2020

A Discrete Analogue for the Poincaré-Hopf Theorem

Savana Ammons

A Discrete Analogue for the Poincaré-Hopf Theorem

Savana Ammons

Francis Edward Su, Advisor

Dagan Karp, Reader

Department of Mathematics

May, 2020

Abstract

In this thesis, we develop a discrete analogue to the Poincaré–Hopf Theorem. We define the notion of a vector field on a graph, and establish an index theory for such a field. Specifically, we create well-defined indices for the nodes and “cells" formed by a planar graph. Then, we show that the sum of these indices remains constant for certain types of planar graphs, regardless of the discrete vector fields they have.

Contents

List of Figures

Acknowledgments

I would like to thank my advisor, Francis Su, for his guidance and patience throughout the school year. Regardless of how frustrated I was when I entered his office, I always left with a clear plan and a brighter outlook, which helped make senior thesis such an enjoyable experience. In addition, Professor Su was an invaluable mentor when it came to navigating graduate school and fellowship applications, and he helped make the process a little less stressful. Overall, I am incredibly grateful for the advice he gave me regarding thesis, my math career, and life beyond mathematics.

Additionally, I would like to thank my second reader, Dagan Karp. His comments made me consider my arguments, and ultimately the scope of my thesis, much more carefully. This helped me refine my goals for my thesis earlier on, which made my path forward much clearer. He also gave me incredibly useful advice on how to develop my voice as a mathematical writer.

Lastly, I would like to thank Oliver Knill for his correspondence, which helped me shape the goals of my thesis, and helped me assess what related work has already been done.

Chapter 1

Introduction

The Poincaré-Hopf theorem is a important result in differential topology; it reveals how an object's topology places limits on its differential structure. It does so by using *smooth vector fields* (analytical objects) to compute a topological invariant of a *smooth manifold* (a set of points that locally looks like a Euclidean space). This topological invariant is the manifold's *Euler characteristic* (χ), which tells us about the manifold's shape and structure. Applications of the theorem can be found throughout mathematics, and even other disciplines, including computer science (Miura and Nakada (2017)), chemistry (Balanarayan and Gadre (2003)), and materials sciences (Zhang et al. (2018)).

The Poincaré-Hopf theorem roughly states that, for any smooth vector field v on a smooth manifold M, we can compute the Euler characteristic of M by examining the points where v vanishes. Here, "examine" means calculating each point's *index*: a number that quantifies how v behaves around the point.

Specifically, the theorem asserts that M's Euler characteristic equals the sum of the indices of v's vanishing points. For example, suppose that our manifold is a two-sphere with a vector field on it, like the one in Figure 1.1. There are two points where this vector field vanishes: at the very top, and at the very bottom. As it turns out, each of these points has an index of +1. According to the Poincaré-Hopf theorem, the Euler characteristic of a two-sphere should then be +2. As we'll see in the next section, this is indeed the case.

It is important to note that our choice of vector field is irrelevant. According to the Poincaré-Hopf theorem, *any* smooth vector field on a two-sphere would yield an index sum of +2. Similarly, a torus' Euler characteristic is 0,

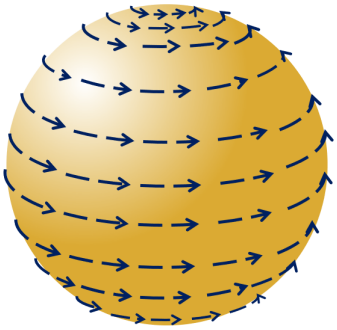


Figure 1.1 A two-sphere with a smooth vector field on it. The two vanishing points in the field each have an index of +1.

and the index sum of any smooth vector field on a torus will always be 0.

A chief characteristic of the Poincaré-Hopf theorem is that it only pertains to continuous settings: it is only defined on compact smooth manifolds, not discrete objects. However, as Knill (2018) and numerous other papers indicate, many continuous theorems have discrete analogues. In fact, work by Oliver Knill in Knill (2002) indicates that the Poincaré-Hopf theorem may as well. Instead of analyzing vector fields on manifolds, Knill put vector fields on *graphs*. He then created a new index definition for those fields' zeros, and showed that the sum of their indices remains constant. However, he only proved this invariance for conservative vector fields, not generic smooth vector fields.

In this thesis, we define a slightly different index than Knill's to prove the more general case, thus providing a discrete analogue to the Poincaré-Hopf theorem. The purpose of this was to hopefully reveal connections between graph theory and differential topology, just as the continuous theorem did for analysis and differential topology. From there, we could ideally prove more discrete results, and perhaps even continuous ones.

As it turns out though, the work presented in this thesis actually could have lead to an important result from knot theory. We know this because, when adding finishing touches to this thesis, we discovered that a discrete analogue to the Poincaré-Hopf theorem that was very similar to ours already existed. According to Gordon (1997), this other discrete analogue was developed by Glass, and played a pivotal role in showing that knots are determined by their complements. Despite this, there are aspects of this thesis that are novel. Namely, this thesis provides intuitive insights into the discrete index theory that lead to our discrete analogue of the Poincaré-Hopf

theorem. We then use this index theory to provide an alternative proof to a recently posed combinatorial problem related to the Game of Cycles.

In the next chapter, we establish the terminology needed to formally present the Poincaré-Hopf theorem. Then, we introduce terms and concepts related to the theorem's discretization.

Chapter 2

The Poincaré-Hopf Theorem's Elements, and their Discrete Counterparts

2.1 Formalizing Poincaré-Hopf

The Poincaré-Hopf theorem takes place on **compact smooth manifolds**. At a high level, a smooth manifold is a surface that "looks locally like $\mathbb{R}^n$" (Milnor and Weaver (1965)); it is "compact" if each of its open covers has a finite subcover. Some examples of compact, smooth manifolds include surfaces like spheres, tori, and Klein bottles.

The theorem specifically involves compact smooth manifolds that have *smooth vector fields* on them.

Definition 2.1.1. A **vector field** is a mapping that assigns a tangent vector to each point in the manifold. The mapping is considered "smooth" if all of its partial derivatives exist, and are continuous (Milnor and Weaver (1965)).

Recall that the Poincaré-Hopf theorem involved a vector field's vanishing points. These points are known as *zeros*.

Definition 2.1.2. A **zero** of a vector field is a point in the vector field that is assigned the zero vector.

Example 2.1.3. Figure 1.1 showed a vector field on a sphere; its zeros were at the very top and very bottom of the sphere. Figure 2.1 shows a vector field on part of a two-dimensional manifold; it has a zero at the origin.

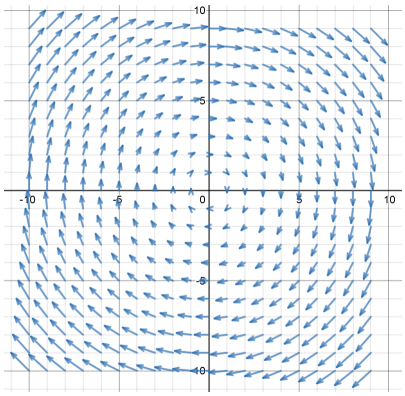


Figure 2.1 A vector field on part of a two-dimensional manifold.myV (2019)

Perhaps the most important definition associated with the Poincaré-Hopf theorem is that of an *index*. The index is a number defined for each zero in a vector field that reveals how the vector field locally behaves around the zero. To motivate what an index is, and how we might calculate it, suppose that you and I are standing in a particularly windy meadow. As we walk around the meadow, we find that the wind almost always pushes and pulls us. Curiously though, there are some points where the wind doesn't blow at all. Wanting to better understand how the wind is behaving around these points, we decide to examine one of them. You choose to stand directly at one of these points, while I walk counterclockwise around you in a circle. As I walk, I feel the wind pushing and pulling me in different directions, and you take note of these directions by marking them on the circle.

Specifically, as you watch me, you look in the direction I was being pushed, and make a marking on that part of the circle. For example, if I am being pushed to your left, you make a mark on the part of the circle that's currently on your left. Similarly, if I was pushed to your right, you make a mark on the rightmost part of the circle. Meanwhile, if I am being pushed away from you, you mark the part of the circle where I currently stand. If I was pulled towards you, you mark the opposite side of the circle.

Once I complete the circle, we follow the markings you made (in the order you made them) by walking along the circle. We then count the number of times we make a complete trip around the circle by following your markings.

In this example, the meadow is a two-dimensional manifold, the wind is a vector field, and the place where the wind stops is a zero. The index

of the zero is the number of times we walked completely around the circle (counterclockwise) by following your markings. By doing this, we attempt to quantify how I felt as I walked around you. Was I always being pushed or pulled towards you, was I constantly being tugged in different directions, or was there a mixture of the two? The index helps us calculate this[1].

To better understand this concept outside of our example, consider Figure 2.2, which shows some vector field zeros, and their indices.

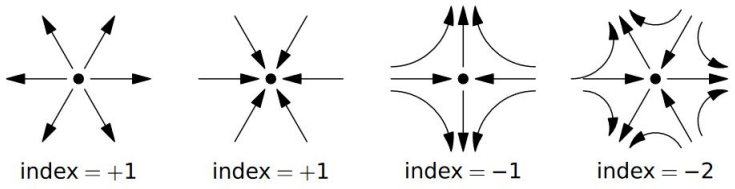


Figure 2.2 Four zeros in vector fields, along with their indices.

Example 2.1.4. Let's calculate the index of the second vector field zero from the left in Figure 2.2. First, we impose a unit circle around the zero, as shown in Figure 2.3a. Then, starting[2] from the vector labeled 1, we follow the vectors counterclockwise, and record what direction they point to on the circle. We end on the same vector we started at. This process is shown in Figure 2.3b.

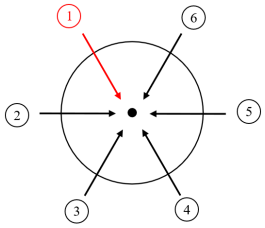
In this case, the vectors trace out one counterclockwise path around the circle. Hence, this zero has an index of +1. Here, the index's sign refers to what direction the path was traced; if the vectors' directions traced out a *clockwise* path instead, the index would be −1.

The second zero from the left in Figure 2.2 is what's known as a **sink**: all the vectors around it point *towards it*. Meanwhile, the leftmost zero from Figure 2.2 is called a **source**: all the vectors around it point *away from it*. It's interesting to note that, while sinks and sources appear to be opposites, they still have the same index: +1, as Figure 2.2 indicates.

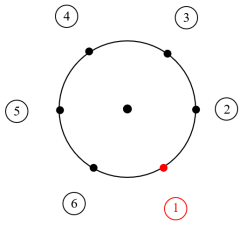
Example 2.1.5. Let's now calculate the index of the third zero from the left

[1] In general, the index applies to manifolds of *any* dimension, not just two dimensions. In this thesis though, only the two-dimensional version is relevant. For a more general index, see Milnor and Weaver (1965).

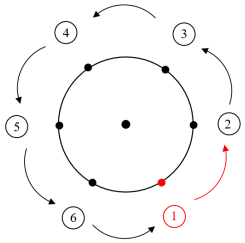
[2] This starting vector was chosen arbitrarily; the index would be the same regardless of which vector we started with.



a. Impose a circle around the zero. Then, follow the vectors that intersect the circle counterclockwise.



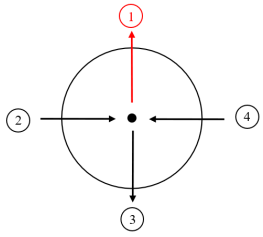
b. Record where each vector pointed to on the circle.



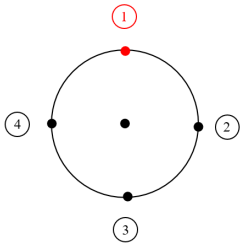
c. Starting from where vector 1 pointed, follow the vectors' directions. In this example, this results in one complete counterclockwise trip around the circle.

Figure 2.3 A zero in a vector field, with a circle imposed on it. Moving from vector 1, to vector 2, and so on, we see that the vectors trace one complete counterclockwise path around the circle. So, the index is +1.

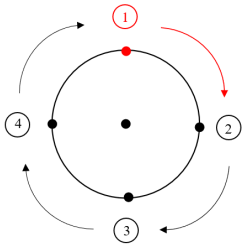
in Figure 2.2. We again start by placing a unit circle around this zero, as shown in Figure 2.4.



a. Impose a circle around the zero. Then, follow the vectors counterclockwise around the circle.



b. Note where each vector pointed to on the circle. In this case, vector 2 pointed to the right, while vector 4 pointed left.



c. Starting from vector 1's direction, follow where the vectors pointed. In this case, doing so results in a clockwise path.

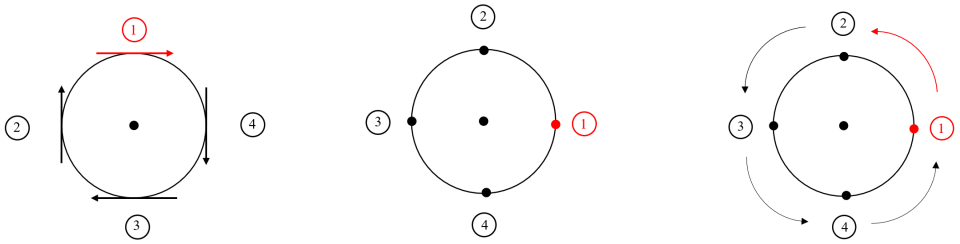
Figure 2.4 Moving from vector 1, to vector 2, and so on, the vectors trace one complete clockwise path around the circle. Hence, the index is -1.

Then, starting from the vector labeled 1, we follow the vectors counterclockwise, and record what direction they point to on the circle. While

doing so, we see that vector 2 points to the right, and vector 4 points to the left. Hence, once we follow these directions, as in Figure 2.4c, we follow the circle in the *clockwise* direction. So, the index for this zero is -1.

Example 2.1.6. For our final index example, let's calculate the index of the vector field zero from Figure 2.1: the vector field that swirled around its zero. As before, we place a unit circle around the zero, as shown in Figure 2.5.

Starting with the vector labeled 1, follow the vectors counterclockwise and record what direction they point to on the circle. Notice how, because vector 1 points to the right, we mark the rightmost point on the circle. Similarly, because vector 2 points up, we make a marking on the top of the circle. Following the vectors' directions, we trace out one counterclockwise rotation, as Figure 2.5c shows. Therefore, the index for this zero is +1.



a. Impose a circle around the zero. Then, follow the vectors counterclockwise around the circle.

b. Vector 1 points right, so mark the rightmost part of the circle for vector 1. Vector 2 points up, so mark the top. Continue for vectors 3 and 4.

c. Following the directions gives us one counterclockwise path.

Figure 2.5 An example of a swirling vector field's zero that's index is +1.

We formally define[3] the index as follows:

Definition 2.1.7. Let M be an n-dimensional compact smooth manifold, and let z be one of finitely many zeros in a smooth vector field v on M. If we impose a circle S centered at z onto v, z's **index** is the number of times the vectors in v trace out a complete counterclockwise rotation around S.

[3]This definition is more intuitive, but less rigorous, than the index definition you will see in most differential topology texts. We omit the details because they aren't necessary to understand the work in this thesis. For a more detailed definition, see Milnor and Weaver (1965).

The last bit of background we need to understand the Poincaré-Hopf theorem is a manifold's *Euler characteristic*: the topological invariant the theorem helps us calculate. Before defining the Euler characteristic though, we must first introduce the concept of a **simplex**. An n-simplex is an n-dimensional triangle. For example, a 0-simplex is a vertex, a 1-simplex is a line segment, a 3-simplex is a tetrahedron, and so on. An **n-dimensional simplicial complex** is a collection of i-simplices (where n is the largest of the i's) that are glued together such that intersections of simplices are simplices (Scoville (2019)).

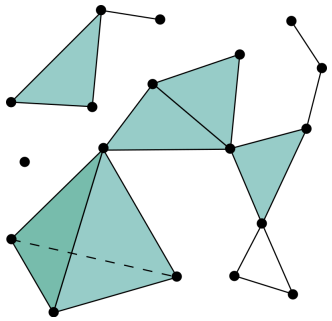


Figure 2.6 A three-dimensional simplicial complex. Figure from sim (2019).

As in Scoville (2019), we define the Euler characteristic as follows:

Definition 2.1.8. Let K be an n-dimensional simplicial complex, and let $c_i(K)$ denote the number of i-simplices of K. The **Euler characteristic** of K, $\chi(K)$, is given by:

$$\chi(K) = \sum_{i=0}^{n} (-1)^i c_i(K).$$

So, to compute a simplicial complex's Euler characteristic, we count the number of vertices it has, minus the number of edges, plus the number of triangles, minus the number of tetrahedra, and so on. To compute the Euler characteristic of a manifold, we first express it as a simplicial complex (i.e. we find a *triangulation* of the manifold). Then, we calculate the Euler characteristic of this simplicial complex. Note that, while there can be many triangulations of the same manifold, each will have the same Euler characteristic.

Example 2.1.9. To calculate a two-sphere's Euler characteristic, we must first express it as a simplicial complex. To do so, imagine drawing an equator

and two meridians on the sphere's surface so that it is divided into eight triangles, as shown in Figure 2.7. Hence, the sphere is really just a simplicial complex consisting of six 0-simplices (i.e. points, which are circled), twelve 1-simplices (edges), and eight 2-simplices (faces). Therefore, the sphere's Euler characteristic is $6 - 12 + 8 = +2$, as we claimed before.

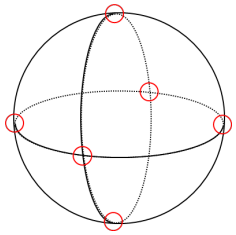


Figure 2.7 A sphere drawn as a two-dimensional simplicial complex. The meridian that creates the sphere's outline, the sphere's equator, and the sphere's prime meridian divide it into simplices.

We can now formally state the Poincaré-Hopf theorem.

Theorem 2.1.10 (Poincaré-Hopf)**.** *Let M be a compact smooth manifold and let v be a smooth vector field on M with finitely many zeros z_i. Then, regardless of our choice of v*:

$$\sum_i index(z_i) = \chi(M).$$

2.2 Discrete Vector Fields and Indices

With the aspects of the Poincaré-Hopf theorem defined, we now discretize these concepts. As alluded to before, we will follow Knill's precedent of replacing compact smooth manifolds with finite, connected, planar graphs. Then, instead of using smooth vector fields, we will place *discrete vector fields* on those graphs.

Definition 2.2.1. A **discrete vector field** on an undirected, finite, connected, planar graph G assigns a "direction" to every edge in E. Specifically, if v_1 and v_2 are the vertices adjacent to an edge e, then the vector on e is either directed towards v_1 or v_2.

Thus, a graph with a discrete vector field is just a directed graph.

Recall that the continuous Poincaré-Hopf theorem involved the zeros of vector fields. What should the zeros of our discrete vector fields be? This will be the first of many instances where we turn to the continuous setting for guidance. Specifically, while discrete vector fields are easy to work with, they lack useful information for discretizing continuous objects and concepts. Hence, it is often helpful to instead use a *completion* of a discrete vector field. Before formally defining a completion, we will need to define a related concept, known as a *cell* of a planar graph.

Definition 2.2.2. Let G be a planar graph with a discrete vector field, and consider an embedding of G into $\mathbb{R}^2$. This embedding divides the plane into regions, some of which are bounded. A **cell** of G is set of edges and vertices that form a bounded region when G is embedded in the plane.

We can now introduce the concept of a completion.

Definition 2.2.3. Let C be a cell of a planar graph G, which has a discrete vector field v. Now, let C' be the bounded region that corresponds to C when G is embedded in $\mathbb{R}^2$. A **completion** of C is a continuous vector field w on C' that agrees with v in the following sense: for any point p on an edge E of C', $w(p)$ has the same direction as v does on E in C.

We will make this concept clearer with the next example, but before doing so, we must make two important remarks: First, w's vectors at the vertices of C' will have to be zero vectors for a completion's continuity condition to be met. Second, note that a cell completion is not unique. There may be many continuous vector fields that "agree" with a cell's discrete vector field in the way described above, yet do very different things away from the edges.

Now, to get a better understanding of a completion, let's see how to construct one:

Example 2.2.4. To construct a completion, we consider what a continuous vector field within the graph would look like if it had to match the discrete vector field on the graph's boundaries. In Figure 2.8, both the vectors in the discrete vector field that are adjacent to the leftmost vertex point into it. Hence, the vectors in a completion must point into this vertex as well. Conversely, the vectors in the discrete vector field flow through the rightmost vertex from top to bottom. Hence, the vectors in our completion do the same. Continuing this line of reasoning, we derive the above completion.

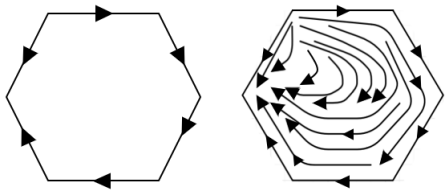


Figure 2.8 On the left: a graph cell with a discrete vector field. On the right: the cell's completion

Let's use the idea of a completion to determine what the zeros for a discrete vector field should be. Suppose we have a graph G with a discrete vector field on it. If we have completions for each of G's cells, then where would these completions' zeros be with respect to G's cells and vertices? The zeros may land directly on one of G's vertices (as in the leftmost node in Figure 2.8). However, they could also lie somewhere within one of G's cells. For example, this would happen if the vectors in G's discrete vector field cycle around the boundary of one of G's cells. This suggests that both vertices *and* cells should be considered zeros.

As we will see in the next chapter, we must treat the vertices on the graph's boundary, and the cells that include boundary edges or boundary vertices, slightly differently. Hence, for this section, we will just discuss the vertices that aren't on the boundary, and the cells that don't share an edge or vertex with the graph's boundary. We will refer to these vertices and cells as the *internal vertices* and *internal cells*.

But what should the indices for these zeros be? One intuitive approach is to make the node index the exact same as the index for a vector field zero on a two-dimensional manifold. Hence, as in Examples 2.3, 2.4, 2.5, we calculate the node index by imposing a circle on the node, and seeing how many rotations the vectors that intersect the circle trace.

To make this more precise, let G be a finite, planar, connected graph with a discrete vector field v, and let z be a node in G. If we impose a circle S centered at z onto G, then we would like to define z's index to be the number of times the vectors in v trace out a complete counterclockwise rotation around S. Note how similar this is to Definition 2.1.7.

We employ a similar technique to calculate the index of a cell. We arbitrarily choose one edge of the cell to begin with. Then, we record what direction that edge's vector points in. Then, moving counterclockwise, we see what direction the next edge's vector points in. We continue following

the cell around its perimeter until we again reach the edge we started at. As with the node index, we record where the vectors would point if they were centered on a circle. The cell index is then the number of times these directions trace a complete counterclockwise path.

Below is an example of several cells, and their indices:

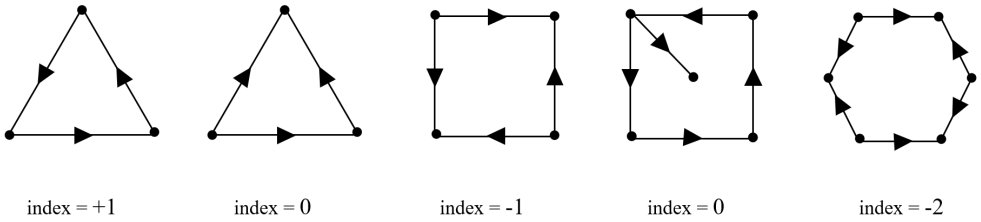
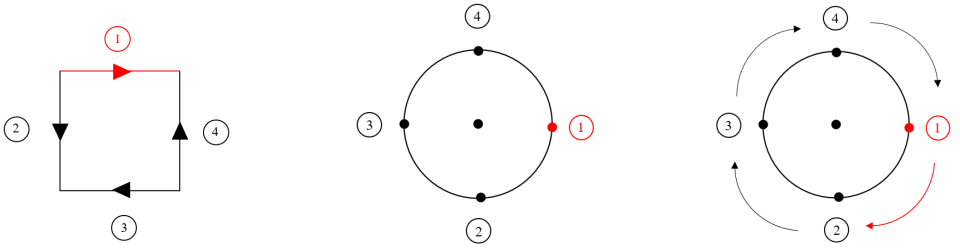


Figure 2.9 Graph cells with discrete vector fields on them, along with their indices.

Example 2.2.5. Let's calculate the index of the middle cell from Figure 2.9. If we choose to start at the top arrow (labeled vector 1 in Figure 2.10), then the vectors point right, then down, then left, then up. Hence, as Figure 2.10c shows, the vectors trace out one complete clockwise rotation. So, the cell's index is -1.



a. Follow the vectors that intersect the circle counterclockwise.

b. Record where each vector pointed to on the circle.

c. The vectors' directions trace out one complete clockwise rotation.

Figure 2.10 A cell in a graph with a discrete vector field. Its index is -1.

One thing you may have noticed from this example is that it is not entirely clear how to follow the points the vectors trace out along the circle. For example, to move from point 1 to point 2 in Figure 2.10b, we could have gone clockwise, as in Figure 2.10c. However, we could have also gone

counterclockwise, turning 270° to get to point 2. While it wouldn't have made a difference in the index in this example, there are other examples where the resulting index would be different. Namely, cells that form non-convex bounded regions when embedded in $\mathbb{R}^2$ yield different indices depending on which way we follow the vector directions. So, how did we know to follow the marked points as we did in Figure 2.10c?

To determine how to follow the markings made on the circle, we consult one of the cell's completions. Specifically, we follow the markings made on our circle by moving in the same direction as the vectors within a given completion. If we use the same scenario as in Example 2.1.3, this is akin to following our markings along the circle by taking the path where the wind blows our backs, not our faces.

Consider Figure 2.11. Suppose that we were in the middle of the top edge of the cell's completion, and we were trying to get to the middle of the left edge by walking through the cell. How would the vectors push us? First, we would be pushed to the right. Then, as we make our way towards the left edge, we'd start to get pushed back to the left, and downward. We would mimic these pushes in Figure 2.10b as we move from point 1 to point 2. Specifically, we'd move down from point 1 and to the left towards point 2. Hence, we'd move in the clockwise direction.

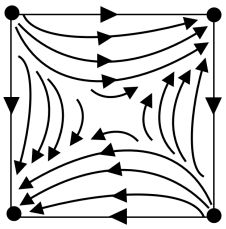


Figure 2.11 A completion of the square cell from Figure 2.10.

Example 2.2.6. Next, consider a cell that has a leaf, like the square one with an index of 0 in Figure 2.9. The same cell is reproduced on the left side of the arrow in Figure 2.12. In that figure, we've colored the leaf blue, and the node it's adjacent to red.

Recall that, when calculating the index of a cell, we must follow the arrows on the cell's perimeter. When doing so for a cell with a leaf though, we will have to traverse the edge that's incident to the leaf twice. Hence, following the perimeter of this cell is really the same as following the perimeter of the hexagonal cell on the right of the arrow in Figure 2.12.

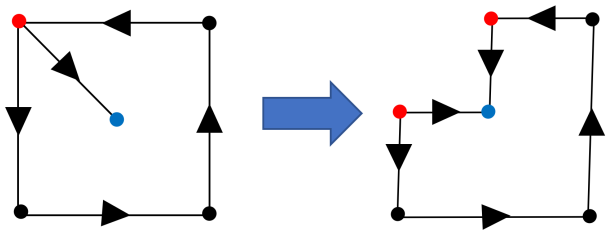


Figure 2.12 We calculate the index of the cell on the left by first copying the node and edge that the leaf is adjacent to. This yields the cell on the right. We then calculate the cell index as usual.

This new cell is obtained by making "copies" of the node and edge that are adjacent to the leaf. Hence, the two red nodes in the hexagonal cell represent the same node as the red node on the left. Similarly, both of the edges in the cell on the right with one red node and one blue node represent the same edge from the left graph. Meanwhile, we leave every other edge and node unchanged. So, for example, the topmost edge in both graphs is the same.

Thus, calculating the index of the cell with the leaf is equivalent to calculating the index of the hexagonal cell on the right.

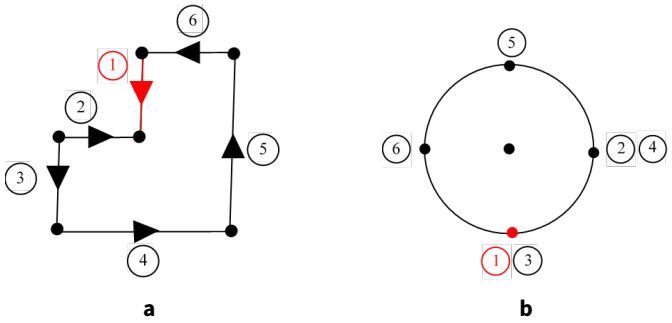


Figure 2.13 As before, we number the edge vectors of the cell, and mark the spots where they point on a circle.

As before, we'll need to follow the vectors counterclockwise around the cell, and mark their directions on a circle. This is shown in Figure 2.13. However, because this cell would not be convex if it was embedded in the plane, we'll need to use one of its completions to determine how exactly to traverse the circle. The main difficulty in this example is determining how to get from point 1 to point 2. So, we turn to a completion to inform our

decision. One such completion is shown in Figure 2.14.

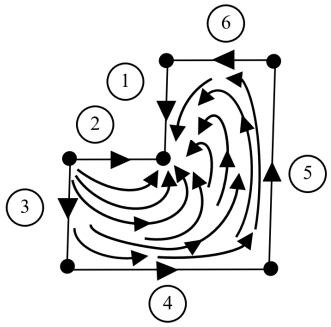


Figure 2.14 A completion of the cell in Figure 2.12. Note how, as we move from edge 1 to edge 2, the completion vectors would begin pushing us down, then towards the upper-left, and then to the right.

Notice how, when moving along edge 1 towards edge 2, the vectors on the cell's interior would first push us down, and then to the upper left. This is because the vectors in the upper right and the lower left cancel out as we move downward. Then, as we near edge 2, we would start being pushed more to the right. This indicates that, from point 1 on Figure 2.13b, we would turn towards the upper left, and then right (i.e. clockwise) until we reach point 2. This results in the traversal shown in Figure 2.15.

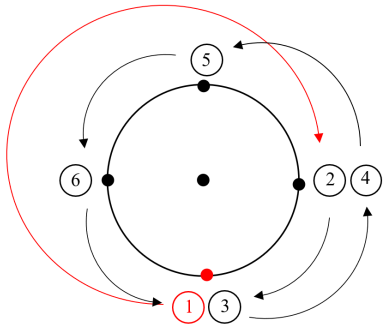


Figure 2.15 This figure shows how we follow the markings enumerated in Figure 2.13b. Since the completion indicated that we would be pushed down, then to the upper left, and then to the right, we should go *clockwise* from point 1 to point 2. This is shown by the red arrow. We then move from 2 to 3 clockwise, then from 4 to 5 to 6 counterclockwise. Hence, we make no complete net rotations around the circle, resulting in an index of 0.

Thus, we see that we make no complete rotation around the circle. So,

this hexagonal cell— and therefore the cell with the leaf— has an index of 0.

For arguments later in this thesis, it will be useful to distinguish cells with leaves (as in Figure 2.12) from those without leaves (like all of the cells but the fourth in Figure 2.9). Thus, we introduce the following definitions:

Definition 2.2.7. A graph N is an **n-gon cell** if the edges and vertices of N form a polygon.

Definition 2.2.8. A graph L is a **leaf cell** if L is an n-gon cell with at least one extra leaf node added to it.

By convention, we draw the leaf nodes inside the area within the polygonal cell boundary. For this reason, we call the leaves of a leaf-cell *internal leaves*.

By constructing our vertex and cell indices in this manner, we closely mimic the continuous index definition (which applies to vector fields of any dimension). However, our current method for calculating discrete indices is complicated. We must first record the direction in which the vectors near the node/around the cell point, and then follow these directions to see how many times they trace out a complete counterclockwise rotation. But does it really need to be this complex? Perhaps in the case of higher dimensional manifolds, but certainly not for two-dimensional objects like graphs.

In the next section, we take advantage of our problem's simplicity by creating simpler, equivalent definition for our vertex and cell indices.

2.3 The Discrete Index: a Simpler Approach

Let's return to the motivational story we gave for the index to see how we might simplify our indices. Recall that you and I are standing in a particularly windy meadow. You stand at a place where the wind doesn't blow, and I walk around you. As I walk, you see the wind pushing and pulling me in different directions. How can we quantify what the wind is doing to me? Instead of recording where the wind pushes me, as we did before, what if we instead just count how many times it pushes me in different directions? Specifically, let's just count the number of times the wind pulls me towards you, and then pushes me away from you, or vice versa.

In general, we can calculate the index of a node by first observing how many times the vectors that touch the node change directions, as we follow the vectors counterclockwise around the node.

Example 2.3.1. Let's return to the vector field zero we saw from Example 2.1.4a, which is reproduced below.

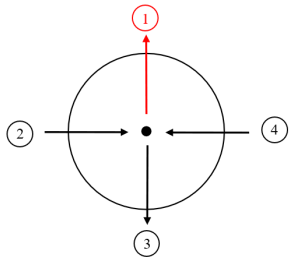


Figure 2.16 A node in a graph with a discrete vector field. As we move counterclockwise from vector 1 (chosen arbitrarily), the vectors around the node change direction four times.

Here, we will imagine this zero as being a node in a graph with a discrete vector field (instead of a zero in a continuous vector field, as we supposed before). Moving from vector 1 to vector 2, the direction of the vectors changes from going out of the node to going into the node. Hence, this is one direction change. Similarly, moving from vector 2 to vector 3, the vectors go in and then out of the node. This is another direction change. As we move counterclockwise from vector 1 to 2 to 3 to 4 and back to 1, we encounter four total direction changes.

We formally define a direction change around a vertex like so:

Definition 2.3.2. A **vertex direction change** occurs at a vertex z whenever one edge vector incident on z points into z, and an adjacent edge vector points out of z (or vice versa).

Note that the number of vertex direction changes does not correspond to the index calculated before, as this node's index was originally calculated to be -1, not 4. There is, however, an equation that converts the number of direction changes to the index that we previously calculated. Specifically:

Definition 2.3.3. Let z be a node in a graph with a discrete vector field, and let d_z denote the number of vertex direction changes around z. Then, we define:

$$\text{index}(z) = 1 - \frac{d_z}{2}.$$

As it turns out, this index is always an integer because the number of direction changes is always even. This is justified below:

Theorem 2.3.4. *The number of direction changes around a node z in a discrete vector field is always even.*

Proof. We prove this by inducting on the number of vectors that are incident on z. For our base case, suppose that there were two vectors incident on z. Then, if both vectors pointed inward, or both vectors pointed outward, there would be zero direction changes. Meanwhile, if one pointed in and one pointed out, there would be two direction changes. This is because if we start from the vector going in and move counterclockwise, we go in, then out, and then in again. Hence, the number of direction changes is even in this case.

Now, for our induction hypothesis, assume that the number of direction changes is even if there were $n \in \mathbb{N}$ vectors incident on z.

For our induction step, suppose that there were $n+1$ vectors $v_1, ..., v_{n+1}$ incident on z and consider a fixed v_j. By our induction hypothesis, there would be an even number of direction changes among the vectors incident on z if v_j was not there. Note that the only differences between this scenario and the one where v_j is incident on z are the possible direction changes from v_{j-1} to v_j and from v_j to v_{j+1}. From here, there are two cases:

(a) Suppose that v_j is between two vectors that go in different directions. Hence, there would be a direction change between these vectors if v_j was not incident on z. So, there is one less direction change among the vectors not including v_j. Thus, by our induction hypothesis, there is an odd number of direction changes between the vectors $v_1, ...v_{j-1}$ and $v_{j+1}, ..., v_{n+1}, v_1$.

However, because the vectors next to v point in different directions, one points in while the other points out. So, one of these vectors points in a different direction from v_j, while the other points in the same direction. Thus, there is exactly one direction change between v_{j-1} an v_{j+1}. Adding this to the odd number of other direction changes gives us an even number, as desired.

(b) Now, suppose that the vectors next to v pointed in the same direction. Then, by our induction hypothesis, there is an even number of direction changes between $v_1, ...v_{j-1}$ and $v_{j+1}, ..., v_{n+1}, v_1$.

If v_j pointed in the same direction as its neighbors, then there would be no direction changes between v_{j-1} and v_{j+1}. So, there would still be an even number of direction changes.

Meanwhile, if v_j pointed in the opposite direction to its neighbors, then there would be a direction change from v_{j-1} to v_j and from v_j to v_{j+1}. Hence, there would be two additional direction changes, and the total number of direction changes would still be even.

Therefore, the number of direction changes is even in either case. So, the number of direction changes is even for any natural number of vertices leaving the node, as desired. □

Thus, this index is always an integer.

Example 2.3.5. Let's confirm that this definition coincides with our previous idea for the node index by considering the node from Example 2.16. We originally calculated that it's index was -1 in Example 2.1.4. We've just seen that the vectors around the node have four direction changes, and $1 - \frac{4}{2} = -1$. So, the indices coincide in this example. This appears to be true in general.

The fact that Definition 2.3.3 matches the node index from Section 2.2 is not a coincidence. Recall that the old index counted the number of complete counterclockwise rotations the vectors made around the node. When calculating our new index, we visit each vector as we move once counterclockwise around the node. This counterclockwise rotation can be thought of as contributing +1 to our index. The first direction change we encounter as we move around the node indicates that the vectors now point roughly 180° in the opposite direction (i.e. moving back in the clockwise direction). Hence, once we reach the second direction change of the pair, we would have made one complete clockwise rotation. Therefore, we deduct 1 from the index each time we encounter a pair of direction changes. Thus, once we reach the first vector from which we started, the index will be $1 - \frac{d}{2}$, where d is the number of direction changes.

Let's now turn to simplifying the cell index. Recall that the way we defined the cell index was similar to the way we defined the node index. The only difference is that we considered the vectors on the cell's edges, instead of the vectors leaving nodes. Therefore, when simplifying the cell index, we should consider the number of times the vectors on the cell's edge change directions as we move around the cell's perimeter. Here, "change directions" refers to the vectors changing from pointing in the counterclockwise direction to clockwise, or vice versa.

Equivalently, a cell direction change occurs whenever consecutive vectors exit the same node or enter the same node. For example, vectors 1 and 2

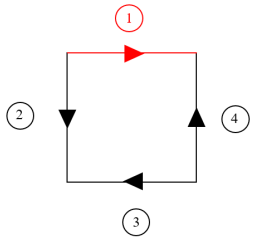


Figure 2.17 A cell in a graph with a discrete vector field. The vectors on the cell's edges change direction four times.

in Figure 2.17 exit the same node (in the top left corner), as do vectors 3 and 4 (in the bottom right). So, both instances count as a direction change. Similarly, vectors 2 and 3 enter the node in the lower left corner, and vectors 1 and 4 enter the same node in the upper right. Hence, the cell has four direction changes. Compare this to a cell whose vectors cycle around the boundary, as in the first cell in Figure 2.10; such a cell has no direction changes.

Example 2.3.6. By representing leaf-cells as n-gon cells, as we did when calculating the old cell index, it is clear that there is a cell direction change at the internal leaf (the blue node). This would also be true if the vector on the edge incident to the leaf pointed in the opposite direction. Thus, in general, there will always be a cell direction change occurring at each internal leaf of a leaf-cell.

That cell direction change, plus the one occurring at the left-most red node in the n-gon representation, means that the cell has two cell direction changes. Thus, its index is zero, as we stated in Example 2.2.6.

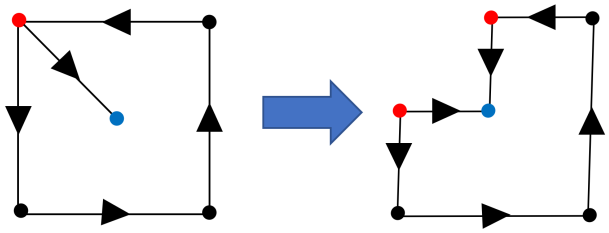


Figure 2.18 As the n-gon representation on the right shows, every leaf-cell will have one cell direction change occurring at each of its internal leaves.

Therefore, we can define a cell direction change like so:

Definition 2.3.7. A **cell direction change** occurs when the vectors of two adjacent edges on a cell both point into or out of the same vertex.

Using this concept, our cell index is equivalent to the following simpler interpretation:

Definition 2.3.8. Let C be a cell in a graph with a discrete vector field, and let d_C denote the number of cell direction changes around C. Then,

$$\text{index}(C) = 1 - \frac{d_C}{2}.$$

Hence, using a slightly different interpretation of "direction changes", our new node and cell index equations are the same! By inducting on the number of edges along the cell's perimeter and using a similar argument to the one used in the proof of Theorem 2.3.4, we find that the number of cell direction changes is also always even. So, our discrete cell index is also always an integer.

In the next chapter, we discuss our plans to use our simpler index definitions to prove a discrete Poincaré-Hopf theorem that applies to certain types of graphs.

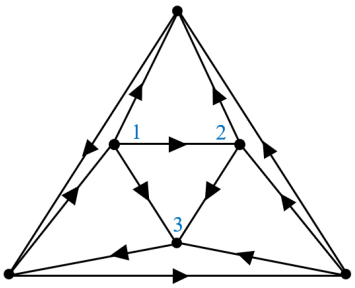
Chapter 3

Narrowing the Scope of our Discrete Theorem

3.1 Problems at the Boundary

With our indices defined, our next step in discretizing the Poincaré-Hopf theorem to determine if the sum of the indices for any graph remains constant. Recall that, for now, we will only consider a graph's *internal* nodes and cells: i.e. those that do not touch the graph's boundary[1]. For example, in Figure 3.1, we will focus on the vertices labeled 1 through 3, and the middle triangle. Our reasons for doing so will become clear soon.

Examining the figure, we see that the indices of the internal nodes and cell sum to -1.

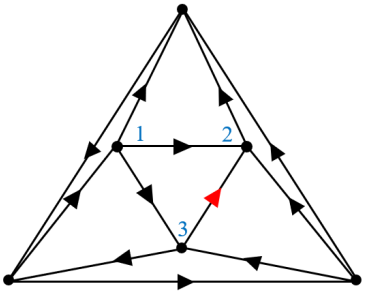


Node/cell	index
1	0
2	-1
3	0
Inner cell	0
Total	-1

Figure 3.1 A graph whose internal nodes and cell (the middle triangle) have indices that sum to -1.

[1]Hence, we exclude all trees from this discussion.

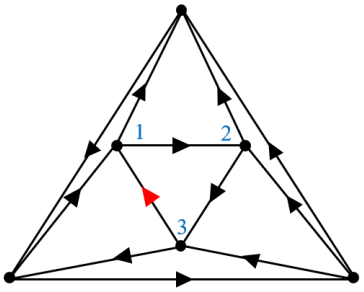
Now, suppose that we replace the discrete vector field on this graph with a slightly different vector field, as in Figure 3.2. While the two vector fields differ because of the red arrow in the second figure, the sum of the graph's indices is still -1. Perhaps this holds in general: maybe the sum of a graph's indices is an invariant property of the graph, just as the sum of the indices was for a manifold.



Node/cell	index
1	0
2	0
3	-1
Inner cell	0
Total	-1

Figure 3.2 The same graph as in Figure 3.1, but with a slightly different discrete vector field. The sum of the graph's internal nodes and vertices is still -1.

Unfortunately though, this is not the case. Suppose we instead flipped the red arrow shown in Figure 3.3. Here, the sum of the indices is +1, not -1. Furthermore, we reach similar inconsistencies when we have the vector fields shown in Figures 3.4 and 3.5.

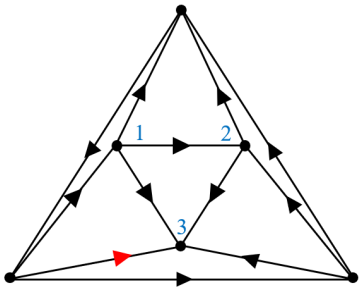


Node/cell	index
1	0
2	0
3	0
Inner cell	+1
Total	+1

Figure 3.3 The same graph as in Figure 3.1, but with an internal cell cycle around the middle cell. Because of this cycle, the graph's index sum is +1, not -1.

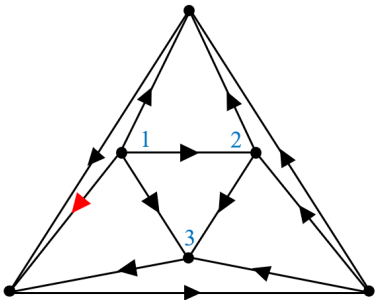
What's causing these inconsistencies? As it turns out, these issues are respectively caused by the discrete vector fields' sinks, sources, and *internal cell cycles*. Recall that a *cycle* in a graph is a path that starts and ends at the same node.

Definition 3.1.1. A **cell cycle** is a cycle on the boundary of a cell in a graph with a discrete vector field. An **internal cell cycle** is on one of the graph's internal cells.



Node/cell	index
1	0
2	0
3	+1
Inner cell	0
Total	+1

Figure 3.4 The same graph as in Figure 3.1, but with a sink node (node 3). Because of this sink, the graph's index sum is different from when there were no sinks, sources, or cycles.

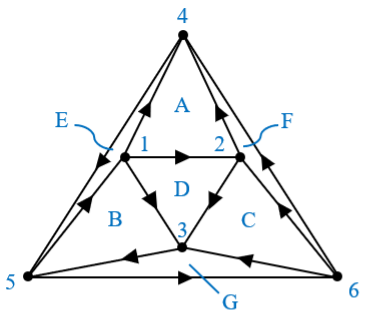


Node/cell	index
1	+1
2	0
3	0
Inner cell	0
Total	+1

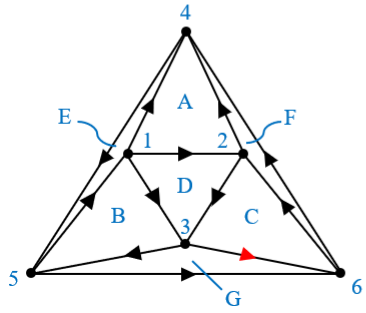
Figure 3.5 The same graph as in Figure 3.1, but with a source node (node 1). Again, the sum of the indices is different from when there were no sinks, sources, or cycles.

In Figure 3.3, we created a cell cycle on the middle cell. Imagine what this means for the graph's completion. Instead of vectors flowing from left to right on the top of the triangle, they now flow from right to left. This then alters any vectors in the topmost triangle. Because we aren't considering the topmost triangle in our calculations (since it's not a internal cell), we aren't accounting for this change. Similarly, both the sink and source examples alter information about the boundary cells. For instance, both examples alter information about the bottom-most cell. This suggests that we need to account for the nodes and cells that touch the boundary.

Why not just include the indices for the nodes and cells that touch the boundary then? Unfortunately, this also leads to inconsistencies. Consider Figure 3.6.



Node	index	Cell	index
1	0	A	0
2	-1	B	+1
3	0	C	0
4	0	D	0
5	-1	E	+1
6	0	F	0
		G	+1
Node Total	-2	Cell Total	+3
		Total index sum: +2	



Node	index	Cell	index
1	0	A	0
2	-1	B	0
3	0	C	+1
4	0	D	0
5	-1	E	+1
6	0	F	0
		G	0
Node Total	-2	Cell Total	+2
		Total index sum: 0	

Figure 3.6 Above: the same graph as in Figure 3.1. Below: the same graph with a slightly different vector field. The nodes are again labeled using numbers, while the cells are labeled using letters. Summing the indices of *all* the nodes and cells (including boundary nodes and cells) of each graph yields different index sums.

On top is the same graph as in Figure 3.1, but with the indices of *all* the cells and nodes calculated. As the table shows, these indices sum to +2. Meanwhile, altering the discrete vector field as in the lower graph results in a sum of 0. So, the sum of the indices is not constant using our index definitions on the boundary nodes and cells.

This could again be explained by considering the graph's completion. While it is clear how the underlying continuous vector field on a graph behaves around the internal nodes and cells (since the discrete vector field surrounds these elements), the same cannot be said for the boundary cells and nodes. This is because our idea of a completion only extends *within* the

graph in question; a completion is not defined outside of the graph. Because the boundary nodes and vertices are adjacent to a region where the graph's completion is undefined, we can't accurately assess how the vector field behaves around these elements.

In the next section, we discuss our plans to reconcile the boundary node/cell issue.

3.2 Handling Boundary Cells and Nodes

An index is supposed to quantify the behavior of a vector field around a node or cell. As we showed in the previous section however, our discrete index definitions don't do this well for a graph's boundary cells and nodes. We could take this to mean that our index definitions are poor: that we should instead try to create another index. However, the fact that a graph's internal node and cell index sum is constant suggests otherwise. Instead of abandoning our index definitions, we can instead try modifying our approach to boundary nodes and cells. Our indices didn't work for these elements because we lacked information about what was happening at all areas around them. One way to resolve this is by *introducing* information around the boundary nodes and cells in a way that doesn't affect the graph's discrete vector field.

Namely, if G was our original graph, we can construct a new graph G^+ that's defined as follows:

Definition 3.2.1. If G is a directed, connected, planar graph, then we define $\mathbf{G^+}$ to be the graph obtained by adding edges that emanate out of G's boundary nodes [2] and into a new external sink node.

An example of this is shown in Figure 3.7.

By introducing external edges and a sink node, we acquire just enough information to assess what's happening around the boundary nodes and cells, while minimally affecting the internal parts of the graph. What's more, by introducing the sink node and new edges, the sum of the cell and node indices (including the sink node's index) *is* constant— even if there are sinks, sources, or cycles. Consider Figure 3.8, which shows the graph on the right of Figure 3.7 with two slightly different vector fields. In both cases, the sum of the indices is +2, even though node 3 in the second graph is a sink node.

[2] Since we draw the leaf nodes of a leaf-cell within the leaf-cell's boundary, we would not add extra edges to these nodes.

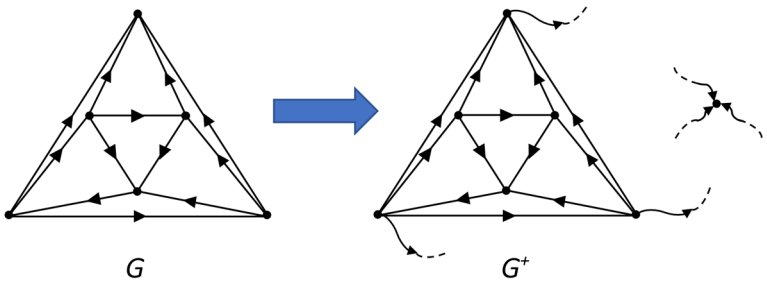
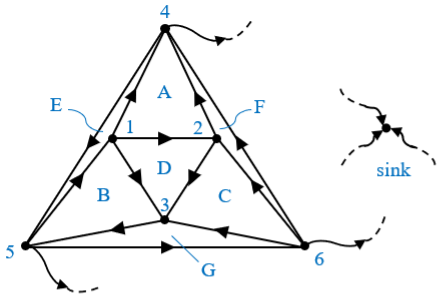
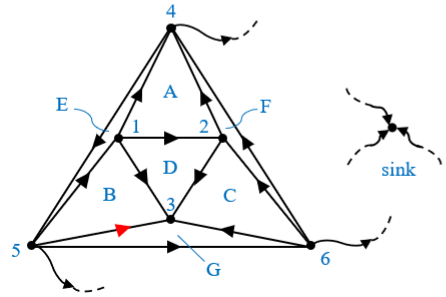


Figure 3.7 On the left is the same graph, denoted G, from Figure 3.1. On the right is G^+, the same graph as G, but with new edges that extend from G's boundary nodes to a new external sink node.



Node	index	Cell	index
1	0	A	0
2	-1	B	+1
3	0	C	0
4	0	D	0
5	-1	E	+1
6	0	F	0
Sink	+1	G	+1
		Extra cells	0
Node Total	-1	Cell Total	+3
		Total index sum: +2	



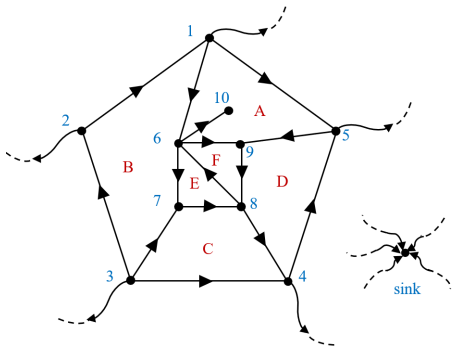
Node	index	Cell	index
1	0	A	0
2	-1	B	0
3	+1	C	0
4	0	D	0
5	0	E	+1
6	0	F	0
Sink	+1	G	0
		Extra cells	0
Node Total	+1	Cell Total	+1
		Total index sum: +2	

Figure 3.8 The index sum remains constant when we add extra edges and a sink node, even if the discrete vector field changes. "Extra cells" denotes the new cells formed by adding a sink node and extra edges.

Note that, by adding the extra edges and sink node, we also create some new cells. However, those cells will always have an index of zero. This is

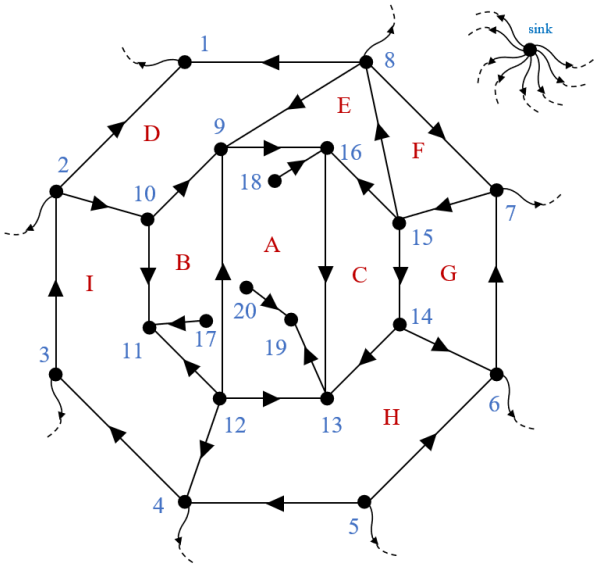
because these cells will always only have three sides, two of which have vectors that point into the sink.

Interestingly, this is not the only graph whose indices sum to +2 once we add these new edges and sink node. Figures 3.9 and 3.10 show more examples of graphs whose index sums are +2 once these new elements are added.



Node	index	Cell	index
1	0	A	-1
2	0	B	0
3	+1	C	0
4	0	D	+1
5	-1	E	+1
6	-1	F	+1
7	0		
8	-1		
9	0		
10	+1		
Sink	+1	Extra cells	0
Node Total	0	Cell Total	+2
		Total index sum: +2	

Figure 3.9 Another example of a graph with a polygonal boundary and extra edges, sink node that's index sum is +2.



Node	index	Cell	index
1	0	A	-2
2	0	B	-2
3	0	C	0
4	0	D	-1
5	+1	E	0
6	0	F	+1
7	-1	G	+1
8	0	H	-2
9	0	I	0
10	0		
11	+1		
12	+1		
13	0		
14	0		
15	0		
16	0		
17	+1		
18	+1		
19	+1		
20	+1		
Sink	+1	Extra cells	0
Node Total	+7	Cell Total	-5
		Total index sum: +2	

Figure 3.10 Yet another example of a graph with a polygonal boundary, extra edges, and an extra sink node that's index sum is +2.

As it turns out, if G is a graph with a polygonal boundary, then G^+ will *always* have an index sum of +2, even if we change the discrete vector field on G (i.e. not the vectors on the extra edges). This is stated as a theorem and proven in the next section.

Why is this result relevant to our goal of creating a discrete version of the Poincaré-Hopf theorem? Recall that, according to the Poincaré-Hopf theorem, if we put vectors on a manifold, then the manifold's index is constant, even if the vectors change. According to this new result, if we put vectors on a graph with a polygonal and add a few edges and a sink node, then the graph's index sum is constant, even if the vectors on the original graph change. Thus, this result mirrors the Poincaré-Hopf theorem.

Chapter 4

A Discrete Analogue

In this chapter, we prove a discrete analogue to the Poincaré-Hopf theorem. Specifically, we prove that for certain types of directed graphs, the sum of the vertex and cell indices is constant, regardless of where the discrete vectors point. This theorem is formally presented below:

Theorem 4.0.1 (Index Sum Invariant)**.** *Let G be a finite, directed, connected, planar graph with a polygonal boundary. Then, the index sum of G^+ is always* +2*.*

We will prove this theorem by first proving a few lemmas and propositions. Specifically, Lemma 4.1.1 will establish that, if N is an n-gon cell, then the index sum of N^+ is +2. From there, we will show that the following operations preserve the a graph's index sum:

(1) Adding an internal leaf to a cell graph

(2) Adding an internal edge to a directed, connected, planar graph with a polygonal boundary, extra edges, and a sink node

These two operations will follow as propositions from Lemma 4.1.4.

Once we've proven these results, we will show that we can build G^+ step-by-step so that, at each step, the graph's index sum is +2. Specifically, we'd start out with an n-gon graph N that's polygonal boundary was the same as G's polygonal boundary. Then, we'd sequentially perform operations (1) and (2) on N^+ to generate G^+'s edges and vertices. Since performing operations (1) and (2) on a graph does not alter its index sum (as we will show), we can ensure that G^+ and N^+ have the same index sum. By Lemma 4.1.1, this will imply that G^+'s index sum is +2.

4.1 Useful Lemmas and Propositions

We begin by proving that the index sum of an n-gon graph with extra boundary edges and sink node added has an index sum of +2. This is useful because these are the graphs we start from in our proof of the Index Sum Invariant Theorem. From them, we can build any finite, directed, connected, planar graph with a polygonal boundary.

Lemma 4.1.1. *Let N be a directed graph consisting of n edges and vertices that form an n-gon. Then, the index sum of N^+ is +2, for all integers $n \geq 3$.*

Proof. Let E denote the set of extra edges added to N to construct N^+, and let s denote the extra sink node added. Lastly, let $2x$ ($x \geq 0$) denote the number of cell direction changes N had. [1]

Note that, if $x = 0$, then N's index is +1, and each of the vertices on N has an index of 0. As discussed in Section 3.2, the cells in N^+ that aren't in N (i.e. the triangular cells that have the sink node as one of their vertices) also each have an index of 0. So, since index(s) = +1, the total index sum is index(s) + index(N) = +2 in this case.

Now, suppose that $x > 0$. Then, there exists x vertices such that both of the vectors incident on the vertex point out of it. Since the edges in E that are incident on these vertices also have vectors that point of the vertex, these vertices are sources. Thus, the combined index of these vertices is $+x$.

Every other vertex on N falls into one of two cases:

(i) Suppose v was one of the other x vertices at which a cell-direction change within N occurred. Then, that cell direction change must have been an "in" direction change. Since the edge from E that's incident on v has a vector pointing out of it, yet the vectors from N that are incident on v point in, index(v) = 0.

(ii) Suppose v was not one of the other x vertices at which a cell-direction within N occurred. Then, one edge from N has a vector pointing into v, and the other incident edge from N has a vector pointing out. So, regardless of the direction that the vector on the incident edge from E points, index(v) = 0.

Hence, every vertex in the graph other than the x sources and s have an index of zero. So, the sum of all the node indices in the graph is $+x+$index(s) = $x+1$.

[1] Recall that the number of cell direction changes must be even. So, this will account for every possible number of direction changes.

Next, we calculate the cell indices in our graph. As noted in Section 3.2, each of the extra cells created by adding s and the edges from E will have an index of zero. Thus, N's index is the only cell index that contributes to the total index sum. Since N has $2x$ cell direction changes:

$$\begin{aligned}\text{index}(N) &= 1 - \frac{2x}{2} \\ &= 1 - x.\end{aligned}$$

Therefore, our graph's total index sum is:

$$(1 - x) + x + 1 = +2.$$

Thus, the index sum of N^+ is +2, as desired.

□

Now, we must prove that neither adding an internal leaf to a cell graph (operation (1)) nor adding an internal edge to a finite directed, connected, planar graph with a polygonal boundary (operation (2)) will alter the graph's index sum.

At first, this task seems daunting, as there are many factors to consider: the direction of the added edge, the directions of the edge vectors around it, the other vectors in the surrounding cell, etc. However, we can simplify the problem greatly by realizing that only some of these factors change once we add the edge/internal leaf. Specifically, when we add an edge, the entire graph will not change: only the vertex that we add the edge to, and the cell that the edge alters will change. Thus, we only need to examine the *local* changes that operations (1) and (2) make to a graph.

In this spirit, the following definition will be useful to us:

Definition 4.1.2. Let v be a vertex in a graph H and let G be a supergraph of H. We define the integer $\boldsymbol{\Delta_{H,G}(v)}$ to be the number of *cell* direction changes that occur at the corner where v is in G minus those that occur at the corner where v is in H.

Example 4.1.3. To clarify this concept, let's consider Figure 4.1, where we add the red edge e to the graph H to create the new graph G. The blue arcs show the cell direction changes that occur at v.

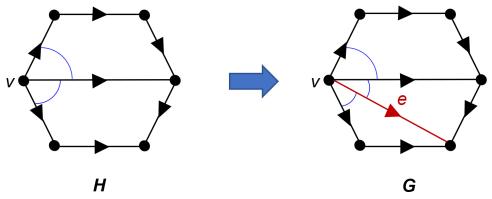


Figure 4.1 By adding e to H, we eliminate the cell direction change that occurred at v in the lower cell, and create two new cell direction changes. Thus, $\Delta_{H,G}(v) = +1$.

The vectors incident on v in the upper trapezoid of H both point out of v, as do the ones incident on v in the lower trapezoid of H. So, there are two cell direction changes at v in H, as indicated by the two blue arcs. However, when we add e to create G, we split the direction change in the lower trapezoid, and create two new cell direction changes on either side of e at v. So, the number of cell direction changes at v in G is 3. Thus, $\Delta_{H,G}(v) = 3 - 2 = +1$.

It's important to realize that we didn't need to know the directions of every vector around v to calculate this number. Specifically, we never needed to know that there was another cell direction change at v in the upper trapezoid cell in H, because this cell direction change never altered. We only needed to know that e eliminated one cell direction change at v, and that two new cell direction changes formed. Thus, to calculate $\Delta_{H,G}(v)$, we only need to consider the parts of H and G that differ.

The following lemma associated with this term may initially appear to be unrelated to operations (1) and (2). However, it will play an integral role in proving that these operations preserve a graph's index sum.

Lemma 4.1.4. *Let v_1 be a vertex in a cell N within a finite, directed, connected, planar graph H that has a polygonal boundary. Now, let G be a supergraph of H that's constructed by either*:

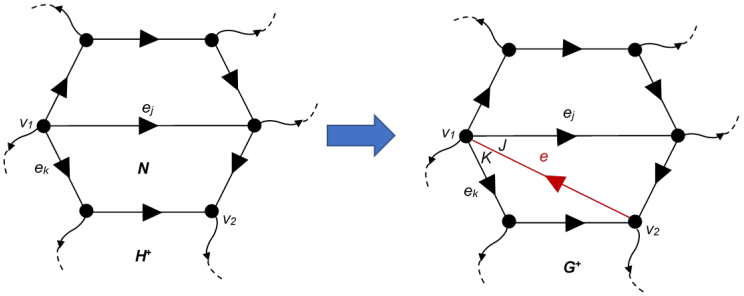
(1) *adding an edge between v_1 and some other non-adjacent vertex in N, or*

(2) *adding a new internal leaf that's edge is incident on v_1.*

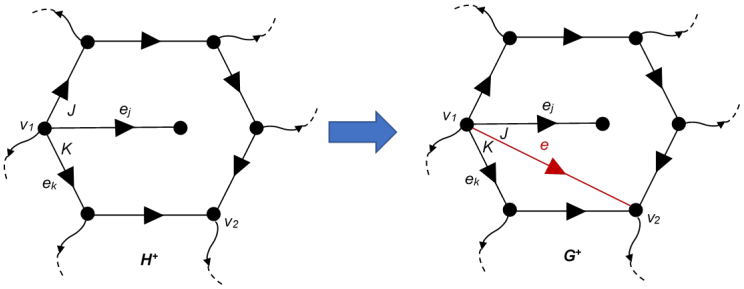
Next, let a_1 and b_1 be the number of vertex direction changes around v_1 in G^+ and H^+ respectively. Then, $\Delta_{H^+,G^+}(v_1) + (a_1 - b_1) = 1$.

In other words, when we add an edge to v_1, the net number of cell direction changes at v_1 that result is always compensated by the net change in vertex direction changes at v_1.

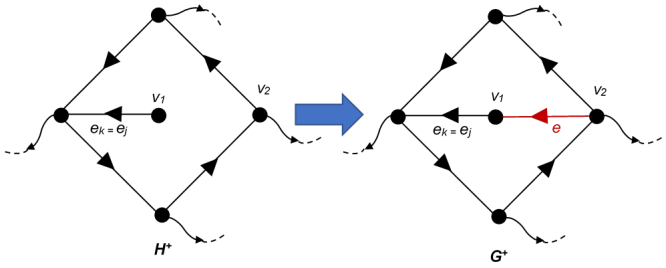
Proof. Let e_K and e_J denote the two edges incident on v_1 in N that e is between. Also, let J be the corner between e_J and e and let K denote the corner between e and e_K. (If v_1 was an internal leaf, then e_J and e_K would be the same edge and J and K are the corners on either side of v_1.) An example of these elements is given in Figure 4.2.



a. An example where N is the lower trapezoid cell within the larger graph H.



b. In this example, $N = H$, the leaf-cell.

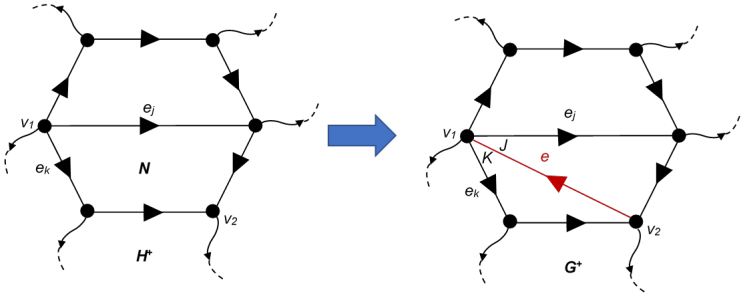


c. In this example, $N = H$, the square cell. Here, v_1 is a leaf, e_J and e_K are the same edge, and J and K are the corners on either side of v_1.

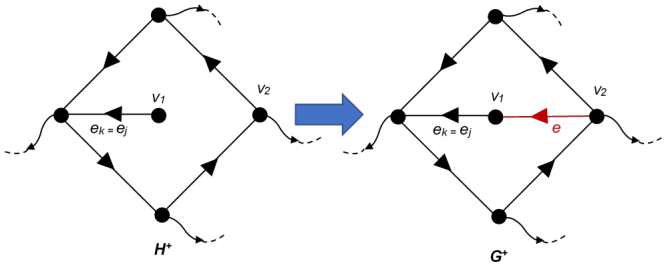
Figure 4.2 Examples of graphs to which Lemma 4.1.4 applies (all sink nodes were omitted for clarity). In each case, v_1 has an edge added to it.

We will prove this lemma by examining how the vertex and cell direction changes at v_1 change as we transform H^+ to G^+. Namely, when we add e to H^+, we potentially modify the number of vertex direction changes that occur at v_1 and the number of cell direction changes in N that occur at v_1. Because e was only added to the cell N, no other cell direction changes at v_1 in H are affected. Thus, $\Delta_{H^+,G^+}(v_1)$ will be completely determined by the change in cell direction changes at v_1 as we split the corner in N at v_1.

(i) First, consider the case where e eliminates a cell direction change in N that occurred at v_1, thus changing v_1's index. Figure 4.3 shows this.



a. The edge e introduced two vertex direction changes at corners J and K. Since there were no vertex direction changes at v_1 in N, the net result is a gain in two vertex direction changes around v. Meanwhile, e eliminated the cell direction change at v_1 in N. Since there are no cell direction changes at corners J and K, there is a net loss of one cell direction change.



b. In H^+, there were no vertex direction changes at v_1. In G^+ though, there are two vertex direction changes at v_1: one from e_K/e_J to e, and one from e to e_K/e_J. Also, while there was a cell direction change at v_1 in H^+, there are no cell direction changes at v_1 in G^+.

Figure 4.3 Example graphs for case (i) (their sink nodes were omitted).

Because e eliminated a cell direction change at v_1 in N, e_J and e_K either both pointed into or out of v_1. Since v_1's index changed, e must have pointed in the opposite direction of these vectors with respect to v_1 (e.g. if e_J and and e_K's vectors pointed into v_1, then e's pointed out, or vice versa). Otherwise, there would be no vertex direction changes between e_K and e_J before or after e was added, and index(v_1) would not have changed.

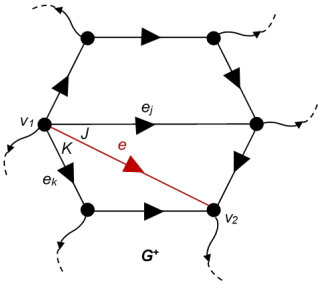
Thus, e introduced two vertex direction changes at v_1 (as shown in Figure 4.3). Therefore, $a_1 - b_1 = +2$.

Furthermore, since e's vector points in the opposite direction of both e_K and e_J's vectors with respect to v_1, there cannot be any cell direction changes at K or J. However, e eliminated the cell direction change in N that occurred at v_1. So, the net new cell direction changes created at v_1 as we transform H^+ to G^+ is -1. Thus, $\Delta_{H^+,G^+}(v_1) = -1$.

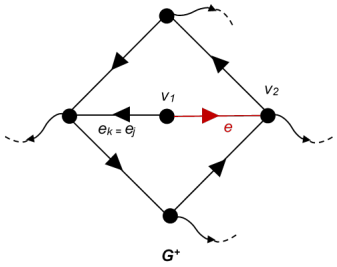
Therefore:

$$\Delta_{H^+,G^+}(v_1) + (a_1 - b_1) = -1 + 2 = 1.$$

(ii) Now, suppose that e eliminates a cell direction change in N that occurred at v_1, but v_1's index did not change. Figure 4.4 shows this.



a. As before, N was the lower trapezoid cell in H^+, which was eliminated when we added e. In both H^+ and G^+, there were no vertex direction changes between e_J and e_K. So, $a_1 = b_1$. Furthermore, while there was one cell direction change at v_1 between e_J and e_K in H^+, there are two in G^+.



b. There were no vertex direction changes at v_1 in H^+ because v_1 only had one incident edge. Also, adding e to create G^+ made no vertex direction changes. So, $a_1 = b_1$. Also, adding e created a cell direction change on either side of v_1.

Figure 4.4 Example graphs for case (ii) (their sink nodes were omitted for clarity). In these examples, the number of vertex direction changes at v_1 stayed the same. Meanwhile, adding e eliminated the cell direction change between e_J and e_K in N^+, but added one cell direction changes in both J and K.

Since there was a cell direction change at v_1 in N, e_K and e_J must point in the same direction with respect to v_1. So, since v_1's index did not change after adding e, e's vector must point into v_1 if e_K and e_J did, or point out of v_1 if e_K and e_J did. This implies that there is a cell direction change at both K and J. However, e also eliminated a cell direction change in N at v_1. Therefore, $\Delta_{H^+,G^+}(v_1) = +2 - 1 = 1$. So:

$$\Delta_{H^+,G^+}(v_1) + (a_1 - b_1) = 1 + (0) = 1.$$

(iii) Lastly, suppose that e did not eliminate a cell direction change in N at v_1. An example of this is shown in Figure 4.5.

So, either e_K's vector pointed into v_1 and e_J's pointed out, or vice versa. Thus, there was a vertex direction change between e_K and e_J in N, and therefore in H^+.

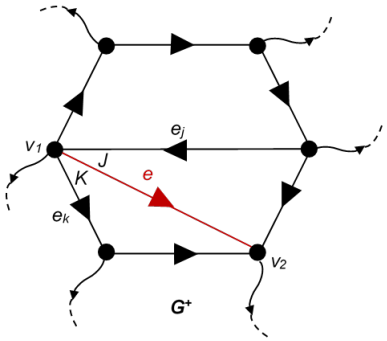


Figure 4.5 An example of a graph where case (iii) applies (its extra sink node was omitted for clarity). Notice how the vertex direction change between e_J and e_K in H^+ was replaced with a vertex direction change between e_J and e. Also, a cell direction change was created between e_K and e.

In G then, there must be exactly one vertex direction change between e_K, e, and e_J. This is because e's vector can only point in the same direction as either e_K's or e_J's vector.

So, there was one vertex direction change between e_J and e_K in H^+, and there is exactly one vertex direction between these edges in G as well. Therefore, there are no net new direction changes at v_1 and $(a_1 - b_1) = 0$.

Let's now examine how the cell indices differ. Adding e must have created exactly one cell direction change at either J or K. This is

because, regardless of what direction e's vector pointed in, it must have the same direction as either e_K's or e_J's vectors, since those vectors pointed in opposite directions. So, if e_K's vector is the one that agrees with e's vector, then K will have a cell direction change. Since there were no cell direction changes at v_1 before we added e, and there is now one cell direction change, $\Delta_{H^+,G^+}(v_1) = +1$. Thus:

$$\Delta_{H^+,G^+}(v_1) + (a_1 - b_1) = 1 + (0) = 1.$$

So, in all scenarios, adding e to v_1 results in $\Delta_{H^+,G^+}(v_1) + (a_1 - b_1) = 1$, as desired. □

With this lemma, we can now prove that performing operations (1) and (2) on a graph each leave the graph's index unchanged. We begin by proving operation (1) as the following proposition.

Proposition 4.1.5. *Let N be a cell (n-gon or leaf), and let G be the cell graph obtained by adding an internal leaf to N. Then, G^+ has the same index sum as N^+.*

Proof. Let r be the internal leaf and let e be the edge it is incident on. Also, let v be the other vertex incident on e. Additionally, let b be the number of vertex direction changes around v in N^+, a be the number of vertex direction changes around v in G^+. Lastly, let d_N and d_G be the number of cell direction changes in N and G respectively.

We will examine how the index sums of N^+ and G^+ differ by seeing how the indices of the vertices and cells that are locally affected by adding the internal leaf change.

When we add e to N^+, we potentially change v's index. We also replace the cell N with a new cell G, which may have a different index from N. Additionally, adding the internal leaf means that we also add the vertex r, whose index we also have to account for. Since no other parts of the graph are affected by adding e and r to N^+, these are the only indices that may differ between N^+ and G^+.

So, to show that G^+ has the same index sum as N^+, we want to prove that:

$$\text{index}(N) + \text{index}_{N^+}(v) = \text{index}(G) + \text{index}_{G^+}(v) + \text{index}(r).$$

Equivalently:

$$\left(1 - \frac{d_N}{2}\right) + \left(1 - \frac{b}{2}\right) = \left(1 - \frac{d_G}{2}\right) + \left(1 - \frac{a}{2}\right) + 1.$$

Clearing the denominator and rearranging the terms yields:

$$(d_G - d_N) + (a - b) = 2.$$

Recall that adding e to N^+ only changed the corner of N where v was. Thus, the difference between d_G and d_N should equal the number of net cell direction changes that occur at v, plus the extra cell direction change we get at r. So, by definition, the above expression is the same as:

$$(\Delta_{N^+,G^+}(v) + 1) + (a - b) = 2.$$

Therefore, proving the proposition is equivalent to showing that:

$$\Delta_{N^+,G^+}(v) + (a - b) = 1.$$

By Lemma 4.1.4, this is true. Thus, Proposition 4.1.5 holds, and adding an internal leaf to any cell with extra edges and sink node preserves the its index sum. □

Next, we prove operation (2), which is stated as the following proposition:

Proposition 4.1.6. *Let N be a cell (n-gon or leaf) within a graph H and let v_1 and v_2 be non-adjacent vertices in N. Next, let G be the supergraph of H obtained by adding a directed edge between v_1 and v_2 in N. Then, G^+ has the same index sum as H^+.*

Proof. Let e denote the extra edge we added between v_1 and v_2, and let J and K denote the cells that e divided N into. Also, for $i \in \{1, 2\}$, let b_i be the number of vertex direction changes at v_i in H^+, and let a_i be the number of vertex direction changes at v_i in G^+. Lastly, let d be the number of cell direction changes on N and let d_K and d_J be the number of cell direction changes on J and K respectively.

The proof of this proposition is very similar to the proof of the last proposition. As before, we will analyze how indices *locally* change as we transform H^+ to G^+.

Specifically, when we add e to H^+, we potentially change v_1 and v_2's indices. Additionally, we eliminate the cell N and add two new cells J and K. So, we must also examine the net effect of losing N's index, yet gaining the indices of J and K as we transform H^+ to G^+. Because no other parts of the graph are affected by adding e, these are the only indices that differ between H^+ and G^+.

So, to show that G^+'s index sum is also +2, we must show that the net change in the index sums as we transform H^+ to G^+ is zero. Hence, we must show that:

$$\text{index}(N) + \text{index}_{H^+}(v_1) + \text{index}_{H^+}(v_2) = \text{index}(J) + \text{index}(K) + \text{index}_G(v_1) + \text{index}_{G^+}(v_2).$$

Equivalently:

$$\left(1 - \frac{d}{2}\right) + \left(1 - \frac{b_1}{2}\right) + \left(1 - \frac{b_2}{2}\right) = \left(1 - \frac{d_K}{2}\right) + \left(1 - \frac{d_J}{2}\right) + \left(1 - \frac{a_1}{2}\right) + \left(1 - \frac{a_2}{2}\right).$$

Multiplying both sides by 2 and rearranging this expression yields:

$$2 = (d_K + d_J - d) + (a_1 - b_1) + (a_2 - b_2).$$

We can simplify this expression further by considering what the term $(d_K + d_J - d)$ represents. When we transform H^+ to G^+ by adding e, we leave every corner of N unchanged, except for the corners where v_1 and v_2 are. Hence, the only cell direction changes in H^+ that potentially change are those that occur at v_1 and v_2. Thus, $(d_K + d_J - d)$ is just the net change in the number of cell direction changes at v_1 plus those that occur at v_2 as we transform H^+ to G^+. Hence:

$$(d_K + d_J - d) = \Delta_{H^+,G^+}(v_1) + \Delta_{H^+,G^+}(v_2).$$

So, we want to show that:

$$2 = \Delta_{H^+,G^+}(v_1) + (a_1 - b_1) + \Delta_{H^+,G^+}(v_2) + (a_2 - b_2). \tag{4.1}$$

But by Lemma 4.1.4, $\Delta_{H^+,G^+}(v_i) + (a_i - b_i) = 1$ for $i \in \{1, 2\}$. So, Equation 4.1 is true. Thus, adding an internal edge to a directed, connected, planar graph with a polygonal boundary and extra edges and sink node does not alter the graph's index sum, as desired.

□

4.2 Proof of the Index Sum Invariant Theorem

Now that we've shown that performing operations (1) and (2) on a graph does not alter the graph's index sum, we can prove the Index Sum Invariant Theorem. The theorem is rewritten below:

Theorem 4.0.1 (Index Sum Invariant)**.** *Let G be a finite, directed, connected, planar graph with a polygonal boundary. Then, the index sum of G^+ is always* +2*.*

Proof. Let N be the subgraph of G that's comprises the vertices and edges of G's polygonal boundary. By Lemma 4.1.1, we know that N^+ has and index sum of +2.

Since G was connected, each of its internal nodes (i.e. all nodes besides those on N) must be connected to some other node in G. So, we can generate each internal node by sequentially adding them as internal leaves to N^+.

Specifically, every node in G that was connected to a node from N can be constructed by adding it in as an internal leaf. We then do the same for each node in G that was adjacent to an internal leaf, but not adjacent to a boundary node. We continue this process of adding the internal vertices of G into our new graph by based on how far away they were from the vertices on N. This process must end eventually because we assumed that G was a finite graph.

According to Proposition 4.1.5, adding each internal leaf will not change the graph's index sum. So, since we began with N^+, the graph we obtain after adding these leaves will have the same index sum as N^+, which is +2.

Now we must construct the rest of G's internal edges (i.e. the edges that were not in N). By Proposition 4.1.6, each edge addition will leave the index sum unchanged, and thus preserve the +2 index sum.

Hence, we can construct G^+ from N^+ using moves that preserve N^+'s index sum. Since N^+ had an index sum of +2, G^+ must as well, as desired. □

Thus, the Index Sum Invariant holds, and we have a discrete analogue to the Poincaré-Hopf Theorem.

In the next chapter, we see how the Index Sum Invariant theorem can be used to given an alternative proof to a recently-posed graph theory problem.

Chapter 5

An Application: The Game of Cycles

As stated in the Introduction, one of the reasons why we wanted to prove a discrete analogue to the Poincaré-Hopf theorem was because it may be used to prove other discrete results. As it turns out, we can use the Index Sum Invariant theorem to do just that. Specifically, in this chapter we will use the theorem to solve a new graph theory problem concerning the Game of Cycles.

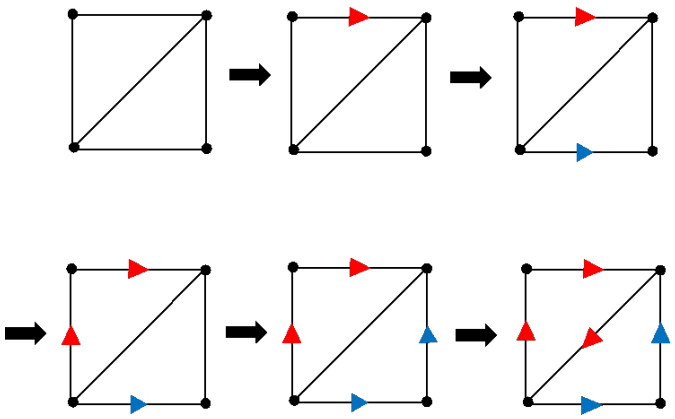


Figure 5.1 An example of the Game of Cycles. Starting with the unmarked board in the top left, the blue and red players take turns marking the edges one at a time. With each move, the players try not to: (1) create a sink or a source node, and (2) create a cell cycle. This game ended with the fully marked board on the lower right. Since the red player created a cell cycle, the red player won.

Introduced by Su (2020), the Game of Cycles is a puzzle in which players take turns marking arrows on the edges of a finite, connected *game board*: an undirected, planar graph with a polygonal boundary. The object of the game is to create a cell cycle, or to be the last player to make a move without creating any sinks (nodes whose incident vectors all point into it) or sources (nodes whose incident vectors all point out of it). An example of the gameplay is given Figure 5.1.

Alvarado et al. (2020) proved that any finite, connected board with no sinks or sources and with every edge marked must have a cell cycle. Their proof of this claim involved first detecting a cycle surrounding multiple cells, and then iteratively producing smaller cells from within. Since the graphs were assumed to be finite, this process would eventually end by discovering a cycle around a single cell, proving that the graph had a cell cycle.

Below, we provide an alternate proof that is arguably simpler; while Alvarado et al. (2020)'s proof spanned two pages and required proving two claims, our proof is only a few paragraphs.

Theorem 5.0.1 (Alvarado et al. (2020))**.** *Let G be a finite, connected game board with no sinks and no sources and such that every edge is marked with an arrow. Then G contains a cycle cell.*

We will prove this theorem by examining which of G^+'s cells and vertices contribute to this index sum.

Proof. By the Index Sum Invariant theorem, G^+'s total index sum must be +2. We know that G^+'s external sink node has an index of +1. Because G has no sinks or sources, the external sink node is the only vertex in G^+ with this index. Furthermore, since a vertex can only have an index of at most +1 (by definition), every other vertex in G^+ has an index of at most 0. So, the sum of G^+'s other vertex indices cannot be positive. Therefore, at least one of the other +1's that contribute to G^+'s index sum must come from the cell indices of G^+.

Recall that the cells in G^+ that are not in G have an index of 0, since they are each triangular cells with two edge vectors that point into the sink node. Thus, the other +1 that contributes to G^+'s index sum must come from the cells of G.

So, the sum of G's cell indices sums to +1. Since a cell can only have an index of at most +1 (by definition), there must be at least one cell in G with an index of +1. Thus, G has at least one cell cycle. □

Thus, the Index Sum Invariant Theorem has at least one application. Still, this theorem is not quite what we set out to prove in the beginning of this thesis, as it is too limited to be a complete discrete analogue to the Poincaré-Hopf theorem. In the next chapter, we discuss some of the theorem's limitations, and ways we could improve them. We also discuss other possible applications of our discrete index theory that we want to investigate in the future.

Chapter 6

Future Work

In the previous chapters, we presented the framework for discretizing the Poincaré-Hopf Theorem. Namely, we constructed indices for the nodes and cells of a graph with a discrete vector field. Then we proved the Index Sum Invariant theorem, which states that the index sum of a finite, connected, planar graph with a polygonal boundary is always +2, as long as we add extra edges and a sink node. As with the Poincaré-Hopf theorem, this result asserts that the sum of the indices of an object is constant, even if the vectors on the object changes. Thus, this result mimics the Poincaré-Hopf theorem. However, there are some important discrepancies that prevent it from being a true discrete analogue.

The first is that, in the Poincaré-Hopf theorem, a vector field's index sum depended on the manifold that had the vector field. For example, while the index sum for a vector field on a sphere will always be +2, the index sum for a vector field on a torus will always be 0. In contrast, *any* graph that our theorem applies to will have the same index sum. So, unlike the Poincaré-Hopf theorem, which is about an invariant property of a given manifold, our theorem is an invariant property about all graphs of a certain type.

This could make intuitive sense if we again look to the continuous case. Imagine that we impose a graph with a discrete vector field onto the surface of a manifold. We then add new external edges and a sink node. We've given no restrictions on where the sink node can lie, nor where the edges can and cannot pass through on the manifold. Hence, we're assuming that we can place the new sink node anywhere on the manifold and still connect it to the graph via the new edges. In other words, we assume that the manifold has no holes. Therefore, the manifold should have the same Euler characteristic

as the sphere (the canonical example of a manifold with no holes) which is +2.

Thus, adding a sink node and external edges may reveal an invariant property of graphs with discrete manifolds, while also alluding to an invariant property of manifolds. This is a theory that we'd like to pursue in the future. Perhaps we can add extra edges, cells, and nodes to graphs in a way that assumes that the graph's underlying manifold is something else, like a torus. Doing so would require some way of encoding that the manifold has a hole. If we successfully did that, we'd expect the graph's index sum to then be the same as the new manifold.

The second limitation of our theorem is that it only applies to very specific types of graphs. Namely, those with polygonal boundaries, and extra edges and a sink node added. Because of this, our theorem is not as widely applicable as the Poincaré-Hopf theorem. One of our future goals is to remedy this issue by creating a theorem that applies to a wider range of graphs. First, we want to investigate how we can extend our theorem to apply to graphs that don't have a polygonal boundary. For example, if T is a tree graph then T^+ does not necessarily have an index sum of +2. This seems to be partly caused by the new cells we create when adding extra edges and a sink node to T. Unlike the cells we create when adding edges and a sink node to a graph with a polygonal boundary, these cells need not have an index of 0, which can unexpectedly alter the index sum. Thus, one goal moving forward is to determine how to add extra edges and sink nodes to tree graphs so that the resultant graphs' index sum is also +2.

Lastly, we plan to prove that the index presented in Definition 2.3.3 and the corresponding cell index are the same as the index presented on page 9 and its cell index. One of the challenges of doing so is that the former definition is defined only on a graph whereas the latter was based off of Definition 2.1.7, which was defined on a manifold. However, we believe that our idea of a completion could provide a way of connecting the two, and showing that they're the same.

We also want to address the issue involving Glass' work. As alluded to in the introduction, it recently came to our attention that Glass developed a discrete analogue to the Poincaré-Hopf theorem appears to be very similar to our Index Sum Invariant Theorem according to Gordon (1997). Because of this, we suspect that Glass' work addressed our inquiries about manifolds other than spheres. However, we do not believe Glass mentioned anything related to the concept of a cell completion, so this aspect of our work is likely completely novel. However, due to the ongoing COVID-19 pandemic, we

have been unable to obtain a copy of Glass' work to verify these speculations.

Despite Glass' work, we showed that the Index Sum Invariant Theorem still has an application. Specifically, we used it to give an alternative proof to a new graph theory problem known as the Game of Cycles. One of our goals moving forward is to see if our discrete index theory also has continuous applications. This would not be the first time that a discrete framework was used to do so. For example, Knaster et al. (1929) proved that Sperner's Lemma, a result about triangulating a triangle and labeling the resulting vertices, is equivalent to the Brouwer Fixed-Point Theorem, which states that any continuous map has a fixed point. Perhaps our discrete framework can also be used to prove a similar equivalence between the Poincaré-Hopf theorem and the Index Sum Invariant Theorem.

Bibliography

2019. Simplicial complex example - Simplicial complex - Wikipedia. URL https://en.wikipedia.org/wiki/Simplicial_complex#/media/File:Simplicial_complex_example.svg. [Online; accessed 25. Nov. 2019].

2019. Vector Field Generator. URL https://www.desmos.com/calculator/eijhparfmd. [Online; accessed 12. Dec. 2019].

Alvarado, Ryan, Maia Averett, Benjamin Gaines, Christopher Jackson, Mary Leah Karker, Malgorzata Aneta Marciniak, Francis Edward Su, and Shanise Walker. 2020. The Game of Cycles: An Introduction. [Author contact: su@math.hmc.edu].

Balanarayan, P., and Shridhar R. Gadre. 2003. Topography of molecular scalar fields. I. Algorithm and Poincaré–Hopf relation. *J Chem Phys* 119(10):5037–5043. doi:10.1063/1.1597652.

Forman, Robin. 2002. A User's Guide to Discrete Morse Theory. *Séminaire Lotharingien de Combinatoire* 48.

Gordon, C McA. 1997. Combinatorial methods in dehn surgery. In *Lectures at KNOTS'96 (Tokyo)*, 263–290.

Knaster, Bronisław, Casimir Kuratowski, and Stefan Mazurkiewicz. 1929. Ein beweis des fixpunktsatzes für n-dimensionale simplexe. *Fundamenta Mathematicae* 14(1):132–137. URL http://eudml.org/doc/212127.

Knill, Oliver. 2002. A Graph Theoretical Poincaré-Hopf Theorem .

———. 2018. The amazing world of simplicial complexes. *arXiv* URL https://arxiv.org/abs/1804.08211. 1804.08211.

———. 2019. A parametrized Poincare-Hopf Theorem and Clique Cardinalities of graphs. *arXiv* URL https://arxiv.org/abs/1906.06611. 1906.06611.

Knudson, Kevin P. 2015. *Morse theory: smooth and discrete.* World Scientific Publishing Company.

Milnor, John W., and David W. Weaver. 1965. *Topology from the Differentiable Viewpoint.* The University Press of Virginia Charlottesville.

Miura, Keiji, and Kazuki Nakada. 2017. Sparse Parallel Algorithms for Recognizing Touch Topology on Curved Interactive Screens. *IEEE Access* 5:14,889–14,897. doi:10.1109/ACCESS.2017.2733722.

Scoville, Nicholas A. 2019. *Discrete Morse Theory*, vol. 90. American Mathematical Society.

Su, Francis. 2020. *Mathematics for Human Flourishing | Yale University Press*. URL https://yalebooks.yale.edu/book/9780300237139/mathematics-human-flourishing. [Online; accessed 25. Nov. 2019].

Zhang, Shaoyu, Shuhao Sun, Hang Zhao, and Shenggang Li. 2018. An Application of the Poincare-Hopf Index Theorem: A Mathematical Model of Earthquake. *IOP Conf Ser: Mater Sci Eng* 466(1):012,055. doi:10.1088/1757-899x/466/1/012055.