Topological Data Analysis
Mid-Term

Prepared By:

Dimitri KACHLER

Mirette MOAWAD

Nader KHALIL

Fall 2023

Table of Contents

1. How to run the program

1. Change the variable “`file_path`” to contain the path of the desired text file.
2. Run all cells.
3. Output is stored in *output.txt* in the same folder as the notebook.

2. Structure of the program

2.1. Algorithm

1. Reading input
2. Sorting of simplices by filtration value (while taking care that simplices with same filtration value get sorted according to dimension)
3. Construction of boundary matrix (sparse representation)
4. Reduction to row echelon form
5. Extraction of intervals
6. Writing Output in file
7. Showing barcode

2.2. Important variables in code

1. boundary_dict:

This is the sparse representation of the boundary matrix. It is a dictionary where:

- key = simplex ID
- value = set of simplices composing the boundary

boundary_dict[i] = {j, k, l} is equivalent to
B[j, i] = 1, B[k, i] = 1, B[l, i] = 1

Where B is the dense representation of the boundary matrix
Vertices have no entries in the dictionary.

Example:

```
B =
[[0. 0. 0. 1. 1. 0. 0. 0. 0. 0.]
 [0. 0. 0. 1. 0. 0. 0. 0. 1. 0.]
 [0. 0. 0. 0. 1. 0. 0. 1. 1. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 1. 1. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 1.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 1.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 1.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]]
```

in dense matrix representation is equivalent to

```
boundary_dict (sparse rep):
{9: {8, 6, 7}, 3: {0, 1}, 4: {0, 2}, 6: {1, 5}, 7: {2, 5}, 8: {1, 2}}
```

in sparse representation

2. pivot_dict:

This dictionary stores the pivot of each column.

- key = maximum value in list of simplices composing the boundary
- value = simplex ID

pivot_dict[i] = j is equivalent to

low(j) = i

Rows that are all zeros have no entry in the pivot dictionary.

Example:

The sample input given in the TD will have a pivot_dict:

```
Pivot dictionary:
{8: 9, 1: 3, 2: 4, 5: 6}
```

3. complex_dict:

This dictionary stores the complex itself.

- key = tuple of vertices
- value = simplex ID

It serves to quickly find the ID of a simplex based on the vertices that compose it. This functionality is needed in the construction of the boundary matrix.

3. Complexity Analysis

The worst-case time Complexity is $O(n^3)$ where n is the number of simplices in the complex.

As a forethought, there are some pre-processing steps such as reading the input in $O(n)$ time, or sorting by filtration values in $O(n \, log \, n)$ time, their complexity is small enough to not impact the main time complexity.

a. The outermost loop is in $O(n)$ as it iterates on all n simplices,

b. Retrieving the first pivot of a column requires for checking, in the worst case, as many faces as there are dimensions, $O(n)$

c. The reduction of a column is more subtle. In the worst case, it meets a pivot for every non-zero column to the left. The maximum number of pivots that will be encountered varies per column, but it can be taken as an average, i.e. $\frac{n+1}{2}$, which is also $O(n)$

d. To access the previous column for reduction, we take advantage of the dictionary's hash-based $O(1)$ lookup time

e. To add two columns together, the worst case is where the dimension is the same size as the number of vertices. This is, at least, is then bounded by $O(n)$

f. To insert a value into the dictionary, the complexity is virtually $O(1)$ depending on the hashing algorithm

Total: $O(n^3)$

Code-based view:

for every simplex:	$\big(O(n)\big)$
pivot = max(vertices)	$(O(n))$
while (pivot in pivot_dict)	$(O(n))$
j_ = pivot_dict[pivot]	$\big(O(1)\big)$
add columns j_ and simplex_id	$(O(n))$
pivot = max(vertices)	$\big(O(n)\big)$

Worst case scenario: $T(n) = n \cdot (n^2 + 2n) = n^3 + 2n^2$ iterations

$$= O(n^3)$$

4. Questions’ Solutions:

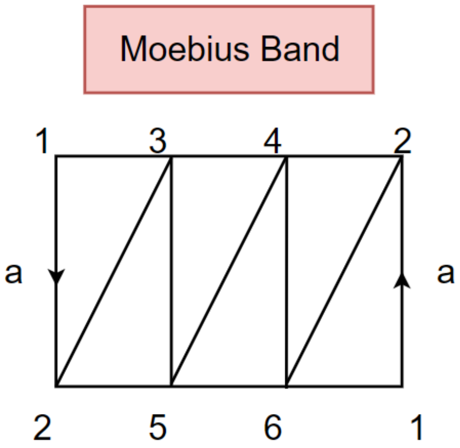
Q5 Solutions

Filtration files are included in the submission.

In all examples, we have chosen the filtration value to be equal to the dimension

So all vertices have value = 0, al edges have value = 1, all triangles have value = 2, and so on.

1. <u>Moebius Band:</u>

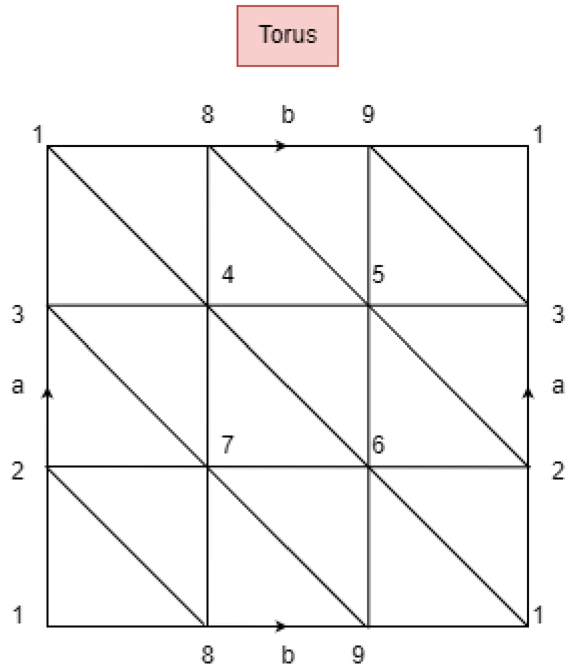


$$0 - simplices = \{0, 1, 2, 3, 4, 5, 6\}$$

$$1 - simplices = \{12, 13, 16, 23, 24, 25, 26, 34, 35, 45, 46, 56\}$$

$$2 - simplices = \{123, 235, 345, 456, 246, 126\}$$

2. <u>Torus:</u>



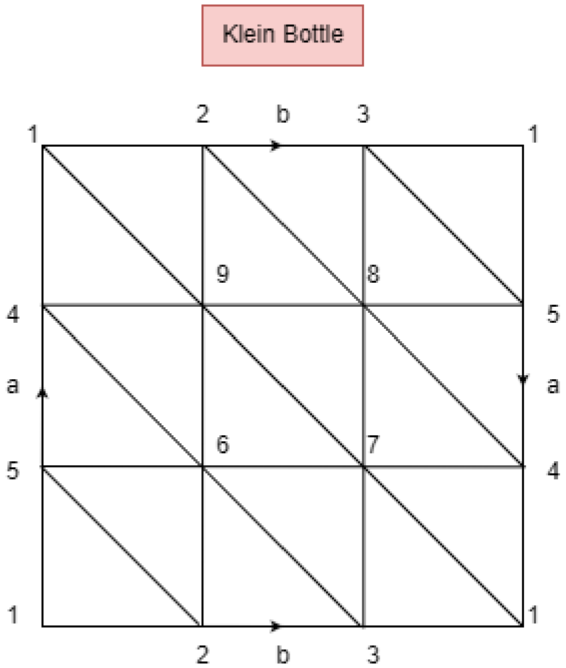
$$0 - simplices \ = \ \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$$

$$1 - simplices$$
$$= \{12, 13, 14, 16, 18, 19, 23, 25, 26, 27, 28, 34, 35, 37, 39, 45, 46, 47, 48, 56, 58, 59, 67, 69, 78, 79, 89\}$$

$$2 - simplices$$
$$= \{134, 148, 458, 589, 359, 139, 237, 347, 467, 456, 256, 235, 128, 278, 789, 679, 169, 126\}$$

3. <u>Klein Bottle:</u>

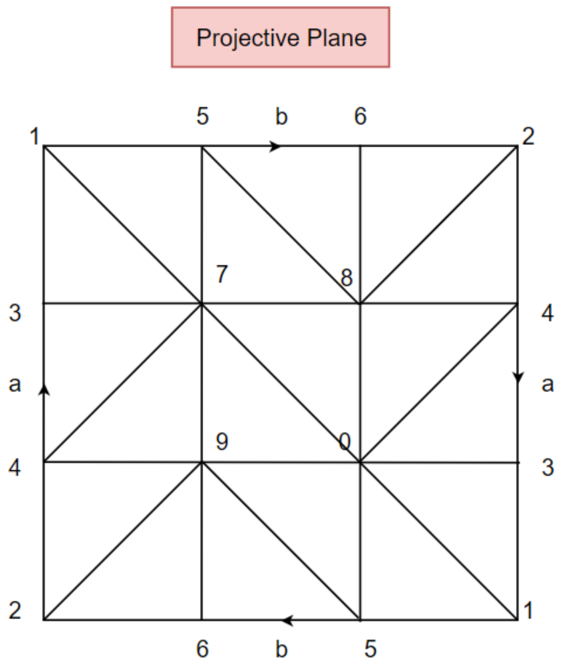
$$0 - simplices \ = \ \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$$



$$1 - simplices = \{12, 13, 14, 15, 17, 19, 23, 25, \\ 26, 28, 29, 35, 36, 37, 38, 45, 46, 47, 48, 49, 56, 58, 67, 69, 78, 79, 89\}$$

$$2 - simplices \\ = \{149, 129, 289, 238, 358, 135, 456, 469, 679, 789, 478, 458, 125, 256, 236, 367, 137, 147\}$$

4. Projective Plane:



$$0 - simplices \;=\; \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$$

$$1 - simplices = \{01, 03, 04, 05, 07, 08, 09, 13, 15, 17, 24, 26, 28, 29, 34, 37, 47, 48, 49, 56, 57, 58, 59, 68, 69, 78, 79\}$$

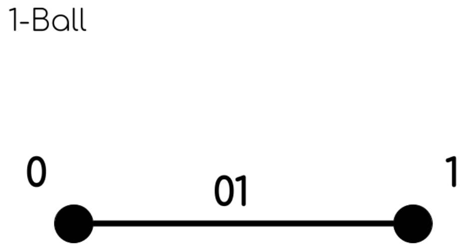
$$2 - simplices = \{137, 157, 578, 568, 268, 248, 347, 479, 097, 078, 048, 034, 249, 269, 569, 059, 015, 013\}$$

5. <u>D-spheres and D-Balls</u>

<u>Filtrations' Generation Method:</u>

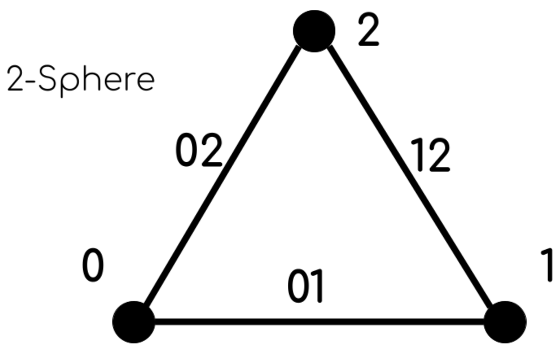
Each d-sphere is composed of $d+1$ vertices, edges that connect all vertices, triangles that connect all edges, and so on. A d-sphere can be obtained by computing all possible combinations of simplices.

A d-ball is a d-sphere with an attached d-simplex that is attached to all (d-1)-simplices in the d-sphere.



$$0 - simplices\ =\ \{0, 1\}$$

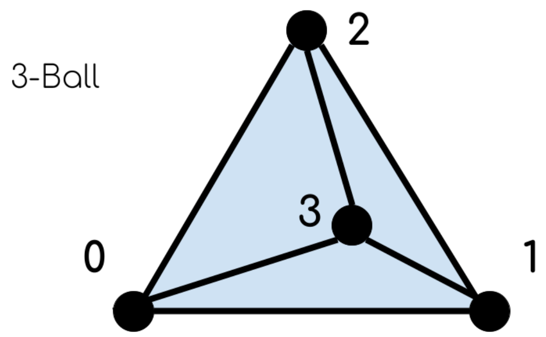
$$1 - simplices = \{01\ \}$$



$$0 - simplices\ =\ \{0, 1, 2\}$$

$$1 - simplices = \{01, 12, 02\ \}$$

Note: For the 3-Ball we have solid tetrahedron. It is not hollow, it's filled in.



$$0 - simplices \ = \{0, 1, 2, 3\}$$

$$1 - simplices = \{01, 12, 02, 03, 13, 23\}$$

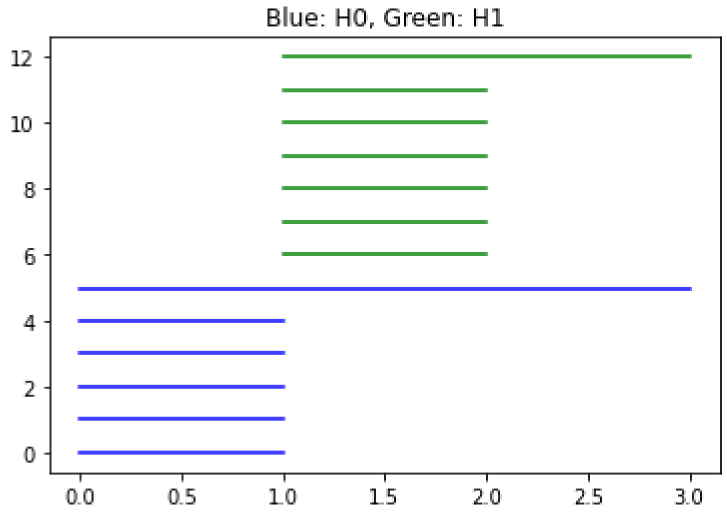
$$2 - simplices = \{023, 123, 012, 013\}$$

$$3 - simplices = \{0123\}$$

Q6: Barcodes

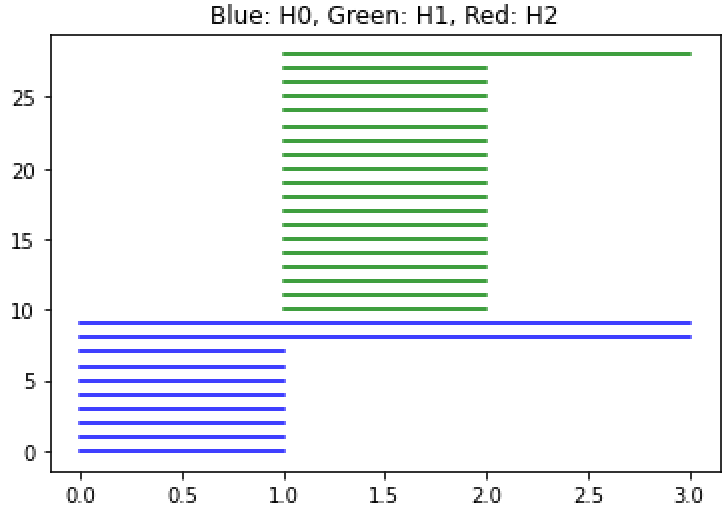
1. <u>Moebius Band:</u>

The bar code graph is consistent with what we know about the Moebius band. Namely, it is one connected component, and it also has one hole, as signified by the bar in H1.

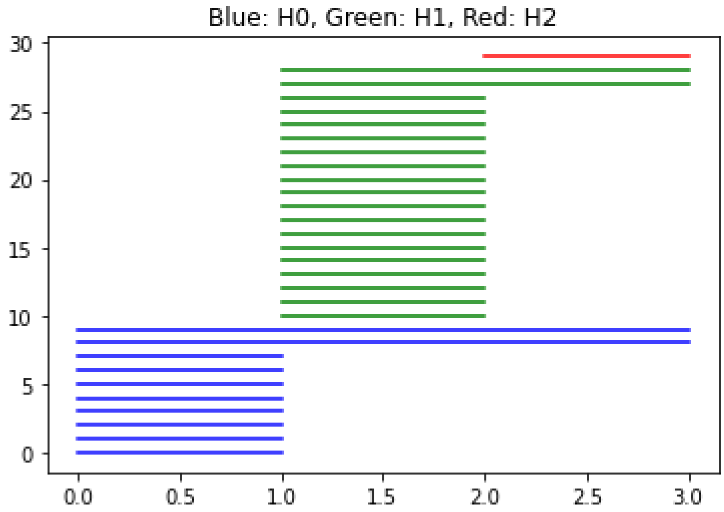


2. <u>Torus:</u>

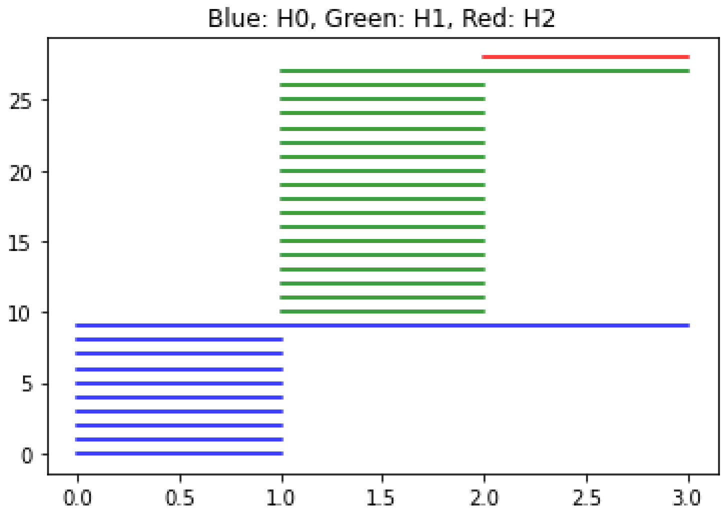
The graph shows that there are two bars in H0 that persist. However, this is inconsistent with what we know about the torus - it is not two separate connected parts. Therefore we can assume that the reduction is not valid in this case.



3. **Klein Bottle:**

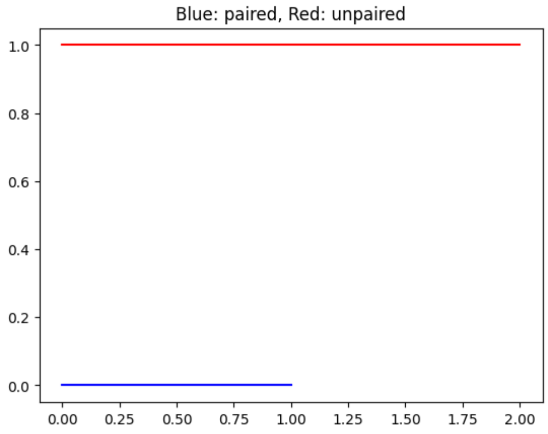


4. **Projective Plane:**

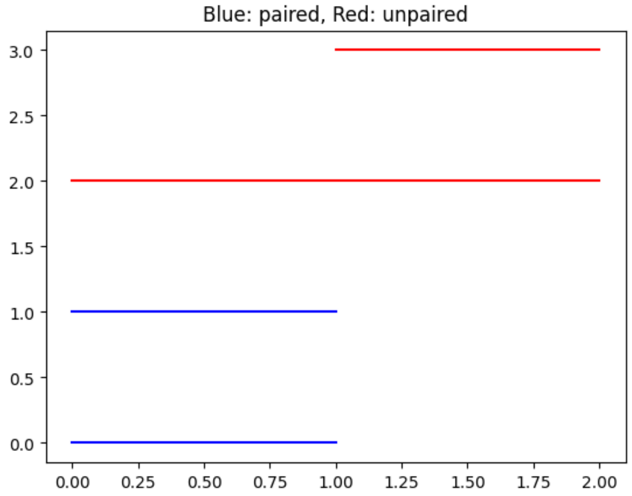


5. D-Sphere & D-Ball

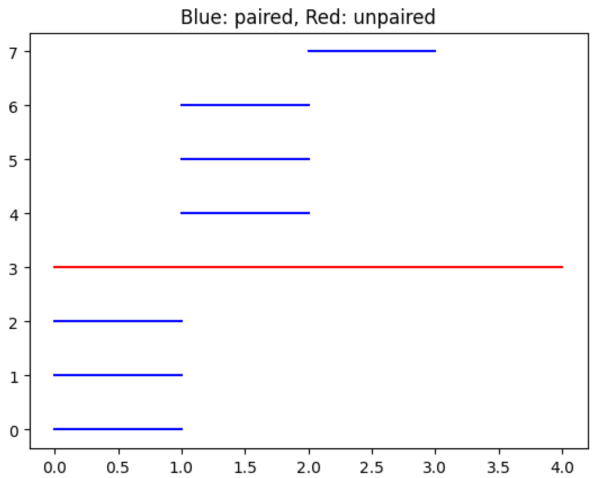
For **1-Ball**, it starts out with two disconnected vertices, and then they get connected with an edge, killing the first vertex, and letting the second persist.



For the **2-Sphere,** we start out with 3 vertices, however, two of the vertices are killed by the addition of the edges, leaving only one connected component. Furthermore, the top line is in H2, which represents the hole in the 2-Sphere.



Then for the **3-Ball,** it has no holes, so we don't expect any bar to persist, except for the one bar in H0 representing the single connected component.

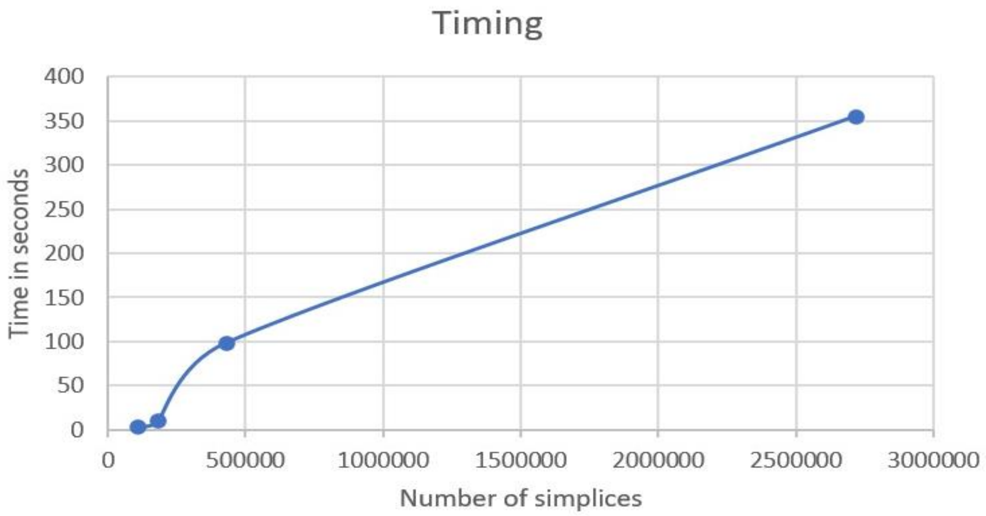


Q7 Solutions

All timings are in seconds.

Program instances were run on [Deepnote]

	Filtration A	Filtration B	Filtration C	Filtration D
Number of simplices	428 643	108 161	180 347	2 716 431
Whole program (including reading input and writing output to file)	209.132	5.351	23.244	393.437
Construction of boundary matrix + row echelon form reduction	158.100	4.733	10.443	371.553
Row echelon form reduction	99.037	3.863	10.138	355.277



This graph shows time complexity which is better than the worst-case scenario $O(n^3)$

Q8 Solutions

• **Filtration A:** In the early stages of the filtration, a lot of H0 bars start to come into existence, this indicates the formation of a lot of disconnected vertices. However, the majority of the disconnected vertices, with the exception of one, die out as edges get made between them forming one large component as seen by the long bar in H0. Some of the edges form the boundary of holes, which can be seen with the H1 bars. After some time, there does appear to be a period of stability where we observe two long-lasting bars - one in H0 and the other in H2. A particular shape that comes to mind with this homology is the sphere, as it is one connected component (H0=1) and it has one 3-dimensional hole (H2=1). → **Suggested Topological Space:** Klein Bottle

• **Filtration B:** Quite early on, there appears to be one main connected component, with the other vertices being hidden after the > 0.05 filter. In the beginning, there appear to be 5 different 1-cycles or holes, and as for the cavities in H2, there appear to be 8 of them before they die out. → **Suggested Topological Space:** Projective Plane

• **Filtration C:** Most of the bars do not seem to survive long, they seem to join the main connected component quite quickly. Early on, there are two different 1-cycles seen in H1, although only one of them seems to survive for much longer. During the period where we have one connected component, H0=1 and a hole, H1=1, the shape that fits this topological description is the circle. → **Suggested Topological Space:** Torus

• **Filtration D:** As per usual, there is one eldest connected component that outlives the rest as the vertices get connected. Furthermore, some holes start to form as the union of ball grows in diameter. Interestingly, if we restrict ourselves from -4 to -1, we notice an interesting pattern, H0=1, H1=2, H2=1. This description seems to line up with that of a torus. Of course, the holes die out eventually as we are left with one connected component**.**