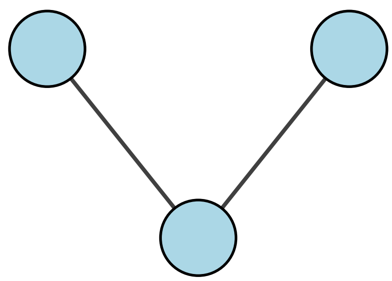
Homology of Finite Digraphs for Beginners

Kellen Hurley

November 2025

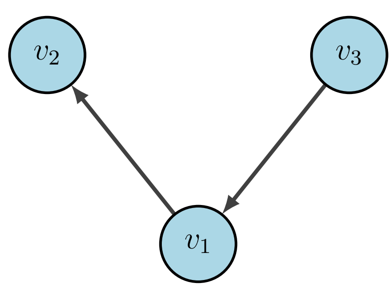
1 Graphs (Informal)

Informally, a graph is a collection of dots with a collection of links between them, like the picture below:



Why do we care about these objects? Graphs are extremely useful models for many real world phenomena such as road networks, cellular networks, social networks, etc. Consider the following example: for each Facebook user, create a dot in the graph. For each user's dot, create a link to another user's dot if the two users are friends. This is a classic example of a graph.

Let's start introducing some terminology. The "dots" in a graph are called **vertices**, and the "links" are called **edges**. There are also different types of graphs. The graphs we would like to concern ourselves with in this discussion are **finite directed graphs** or **finite digraphs** for short. The graph pictured above is what's known as an **undirected graph**. This is because the edges have no specified direction. One can traverse the edges freely without concern for orientation. The Facebook example we discussed is also an undirected graph since being friends with someone on Facebook requires that they be friends with you. It is a directionless relationship. So what is a **directed graph**? A directed graph is simply a graph in which the edges are given a direction. Let's take a look at an example. I will label the vertices so we can discuss direction much more clearly:



In this graph, we can travel from v_3 to v_1, but we cannot travel from v_1 to v_3 since this edge is directed from v_3 to v_1. When might we use a directed graph? Consider the following example: for each website on the internet, create a vertex in the graph. For each website's vertex, create a directed edge to another website's vertex if the website has a link to the other website. This is a great example of a directed graph since some websites have links to others without those websites have links in return. It should be noted that some websites might have links to each other, in which case there are directed edges in both directions. These directed edges essentially function like an edge from an undirected graph.

Let's now discuss the final adjective we would like to apply to our graph: **finite**. A **finite directed graph** is simply a directed graph in which there are finite vertices and finite edges. The directed graph pictured above is an example of a finite directed graph. Many people don't even consider the possibility of graphs with infinite vertices and infinite edges upon first learning about them so the finite descriptor probably feels natural anyway.

2 Graphs (Formal)

We now define the notion of a finite directed graph formally. In mathematics, we must precisely define concepts in terms of concrete mathematical objects. A picture of the object is not actually the object, it is a just a helpful tool for visualizing it. We first need to understand some preliminary definitions. These definitions will not be completely rigorous, but they will be rigorous *enough* for us to understand the definition of a finite directed graph.

Definition 1: An **n-tuple** is an ordered list of n items.

The formal term for an "item" in an n-tuple is **element**. Let's look at an example. Here is a 3-tuple:

$$(a, b, c)$$

It consists of the elements a, b, and c. Notice the notation here. We represent an n-tuple by placing the elements in parentheses separated by commas. Here is a 6-tuple:

$$(1.12, \pi, \text{red}, f, 678.09, \forall)$$

Notice that there need not be any correlation between the elements. You can put whatever you want inside an n-tuple. Let's talk about the *ordered* part of the definition. "Ordered" in this context means that swapping the ordering of the elements in an n-tuple results in a different object. To give a concrete example, this is to say:

$$(1, 2, 3) \neq (2, 1, 3)$$

While these 3-tuples contain the same elements, the elements appear in different orders so they are considered not equal.

Definition 2: A **set** is an unordered collection of items without duplicates.

We also call the "items" in sets **elements**. Let's look at an example of a set:

$$\{1, 2, 3\}$$

this set consists of the elements 1, 2, and 3. Notice the notation here. We represent a set by placing the elements in curly brackets separated by commas. Sets are unordered, which is to say that as long as two sets contain the same elements, they are equal, regardless of the order in which the elements appear. For example:

$$\{1, 2, 3\} = \{2, 1, 3\}$$

Sets also cannot contain duplicate elements. So

$$\{1, 1, 2, 3\}$$

is not a set. It should be noted however that n-tuples *can* contain duplicates. We use the symbol $\in$ to indicate that something is an element of a set. For example:

$$1 \in \{1, 2, 3\}$$

can be read as, "1 is an element of the set $\{1, 2, 3\}$".

Definition 3: A **function** or **map** f from a set X to a set Y is a mapping of elements in X to elements in Y such that

- For all $x \in X$, there exists $y \in Y$ such that $f(x) = y$
- For all $x \in X$, and all $y, y' \in Y$, if $f(x) = y$, and $f(x) = y'$, $y = y'$.

We call X the **domain** of f, and Y the **codomain** of f. (Note!!! Normally, we'd use the concept of **binary relations** to define maps. I've decided against this because I don't want to get stuck in the weeds, especially since this document is intended for beginners).

We use the notation $f : X \to Y$ to mean "f is a map from X to Y". Let's break this down a little bit. The first bullet point is just saying that all elements in X must map to something in Y. We can't just leave elements unmapped. The second bullet point is saying that we can only map each element in X to one element in Y, we can't map to multiple elements. Here is an example of a map with domain $\{1, 2, 3\}$ and codomain $\{a, b\}$:

$$f(1) = a,\ f(2) = a,\ f(3) = b$$

Each element in $\{1, 2, 3\}$ is mapped to an element in $\{a, b\}$, and each element in $\{1, 2, 3\}$ is mapped to only one element in $\{a, b\}$. Here is an example of something that is not a map:

$$f(1) = a, b,\ f(2) = a,\ f(3) = b$$

This is not a map because 1 gets mapped to two elements of $\{a, b\}$ which is forbidden by definition.

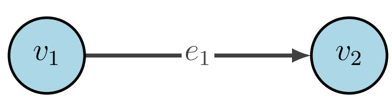
We now know just enough to understand the definition of a finite directed graph.

Definition 4: A **finite directed graph** $\Gamma = (V, E, s, t)$ is a 4-tuple in which:

- V and E are finite sets, referred to as the set of vertices and the set of edges respectively.
- $s : E \to V$ is a map, referred to as the source map
- $t : E \to V$ is a map, referred to as the target map

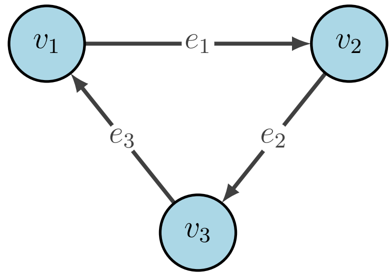
Let's examine this definition a little bit. V and E are the objects that represent our collections of vertices and edges. The source map s is the map that assigns each edge to the vertex it originates from, and the target map t is the map that assigns each edge to the vertex it points to. So s and t determine where edges go in the graph, and in what direction they face. Let's practice drawing a picture given the formal object to help us understand this better.

Let us draw a picture of the graph $\Gamma = (V, E, s, t)$ where $V = \{v_1, v_2\}$, $E = \{e_1\}$, $s(e_1) = v_1$, and $t(e_1) = v_2$:



Why is this the correct picture? Well, we have two vertices given by $V = \{v_1, v_2\}$, and one edge given by $E = \{e_1\}$. $s(e_1) = v_1$ which means the edge e_1 will originate, or start from, v_1. $t(e_1) = v_2$ which means the edge e_1 will point towards, or end at, v_2. This is precisely what we have drawn.

Let's now start with a picture, and define the formal object from the picture. Consider this picture of a graph:



Identifying the sets V and E is simple, simply look at the labelings of the vertices and the edges and we get: $V = \{v_1, v_2, v_3\}$, $E = \{e_1, e_2, e_3\}$. We must now define the source and target maps. Let's start with the source map. What is $s(e_1)$? e_1 starts from v_1, so $s(e_1) = v_1$. What about $s(e_2)$? e_2 starts from v_2, so $s(e_2) = v_2$. By similar reasoning, $s(e_3) = v_3$. Now let's define the target map. What is $t(e_1)$? e_1 ends at/points to v_2, so $t(e_1) = v_2$. By similar reasoning, $t(e_2) = v_3$ and $t(e_3) = v_1$. We have specified a mapping for each edge for s and t so these maps are now well-defined. In this case, there's actually a pattern in how edges are assigned, so we can define them more concisely like so:

$$s(e_i) = v_i, \quad (1 \leq i \leq 3)$$

$$t(e_i) = v_{(i \bmod 3)+1}, \quad (1 \leq i \leq 3)$$

You don't have to define s and t like this, this is just a more concise way to do it relative to the method of just explicitly writing down all of the mappings. The concise way of writing t uses a concept called "modular arithmetic", but it is not important that you understand this for this discussion.

This is all we need to know about graphs for our discussion about homology. Next, we will discuss the prerequisite knowledge we need in *linear algebra*.

3 Linear Algebra

In linear algebra, we are generally concerned with vector spaces, and linear transformations between them. For the purposes of this document however, we only need to worry about Euclidean space. Don't worry if these words don't mean anything to you, I will define everything we will need.

Definition 5a (Very Informal): A **real number** is any number that can be represented using a whole part, and a fractional part. $\mathbb{R}$ is the **set of real numbers**.

Most people do not know of any numbers that are not real (other than $i = \sqrt{-1}$ potentially), so you can probably just take "real number" to mean any number you can possibly think of. I will give the formal definition of the set of real numbers as well, but it requires knowledge of real analysis and set theory to understand.

Definition 5b (Formal) (Optional): Let $C = \{(x_n)_{n \in \mathbb{N}} : x_n \in \mathbb{Q},\ (x_n) \text{ is Cauchy}\}$. Define an equivalence relation $\sim$ on C such that $(x_n) \sim (y_n)$ iff $\lim_{n \to \infty}(x_n - y_n) = 0$. The **set of real numbers** is defined to be:

$$\mathbb{R} := C_{/\sim}$$

Let's take a look at some real numbers just so there's no confusion. The following are all examples of real numbers:

- 1
- 2.564
- $\sqrt{2} \approx 1.4142$
- $\pi \approx 3.14159$
- -3.3332322
- $5/4 = 1.25$

Definition 6: $\mathbb{R}^n$ (n a whole number greater than or equal to 1) is the set of n-tuples in which all elements are real numbers. (In set-theoretical terms, $\mathbb{R}^n$ is the Cartesian product of $\mathbb{R}$ with itself n times).

Let's look at some examples. Here are some elements of $\mathbb{R}^2$:

- $(1, 2)$
- $(\pi, 3.998)$
- $(0, -675)$

Here are some elements of $\mathbb{R}^3$:

- $(0, 0, 200)$
- (π, π, π)
- $(-19, 76, -78.76)$

For the rest of this discussion, we want to consider $\mathbb{R}^n$ as a vector space. The idea of a vector space is quite involved, requiring knowledge of abstract algebra, so I will define it very very informally just so that we get the big picture.

Definition 7 (Very Informal): A **vector space** consists of two sets, one which we call the set of **vectors**, and one which we call the set of **scalars**, with an addition operation between vectors (called vector addition), and a multiplication operation between vectors and scalars (called scalar multiplication) that work how we want them to (lol).

Definition 8 (Very Informal): n-dimensional Euclidean space is the vector space in which $\mathbb{R}^n$ is the set of vectors and $\mathbb{R}$ is the set of scalars, with addition between vectors given by:

$$(x_1, \ldots, x_n) + (y_1, \ldots, y_n) = (x_1 + y_1, \ldots, x_n + y_n)$$

and multiplication between vectors and scalars given by:

$$a \cdot (x_1, \ldots, x_n) = (ax_1, \ldots, ax_n)$$

n-dimensional Euclidean space is actually an even more specific structure called a **normed inner product space** (usually just called an inner product space since every inner product induces a norm), but we will ignore norms and inner products because they are not relevant to our discussion. Let's do some examples of vector addition and scalar multiplication in Euclidean space so that these words don't scare us anymore, it is really easy. Here are some examples of vector addition in $\mathbb{R}^2$:

- $(1, 2) + (3, 2) = (1 + 3, 2 + 2) = (4, 4)$
- $(-1, 5.5) + (0, -1) = (-1 + 0, 5.5 - 1) = (-1, 4.5)$

Here are some examples of scalar multiplication in $\mathbb{R}^2$:

- $2 \cdot (1,4) = (2 \cdot 1, 2 \cdot 4) = (2,8)$
- $-1 \cdot (\pi, 4/5) = (-1 \cdot \pi, -1 \cdot 4/5) = (-\pi, -4/5)$

Great, let's adopt some new notation. In linear algebra of Euclidean space, it is common to write vectors as columns of numbers like so:

$$\begin{bmatrix} 1 \\ 3 \\ 4 \end{bmatrix}$$

instead of the notation $(1,3,4)$. They both mean exactly the same thing, but the column notation is what's common in linear algebra. When we want to refer to an arbitrary vector in a vector space, we use the notation $\vec{v}$. We could use any symbol, but the arrow above it distinguishes this object as a vector. Next, let's talk about matrices.

Definition 9: An $n \times m$ **matrix with real entries** (n, m whole numbers greater than or equal to 1) is an $n \times m$ grid of real numbers where n is the number of rows and m is the number of columns.

Let's look at some examples. Here's a 2×2 matrix:

$$\begin{bmatrix} 1 & 2 \\ 3 & -1 \end{bmatrix}$$

Here's a 3×2 matrix:

$$\begin{bmatrix} 4 & 18 \\ -\pi & -87 \\ 1 & 3 \end{bmatrix}$$

Here's a 2×1 matrix:

$$\begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

and here's a 1×1 matrix:

$$\begin{bmatrix} 1 \end{bmatrix}$$

Notice that a 2×1 matrix is just a vector in $\mathbb{R}^2$, and a 1×1 matrix is just a scalar. Okay so what does a matrix do exactly? Well, an $n \times m$ matrix actually transforms vectors in $\mathbb{R}^m$ into vectors in $\mathbb{R}^n$ (a matrix is a map $T : \mathbb{R}^m \to \mathbb{R}^n$). Matrices are special kinds of maps known as **linear transformations**. We don't really need to know about linear transformations, but for those of you who are curious, I'll leave a definition below.

Definition 10 (Optional): Let U and V be vector spaces over a field F (the field of scalars). A map $T : U \to V$ is a **linear transformation** if

- $T(\vec{u_1} + \vec{u_2}) = T(\vec{u_1}) + T(\vec{u_2}),$ for all $\vec{u_1}, \vec{u_2} \in U$
- $T(c \cdot \vec{u}) = c \cdot T(\vec{u}),$ for all $\vec{u} \in U,\ c \in F$

Okay let's now learn how to "transform" vectors in Euclidean space using a matrix. We call this transformation process **matrix-vector** multiplication. Let's look at an example in which we transform a vector in $\mathbb{R}^2$ into another vector in $\mathbb{R}^2$. We require that our matrix have dimensions 2×2 to make such a transformation:

$$\begin{bmatrix} 1 & 3 \\ 2 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

How do we evaluate this expression? Firstly, extract the columns of the matrix from left to right like so:

$$\begin{bmatrix} 1 \\ 2 \end{bmatrix}, \begin{bmatrix} 3 \\ 1 \end{bmatrix}$$

Next, from top to bottom in our vector, extract each scalar and multiply it by the corresponding extracted column vector from left to right like so:

$$1 \cdot \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \; 2 \cdot \begin{bmatrix} 3 \\ 1 \end{bmatrix} = \begin{bmatrix} 6 \\ 2 \end{bmatrix}$$

Lastly, add together these extracted vectors:

$$\begin{bmatrix} 1 \\ 2 \end{bmatrix} + \begin{bmatrix} 6 \\ 2 \end{bmatrix} = \begin{bmatrix} 7 \\ 4 \end{bmatrix}$$

So this matrix transformed $\begin{bmatrix} 1 \\ 2 \end{bmatrix}$ into $\begin{bmatrix} 7 \\ 4 \end{bmatrix}$. As an equation, we write:

$$\begin{bmatrix} 1 & 3 \\ 2 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 7 \\ 4 \end{bmatrix}$$

Now let's do an example where we transform a vector in $\mathbb{R}^3$ into a vector in $\mathbb{R}^2$. In order to do this, we need a matrix with dimensions 2×3:

$$\begin{bmatrix} 1 & 1 & 2 \\ 0 & 4 & -1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 3 \\ -2 \end{bmatrix}$$

We follow the same process. First extract the columns of the matrix from left to right:

$$\begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 4 \end{bmatrix}, \begin{bmatrix} 2 \\ -1 \end{bmatrix}$$

Now from top to bottom in our vector, extract each scalar and multiply it by the corresponding extracted column vector from left to right:

$$1 \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \; 3 \cdot \begin{bmatrix} 1 \\ 4 \end{bmatrix} = \begin{bmatrix} 3 \\ 12 \end{bmatrix}, \; -2 \cdot \begin{bmatrix} 2 \\ -1 \end{bmatrix} = \begin{bmatrix} -4 \\ 2 \end{bmatrix}$$

Now we just add them together:

$$\begin{bmatrix} 1 \\ 0 \end{bmatrix} + \begin{bmatrix} 3 \\ 12 \end{bmatrix} + \begin{bmatrix} -4 \\ 2 \end{bmatrix} = \begin{bmatrix} 0 \\ 14 \end{bmatrix}$$

This is our solution. So

$$\begin{bmatrix} 1 & 1 & 2 \\ 0 & 4 & -1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 3 \\ -2 \end{bmatrix} = \begin{bmatrix} 0 \\ 14 \end{bmatrix}$$

Fantastic. We can now do matrix-vector multiplication. This leads us straight into the definition of the nullspace of a matrix.

Definition 11: The **nullspace** of an $n \times m$ matrix A denoted $\text{Null}(A)$ is the set of vectors $\vec{v} \in \mathbb{R}^n$ such that $A \cdot \vec{v} = \vec{0}$. (where $\vec{0}$ is the vector in $\mathbb{R}^m$ with all entries equal to 0).

Intuitively, the nullspace of a matrix is the set of vectors that get transformed into the zero vector. The nullspace is actually a concept that exists for all linear transformations, but we will only focus on nullspaces of matrices for this discussion. In order to compute homology groups for finite directed graphs, we will need to know how to find a **basis** for the nullspace of a matrix. We will not discuss what a basis actually is, but we will learn a method by which we can find one.

The method we will use is called **row reduction**, which is a process that turns a matrix into its **reduced row echelon form** or **RREF** for short. We have the following definition

Definition 12: A matrix A is in **reduced row echelon form** if it satisfies the following conditions:

- Any row consisting of all zeros is at the bottom of the matrix
- In each non-zero row, the leftmost non-zero entry is a 1. This entry is called a leading 1
- If a leading 1 appears in row i, column j, any leading 1 in a lower row $k > i$ must appear in a column strictly to the right $l > j$
- Any column with a leading 1 has zeros in all other entries

Let's look at an example of a matrix in RREF:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Let's verify all of the conditions. There are no rows with all zeroes so this matrix cannot possibly break the first condition. In each non-zero row (all of them), the leftmost non-zero entry is 1. There is a leading 1 in row 1 column 1, the leading 1 in row 2 appears in column 2 so the condition is not broken, and the leading 1 in row 3 appears in column 3 so the condition is again not broken. So condition three is satisfied. Condition four is easy to verify.

Okay now that we know what RREF looks like, let's learn how to turn a matrix into its RREF through row reduction. To be completed...