&1\\ 0&0&0&1&1&1&1\\ 1&0&1&0&1&1&0\\ 1&0&1&1&0&0&1\\ 0&1&1&1&0&0&1\\ 0&1&1&0&1&1&0 \end{bmatrix}$	8	7	13
	$\begin{bmatrix} 0&0&0&1&1&0&1\\ 0&0&0&1&0&1&1\\ 0&0&0&1&1&1&0\\ 1&1&1&0&0&0&1\\ 1&0&1&0&0&1&1\\ 0&1&1&0&1&0&1\\ 1&1&0&1&1&1&0 \end{bmatrix}$	8	7	13
	$\begin{bmatrix} 0&1&0&1&0&1&1\\ 1&0&1&0&0&0&1\\ 0&1&0&0&1&0&1\\ 1&0&0&0&1&1&0\\ 0&0&1&1&0&1&1\\ 1&0&0&1&1&0&1\\ 1&1&1&0&1&1&0 \end{bmatrix}$	8	7	13
	$\begin{bmatrix} 0&1&1&0&0&1&0\\ 1&0&0&1&1&1&1\\ 1&0&0&1&0&1&1\\ 0&1&1&0&1&0&0\\ 0&1&0&1&0&0&1\\ 1&1&1&0&0&0&1\\ 0&1&1&0&1&1&0 \end{bmatrix}$	8	7	13
	$\begin{bmatrix} 0&1&1&0&0&0&1\\ 1&0&0&1&0&1&1\\ 1&0&0&0&1&1&1\\ 0&1&0&0&1&1&1\\ 0&0&1&1&0&1&1\\ 0&1&1&1&1&0&0\\ 1&1&1&1&1&0&0 \end{bmatrix}$	9	7	14

Polytope	**Adjacency Matrix**	**# Faces**	**# Vertices**	**# Edges**
	$\begin{bmatrix} 0&1&1&1&0&1&0\\ 1&0&1&1&1&0&1\\ 1&1&0&0&0&1&1\\ 1&1&0&0&0&1&0\\ 0&1&0&0&0&1&1\\ 1&0&1&1&1&0&1\\ 0&1&1&0&1&1&0 \end{bmatrix}$	9	7	14
	$\begin{bmatrix} 0&1&1&1&0&0&1\\ 1&0&1&1&1&1&0\\ 1&1&0&0&1&0&1\\ 1&1&0&0&0&1&1\\ 0&1&1&0&0&1&1\\ 0&1&0&1&1&0&1\\ 1&0&1&1&1&1&0 \end{bmatrix}$	10	7	15
	$\begin{bmatrix} 0&0&1&1&1&0&0&0\\ 0&0&1&1&0&1&0&0\\ 1&1&0&0&0&0&1&0\\ 1&1&0&0&0&0&0&1\\ 1&0&0&0&0&0&1&1\\ 0&1&0&0&0&0&1&1\\ 0&0&1&0&1&1&0&1\\ 0&0&0&1&1&1&1&0 \end{bmatrix}$	7	8	13
	$\begin{bmatrix} 0&1&1&0&0&0&0&1\\ 1&0&0&0&0&1&0&1\\ 1&0&0&0&0&0&1&1\\ 0&0&0&0&1&1&0&1\\ 0&0&0&1&0&0&1&1\\ 0&1&0&1&0&0&1&0\\ 0&0&1&0&1&1&0&0\\ 1&1&1&1&1&0&0&0 \end{bmatrix}$	7	8	13
	$\begin{bmatrix} 0&1&1&0&0&0&0&1\\ 1&0&0&1&0&0&1&0\\ 1&0&0&0&1&0&1&0\\ 0&1&0&0&0&1&0&1\\ 0&0&1&0&0&0&1&1\\ 0&0&0&1&0&0&1&1\\ 0&1&1&0&1&1&0&0\\ 1&0&0&1&1&1&0&0 \end{bmatrix}$	7	8	13

Polytope	**Adjacency Matrix**	**# Faces**	**# Vertices**	**# Edges**
	$\begin{bmatrix} 0&1&1&0&0&1&0&0 \\ 1&0&0&1&0&0&1&0 \\ 1&0&0&0&1&0&0&1 \\ 0&1&0&0&1&0&1&0 \\ 0&0&1&1&0&0&0&1 \\ 1&0&0&0&0&0&1&1 \\ 0&1&0&1&0&1&0&1 \\ 0&0&1&0&1&1&1&0 \end{bmatrix}$	7	8	13
	$\begin{bmatrix} 0&1&1&0&0&0&1&0 \\ 1&0&0&1&0&0&1&0 \\ 1&0&0&0&1&0&0&1 \\ 0&1&0&0&0&1&0&1 \\ 0&0&1&0&0&0&1&1 \\ 0&0&0&1&0&0&1&1 \\ 1&1&0&0&1&1&0&0 \\ 0&0&1&1&1&1&0&0 \end{bmatrix}$	7	8	13
	$\begin{bmatrix} 0&0&0&0&1&1&1&0 \\ 0&0&0&0&1&1&0&1 \\ 0&0&0&0&1&0&1&1 \\ 0&0&0&0&0&1&1&1 \\ 1&1&1&0&0&0&0&1 \\ 1&1&0&1&0&0&1&0 \\ 1&0&1&1&0&1&0&0 \\ 0&1&1&1&1&0&0&0 \end{bmatrix}$	8	8	14
	$\begin{bmatrix} 0&1&0&0&1&1&0&0 \\ 1&0&0&0&0&0&1&1 \\ 0&0&0&1&1&0&1&0 \\ 0&0&1&0&0&1&0&1 \\ 1&0&1&0&0&1&1&0 \\ 1&0&0&1&1&0&0&1 \\ 0&1&1&0&1&0&0&1 \\ 0&1&0&1&0&1&1&0 \end{bmatrix}$	8	8	14
	$\begin{bmatrix} 0&1&1&0&0&1&0&0 \\ 1&0&0&1&1&0&0&0 \\ 1&0&0&0&1&0&1&0 \\ 0&1&0&0&0&1&0&1 \\ 0&1&1&0&0&0&1&1 \\ 1&0&0&1&0&0&1&1 \\ 0&0&1&0&1&1&0&1 \\ 0&0&0&1&1&1&1&0 \end{bmatrix}$	8	8	14

Polytope	**Adjacency Matrix**	**# Faces**	**# Vertices**	**# Edges**
	$\begin{bmatrix} 0&1&0&0&1&1&0&0 \\ 1&0&0&0&1&0&1&0 \\ 0&0&0&0&1&1&0&1 \\ 0&0&0&0&1&0&1&1 \\ 1&1&1&1&0&0&0&0 \\ 1&0&1&0&0&0&1&1 \\ 0&1&0&1&0&1&0&1 \\ 0&0&1&1&0&1&1&0 \end{bmatrix}$	8	8	14
	$\begin{bmatrix} 0&1&1&0&1&0&0&0 \\ 1&0&0&1&1&0&0&0 \\ 1&0&0&0&0&1&1&0 \\ 0&1&0&0&0&1&0&1 \\ 1&1&0&0&0&0&1&1 \\ 0&0&1&1&0&0&1&1 \\ 0&0&1&0&1&1&0&1 \\ 0&0&0&1&1&1&1&0 \end{bmatrix}$	8	8	14
	$\begin{bmatrix} 0&0&0&0&1&0&1&1 \\ 0&0&0&0&0&1&1&1 \\ 0&0&0&1&1&1&0&1 \\ 0&0&1&0&1&1&1&0 \\ 1&0&1&1&0&0&1&0 \\ 0&1&1&1&0&0&0&1 \\ 1&1&0&1&1&0&0&0 \\ 1&1&1&0&0&1&0&0 \end{bmatrix}$	9	8	15
	$\begin{bmatrix} 0&0&0&1&1&0&0&1 \\ 0&0&1&1&0&1&0&0 \\ 0&1&0&0&1&0&1&0 \\ 1&1&0&0&0&1&0&1 \\ 1&0&1&0&0&0&1&1 \\ 0&1&0&1&0&0&1&1 \\ 0&0&1&0&1&1&0&1 \\ 1&0&0&1&1&1&1&0 \end{bmatrix}$	9	8	15
	$\begin{bmatrix} 0&1&0&0&0&1&0&1 \\ 1&0&0&0&0&0&1&1 \\ 0&0&0&1&1&1&0&1 \\ 0&0&1&0&1&0&1&1 \\ 0&0&1&1&0&1&1&0 \\ 1&0&1&0&1&0&1&0 \\ 0&1&0&1&1&1&0&0 \\ 1&1&1&1&0&0&0&0 \end{bmatrix}$	9	8	15

Polytope	**Adjacency Matrix**	**# Faces**	**# Vertices**	**# Edges**
	$\begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \end{bmatrix}$	10	8	16
	$\begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \end{bmatrix}$	7	9	14
	$\begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$	11	9	18
	$\begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$	8	10	16

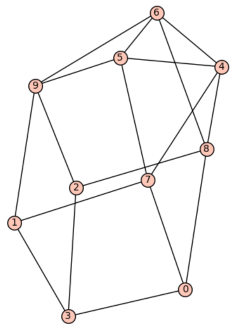
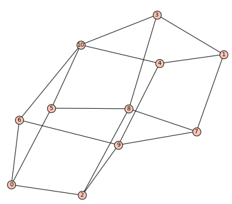
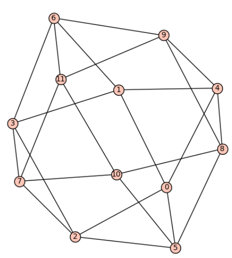
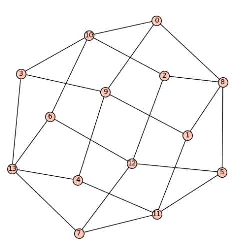
Polytope	**Adjacency Matrix**	**# Faces**	**# Vertices**	**# Edges**
	$\begin{bmatrix} 0&0&0&1&0&0&0&1&1&0 \\ 0&0&0&1&0&0&0&1&0&1 \\ 0&0&0&1&0&0&0&0&1&1 \\ 1&1&1&0&0&0&0&0&0&0 \\ 0&0&0&0&0&1&1&1&1&0 \\ 0&0&0&0&1&0&1&1&0&1 \\ 0&0&0&0&1&1&0&0&1&1 \\ 1&1&0&0&1&1&0&0&0&0 \\ 1&0&1&0&1&0&1&0&0&0 \\ 0&1&1&0&0&1&1&0&0&0 \end{bmatrix}$	10	10	18
	$\begin{bmatrix} 0&0&1&0&0&1&1&0&0&0&0 \\ 0&0&0&1&1&0&0&1&0&0&0 \\ 1&0&0&0&0&0&0&0&1&1&0 \\ 0&1&0&0&0&0&0&0&1&0&1 \\ 0&1&0&0&0&0&0&0&0&1&1 \\ 1&0&0&0&0&0&0&0&1&0&1 \\ 1&0&0&0&0&0&0&0&0&1&1 \\ 0&1&0&0&0&0&0&0&1&1&0 \\ 0&0&1&1&0&1&0&1&0&0&0 \\ 0&0&1&0&1&0&1&1&0&0&0 \\ 0&0&0&1&1&1&1&0&0&0&0 \end{bmatrix}$	9	11	18
	$\begin{bmatrix} 0&1&1&0&1&1&0&0&0&0&0&0 \\ 1&0&0&1&1&0&1&0&0&0&0&0 \\ 1&0&0&1&0&1&0&1&0&0&0&0 \\ 0&1&1&0&0&0&1&1&0&0&0&0 \\ 1&1&0&0&0&0&0&0&1&1&0&0 \\ 1&0&1&0&0&0&0&0&1&0&1&0 \\ 0&1&0&1&0&0&0&0&0&1&0&1 \\ 0&0&1&1&0&0&0&0&0&0&1&1 \\ 0&0&0&0&1&1&0&0&0&1&1&0 \\ 0&0&0&0&1&0&1&0&1&0&0&1 \\ 0&0&0&0&0&1&0&1&1&0&0&1 \\ 0&0&0&0&0&0&1&1&0&1&1&0 \end{bmatrix}$	14	12	24
	$\begin{bmatrix} 0&0&0&0&0&0&0&0&1&1&1&0&0&0 \\ 0&0&0&0&0&0&0&0&1&1&0&1&0&0 \\ 0&0&0&0&0&0&0&0&1&0&1&0&1&0 \\ 0&0&0&0&0&0&0&0&0&1&1&0&0&1 \\ 0&0&0&0&0&0&0&0&0&1&0&1&0&1 \\ 0&0&0&0&0&0&0&0&1&0&0&1&1&0 \\ 0&0&0&0&0&0&0&0&0&0&1&0&1&1 \\ 0&0&0&0&0&0&0&0&0&0&0&1&1&1 \\ 1&1&1&0&0&1&0&0&0&0&0&0&0&0 \\ 1&1&0&1&1&0&0&0&0&0&0&0&0&0 \\ 1&0&1&1&0&0&1&0&0&0&0&0&0&0 \\ 0&1&0&0&1&1&0&1&0&0&0&0&0&0 \\ 0&0&1&0&0&1&1&1&0&0&0&0&0&0 \\ 0&0&0&1&1&0&1&1&0&0&0&0&0&0 \end{bmatrix}$	9	11	18

TABLE 2. Polytopes with everywhere positive Forman Ricci curvature

Altogether, using our dataset, we were able to find a total of 47 polytopes with everywhere positive Forman Ricci curvature.

2.3.2. Resistance. We will now discuss the polytopes that have everywhere positive effective resistance curvature. One important discovery is that the prism graphs contain everywhere positive resistance curvature. A prism graph is the 1-skeleton of a k-gonal prism, meaning it is formed by connecting the vertices of two parallel k-gons with edges between corresponding vertices of the two polygons. One common example of a prism graph is the cube, which is shown in Figure 7. Since the prism graphs form an infinite family, this implies there are infinitely many polytopes with everywhere positive effective resistance curvature. As a result, we cannot display all of these polytopes, as there are infinitely many of them.

2.4. Conclusion

In conclusion, we have completed our exploration of polytopes with positive curvature in dimension 3 for these two curvature notions. Through our investigation, we identified and displayed a finite number of polytopes exhibiting everywhere positive Forman Ricci curvature, exemplifying the results of our theoretical findings. Additionally, we found that, in contrast, there are infinitely many polytopes with positive effective resistance curvature. These results are particularly relevant to the study of diameter bounds, as the properties of positive discrete curvatures can be leveraged to bound the diameter of polytopes, which has important implications for optimizing search algorithms and computational complexity.

Bibliography

1. K. Coolsaet, S. D'hondt, and J. Goedgebeur, *House of graphs 2.0: A database of interesting graphs and more*, 2023, pp. 97–107. 13
2. K. Devriendt and R. Lambiotte, *Discrete curvature on graphs from the effective resistance*, J. Phys. Complex **3** (2022), 025008. 3
3. K. Devriendt, A. Ottolini, and S. Steinerberger, *Graph curvature via resistance distance*, Discrete Applied Mathematics **348** (2024), 68–78. 8
4. R. Forman, *Bochner's method for cell complexes and combinatorial ricci curvature*, Discrete Computational Geometry **29** (2003), 323–374. 3
5. C. Godsil and G. Royle, *Algebraic graph theory*, 1 ed., Graduate Texts In Mathematics, Springer Science+Business Media, LLC, New York, NY, 2001. 8
6. J. Malkevitch, *Euler's polyhedral formula*, 2005. 1, 2
7. M. Miller and J. Sirán, *Moore graphs and beyond: A survey of the degree/diameter problem*, The Electronic Journal of Combinatorics (2005). 2
8. N. Peyerimhoff, *Curvature notions on graphs*, Leeds Summer School (18-19 July 2019). 1
9. G.M. Ziegler, *Steinitz' theorem for 3-polytopes*, Lectures on Polytopes, Graduate Texts in Mathematics, vol. 152, Springer, New York, NY, 1995. 13