

## Approximating torus-torus intersection for hollow double torus generation under the supervision of persistent homology

Md. Morshed Bin Shiraj<sup>1</sup> , Md. Safik Ullah<sup>1</sup> , Tozam Hossain<sup>1,2</sup>, Md. Mizanur Rahman<sup>1</sup> ,  
Md. Masum Murshed<sup>1</sup> , Nasima Akhter<sup>1</sup>

<sup>1</sup>Department of Mathematics, University of Rajshahi, Rajshahi, Bangladesh

<sup>2</sup>Department of Mathematics, Bangamata Sheikh Fojilatunnesa Mujib Science and Technology University, Jamalpur, Bangladesh

### ABSTRACT

**Background:** Visualization of complex mathematical surfaces, like the  $n$ -torus, is an open challenge. In this study, a double torus data generation process has been proposed using the deformation of a torus by a sphere.

**Methods:** It is proved that a torus  $T(R,r)$  can be deformed into a 3-sphere  $S^3(R+r)$  with the same center and it is the smallest ball to cover the torus  $T(R,r)$ . Then 3D and 4D torus data have been generated from their parametric equations. After that, the generated data have been compared with the known knowledge of the shape of a torus using the persistent diagram (PD). Then following the theoretical findings, an approximation to the torus-torus intersection has been computed to extract it from the union of two 4D torus data. Finally, the generated double torus data has been validated by explaining the hollowness in the intersection of two sampled torus introducing subtraction operation among the PDs of each of the generated structures on the proposed generation process.

**Results:** In the PDs, it is found that the 4D torus data gives more significant results than the 3D torus data which supports the claim of changing the original topology of a higher dimensional manifold by using lower dimensional reduction. The result shows that the double torus data with hollow intersection has been generated properly.

**Conclusion:** The successful generation of the double torus paves the way for creating more complex data with well-defined topology. This approach is particularly significant in scientific computation, especially for researchers focused on topological aspects.

### ARTICLE HISTORY

Received July 25, 2024

Accepted August 26, 2024

Published September 04, 2024

### KEYWORDS

Computer-aided geometric design; data visualization; double torus data generation; geometry processing; persistent homology; torus-torus intersection  
2020 MSC: 65D17, 65D18, 68W25.

### Introduction

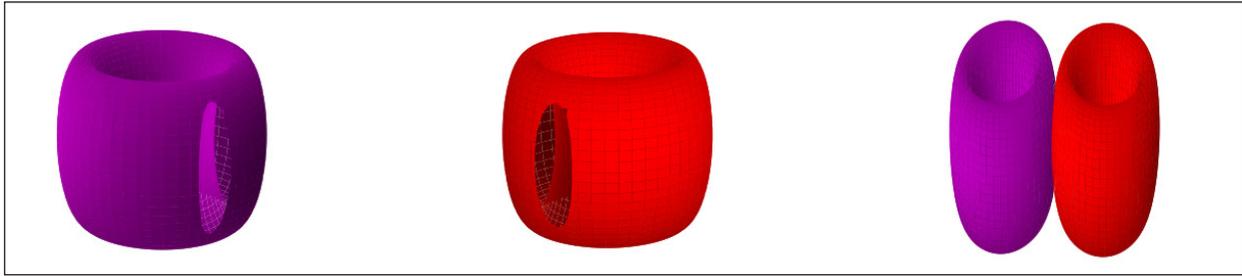
The study of shape is one of the fundamental fields of knowledge. Human has the natural intention of constructing shapes from vision. Torus is one of the most well-recognized structures in human history. These recognized structures can be regenerated from observed data and can be used in different simulations. For example, torus-shaped geometry has been detected from computed tomography data of some patients' prostates and has been regenerated to model prostate cancer growth [1].

Mathematicians have been fascinated by different structures for centuries. The double torus is the genus-2 torus which is the connected sum of two

tori. Some pure mathematical studies such as affine structures [2], homeomorphisms [3], and geometric uniformization [4] have been investigated on the double torus. Coloring a double torus surface [5] and some embedding on double torus have been studied [6]. Therefore, double torus structures have been used as a source of many rigorous mathematical studies.

On the other hand, double torus geometry has been mentioned for describing the 4D brain model [7], atomic structure, earth's electromagnetic field, and dynamic model of the universe [8]. Double torus shapes in galaxies have been found in [9]. A double torus 2D cosmological model has been

**Contact** Md. Morshed Bin Shiraj ✉ mbshiraj@gmail.com 📧 Department of Mathematics, University of Rajshahi, Rajshahi, Bangladesh.



**Figure 1.** Gluing two tori after cutting a disc from each of them to generate 2-torus.

proposed in [10]. A Double torus has been found as a core structure to explore some astrophysical phenomena [11]. In the field of electronics, double torus-shaped chaotic attractors have been designed in [12], [13], and [14]. Also, a double torus network has been proposed in [15], while a  $5D$  torus has been considered in [16]. Therefore, double torus-shaped structures are getting familiar with different fields of study. Therefore, it is important to generate double torus data to make a model of its relevant study object and to use it in further simulations before application.

The canonical and non-canonical homology basis of the double torus has been considered in [17] to define a continuous mapping between two structures of the same topology. But the generation of specific double torus data was not been mentioned in [17] which is important in geometric modeling. In [18], an Auto-Chart Encoder has been proposed and double torus data has been generated and studied as an example to make some fruitful conclusions based on the approximated manifold of the input data. But the generated double torus data is a lower dimensional approximation of its original geometry which can't preserve its topological properties including shape which leads to topological errors of the model. Also, the shape of the generated data had not been validated by evaluating its shape. In [19], the authors pointed out the necessity of generating proper topological data to count network dependencies and took sample points from the polygon quotient space of the double torus. Though double torus can't be parameterized globally, it can be generated by gluing two tori or taking a connected sum of two tori. Persistent cohomology has been introduced to count significant cocycles in [20] and has been tested to recognize circular structures in some data including double torus data. A double torus data of 3120 points merging two torus data of 1600 points each after cutting a torus by a plane and taking its reflection as a second torus has been generated by the authors. Though persistent diagrams

have been checked, they didn't provide much detail on generating double torus data and this approach can't generate double torus data by taking the connected sum of two different torus especially concerning the torus-torus intersection of two sampled torus.

The Intersection problem is one of the fundamentals of geometric modeling [21], [22]. Surface-surface intersection problems have been studied to form a nonlinear system of equations and computer-aided modeling (CAM/CAD) has been used to solve them in [23], [24], [25], [26], and [27]. But specifically, the torus-torus intersection problem has not been addressed especially considering their topology.

On the other hand, one of the major reasons for triangulation failure is the poor fitting of two intersecting surfaces allowing one surface's point into another [28]. So it is important to exclude all such points of torus-torus intersection in the generation of double torus data. To do so, the structure of the torus-torus intersection should be calculated to detect points of the torus-torus intersection. The structure of the surface-surface intersection problem has been addressed in [29]. Similar approaches have been studied in many more researches (such as [30], [31], [32], [33], [34], [35], and [36]). In [37], authors constructed two tori from parametric equations introducing characteristic points of their topology pre-images. Then topological features of characteristic points have been calculated using the method of perturbation and thus intersection of two tori has been computed. Though the methodology has better accuracy than the Tracing method, they did not generate double torus data using it. Also, double torus has been considered to calculate circles of the torus-torus intersection in [38], but double torus data has not been generated.

In this study, our main hypothesis in torus-torus intersection is that a torus can be covered by a  $3$ -sphere and hence we can construct a  $4D$  ball to cut the intersecting torus to separate its intersection

part from the union of two given torus by which double torus can be constructed. To execute the plan, two torus data have been constructed from the parametric equation of 4D torus. To ensure the

topological structure of the generated double torus data, we used persistent homology as it can be used as validation [39].

### Background

#### Statement of the problem

Double torus or 2-torus is a torus of genus 2 which is a cross product of two circles ( $S^1$ ) that can be written as  $S^1 \times S^1$ . A 2-torus is constructed as a connected sum of two tori that are glued by defining a homeomorphism between the boundaries of a disc that has been cut from one torus into the boundary of the same disc that has been cut from another torus [40] which is shown in Figure 1. In a topological sense, this construction of 2-torus is okay; but in reality, generating double torus from two given tori of specific geometry is not possible

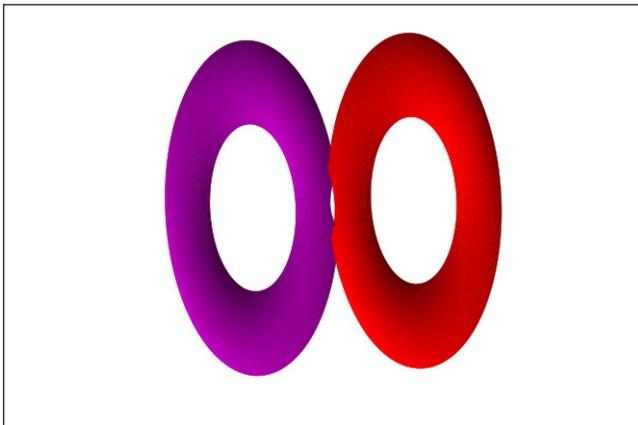


Figure 2. A Gap in the intersection area of the glued tori.

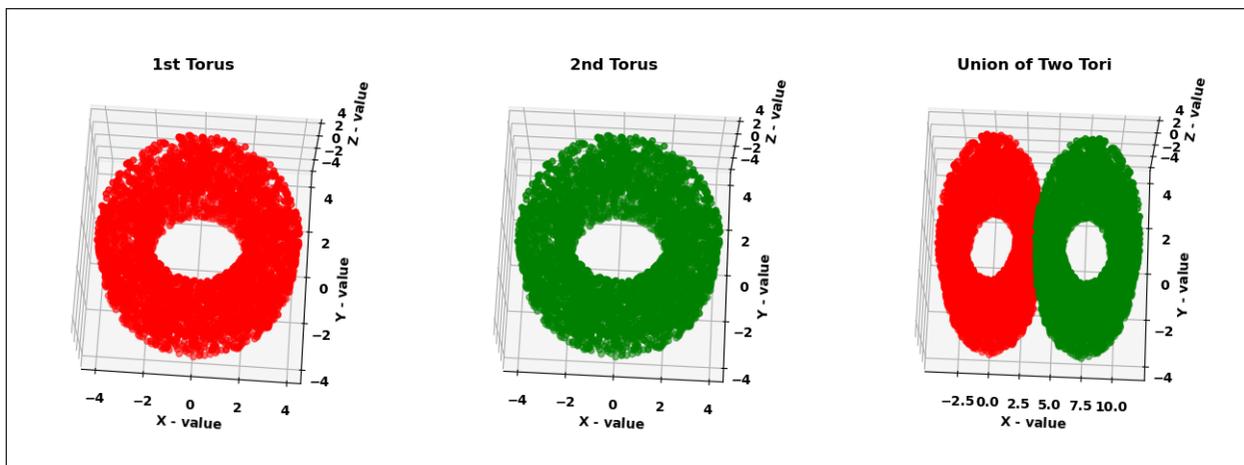


Figure 3. Two sampled torus of 5,000 points and their union.

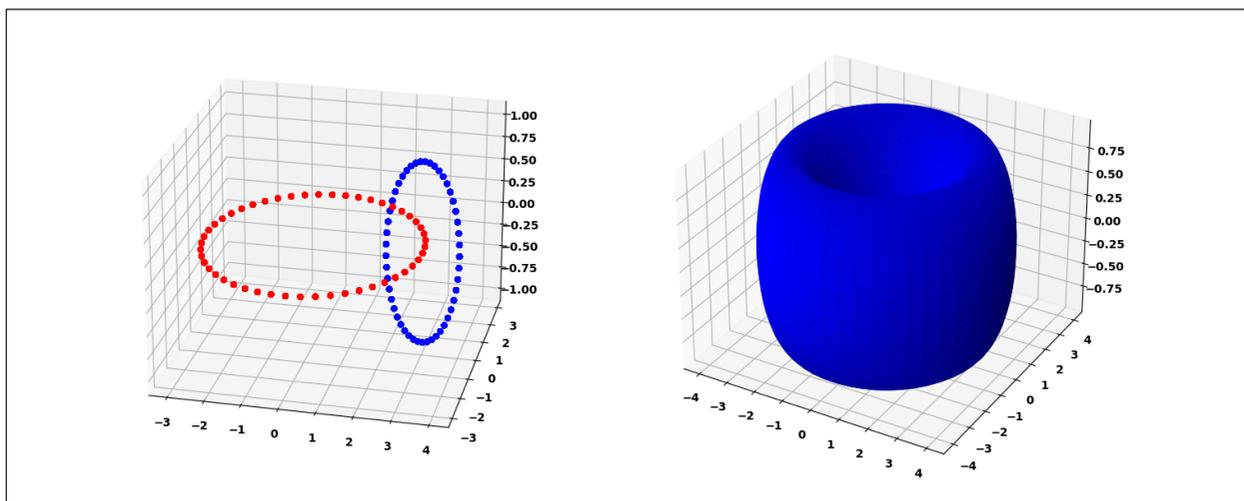
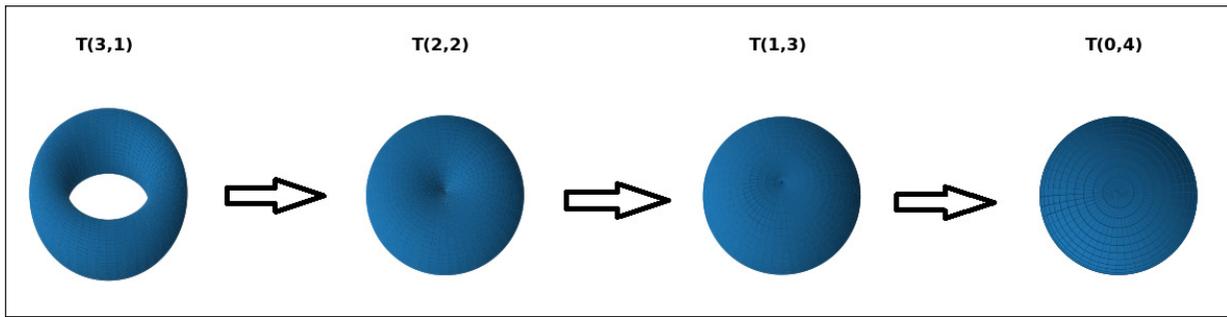


Figure 4. Generating a torus by rotating one circle around the other circle.



**Figure 5.** Deforming of a torus  $T(3,1)$  into a sphere  $T(0,4)$  or  $S^3(4)$ .

without specifying the radius of the ball by which discs have been cut from the given tori before gluing. Otherwise, two tori cannot be glued properly and so a certain gap will be produced in the intersection area as shown in Figure 2.

It is important to note that generating a double torus with a hollow intersection is an *open problem* in the field of computer graphics, and it is a challenging task that requires a deep understanding of mathematical and computational methods.

Let us think of a simpler way to construct a genus 2 torus. Suppose we have two sets of torus  $T_1$  and  $T_2$ . Shift the second torus  $T_2$  along the  $x$ -axis such that  $T_1$  and  $T_2$  have a common sphere-like intersection which is shown in Figure 3. Therefore, we have their union  $T_1 \cup T_2$ . Then the last step is to remove all the intersection points from  $T_1 \cup T_2$  to construct a double torus of  $T_1$  and  $T_2$  denoted by  $DT$ .

However, higher dimensional geometry is not developed enough to find the intersection of two higher dimensional objects. Hence, the problem is to determine the intersection of two higher dimensional objects which is known as the intersection problem. In our case, calculating the intersection between two tori, called the torus-torus intersection is one of the major focuses of this study.

**Proposed Solution**

However, some approximation techniques might be helpful to overcome this situation in some specific cases of intersection problems, especially for torus-torus intersection problem. Before going to the methodology, let us check the foundation of our hypothesis as started with the following theorem.

**Theorem 1.** *A torus  $T(R, r)$  can be degenerated into a 3-sphere  $S^3(R + r)$  centered at the center of the torus  $T(R, r)$ .*

*Proof.* Let  $R$  be the radius of the first circle and  $r$  be the radius of the second circle of a torus  $T(R, r)$ , where  $r < R$ . Rotating the second circle along the first circle, the torus can be generated which is shown in Figure 4.

Let us consider an arbitrary constant  $0 \leq \Delta r \leq R$  as a deforming constant to change  $R$  into  $R - \Delta r$  and  $r$  into  $r + \Delta r$  so that the sum of the radii  $R - \Delta r + r + \Delta r = R + r$  remains unchanged.

If we increase  $\Delta r$  from 0 to  $R$ , radius  $R - \Delta r$  will be changed from  $R$  to 0, and radius  $r + \Delta r$  will be changed from  $r$  to  $r + R$ . The torus will be deformed into 3-sphere sphere  $S^3$  of radius  $R + r$  without changing the sum of the radii. Therefore, the torus  $T(R, r)$  degenerated into the 3-sphere  $S^3(R + r)$  centered at the center of the torus  $T(R, r)$ . A particular example starting from  $R = 3$  and  $r = 1$  has been shown in Figure 5.

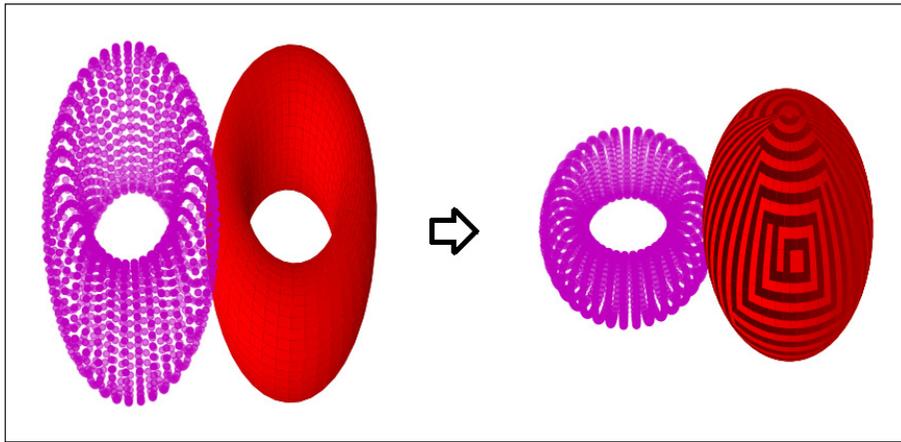
Let us denote the 3-sphere  $S^3(R + r)$  centered at the center of the torus  $T(R, r)$  by  $S^3(T(R + r))$ .

Obviously, all 3-spheres of radius greater than  $R + r$  centered at the center of the torus  $T(R, r)$  will cover the torus. Therefore, Theorem 1 gives us the following lemma.

**Lemma 1.** *A 3-sphere  $S^3(T(R + r))$  is the smallest ball to cover the torus  $T(R, r)$ .*

Thus a torus  $T(R, r)$  has been replaced by a 3-sphere  $S^3(T(R + r))$  to consider a sphere-like intersection of the torus with another torus. In this way, the torus-torus intersection has been approximated into a torus-sphere intersection to calculate the intersection of two tori which is shown in Figure 6.

Let us consider  $T_1 \cap T_2$  be the intersection of two tori  $T_1(R_1, r_1)$  and  $T_2(R_2, r_2)$ . To calculate  $T_1 \cap T_2$ , we need to use our approximated 3-sphere



**Figure 6.** Torus has been approximated by sphere to approximate the torus-torus intersection into a torus-sphere intersection.

$S^3(T_1(R_1, r_1))$  and  $S^3(T_2(R_2, r_2))$  to separate intersection points from  $T_2$  and  $T_1$ , respectively. Thus,

$T_1 \cap S^3(T_2(R_2, r_2)) = \{x \in T_1 : |x - x_0| \leq R_2 + r_2, x_0 \text{ is the center of } S^3(T_2(R_2, r_2))\}$ .

$S^3(T_1(R_1, r_1)) \cap T_2 = \{x \in T_2 : |x - x_0| \leq R_1 + r_1, x_0 \text{ is the center of } S^3(T_1(R_1, r_1))\}$ .

Thus, we have

$$T_1 \cap T_2 \approx \{T_1 \cap S^3(T_2(R_2, r_2))\} \cup \{S^3(T_1(R_1, r_1)) \cap T_2\}.$$

Therefore, we need to consider a 3-sphere/ball  $S^3$  of radius  $(R_1 + r_1)$  or  $(R_2 + r_2)$  centered at the center of its mother torus to separate intersection points from  $T_1 \cup T_2$ . Thus, the double torus generated from the given tori,  $DT$  can be calculated by the following equation:

$$DT = (T_1 \cup T_2) - (T_1 \cap T_2).$$

From another point of view, a double torus can be defined from given two sets of torus points  $T_1$  and  $T_2$  using a set operation called symmetric difference (denoted by  $\Delta$ ). Mathematically,

$$DT = T_1 \Delta T_2 = (T_1 \setminus T_2) \cup (T_2 \setminus T_1) \\ \approx \{T_1 \setminus S^3(T_2(R_2, r_2))\} \cup \{T_2 \setminus S^3(T_1(R_1, r_1))\}.$$

Let us denote  $T_1 \setminus S^3(T_2(R_2, r_2))$  by  $T_2$  cut and  $T_2 \setminus S^3(T_1(R_1, r_1))$  by  $T_1$  cut. Thus

Based on this finding, we have designed our methodology which is discussed in the Methodology section.

In both of the above cases generated double torus  $DT$  will be the same, since

$$(T_1 \cup T_2) - (T_1 \cap T_2) = \{T_1 - (T_1 \cap T_2)\} \cup \{T_2 - (T_1 \cap T_2)\} \\ = (T_1 \setminus T_2) \cup (T_2 \setminus T_1) = T_1 \Delta T_2.$$

### Other Tools to Validate this Study

In this study, double torus data has been generated theoretically in the sections named proposed solution and practically in “Results and Discussion.” Since  $DT$  is a 4-dimensional object, it is hard to visualize it. Though  $DT$  can be visualized as a 3-dimensional projection, topological invariants of the 3-dimensional projected torus were not as proper as a torus should have been (see “Results and Discussion” for details). Also, using lower dimensional reduction of a higher dimensional manifold might change the original topology of the manifold claimed in [39]. So, to study the topological property of higher dimensional objects, it is important to generate the object on its own domain. After generating higher dimensional data, one of the major challenges is to check the shape of it. This challenge pushed us to the validation of the generated double torus data.

In algebraic topology, a topological invariant named homology can count holes in a structure [41]. Then, 0-dimensional hole  $H_0$  (called as connected components), 1-dimensional hole  $H_1$

(known as loops), and 2-dimensional hole  $H_2$  (known as voids) of a torus have been calculated in [42]. Considering the double torus as a compact orientable surface, the homology of  $DT$  has been calculated. As these holes have been used frequently in algebraic topology, this information of holes of a

torus and  $DT(H_0, H_1, H_2)$  is assumed as a known knowledge of the shape of a torus and  $DT$ . Thus, the knowledge of holes of a torus and expected holes of a double torus are the reference knowledge that will validate our study. To do so, a strong tool to detect the shape of torus and double torus data is needed. A new emerging tool called topological data analysis (TDA) has been used to calculate  $H_0, H_1$ , and  $H_2$  of any finite higher dimensional data. So, we plan to detect holes  $H_0, H_1$ , and  $H_2$  of our generated double torus data, then compare it with our reference knowledge and finally discuss our findings in the "Methodology" section.

In TDA, filtration has been defined by introducing a constant  $\epsilon$  as the radius of a ball centered at each point of the given data points to construct a simplicial complex and chain complex from which the number of holes ( $H_0, H_1, H_2$ ) are calculated (see [40] for details). Plotting  $H_0, H_1, H_2$  against  $\epsilon$ , the persistence of holes has been visualized as a persistent diagram (PD). Also, a barcode is used to visualize the lifetime of each  $H_0, H_1$ , and  $H_2$  against

$\epsilon$ . In this study, a module has been written using the Python package ripser [43] to calculate PDs and barcodes to generate  $DT$  in "Results and Discussion." In the whole generation process of  $DT$  data, PDs of each step have been checked to validate our reference known knowledge ensuring the shape of it.

### Methodology

The following steps have been followed to generate double torus data:

Step 1: Two torus data  $T_1$  and  $T_2$  have been constructed by using parametric equations 2 of the 4D torus.

Step 2: Then shift  $T_2$  along the x-axis so that it intersects  $T_1$  spherically.

Step 3: Then join  $T_1$  and  $T_2$  using union operation which has a torus-torus intersection part inside.

Step 4: After that, using our hypothesis a 4D sphere (considered a ball) has been constructed.

Step 5: Then  $T_1$  cut and  $T_2$  cut have been calculated separating the torus-torus intersection data from  $T_2$  and  $T_1$ , respectively.

Step 6: Then, the union of the  $T_1$  cut and  $T_2$  cut constructs the double torus data.

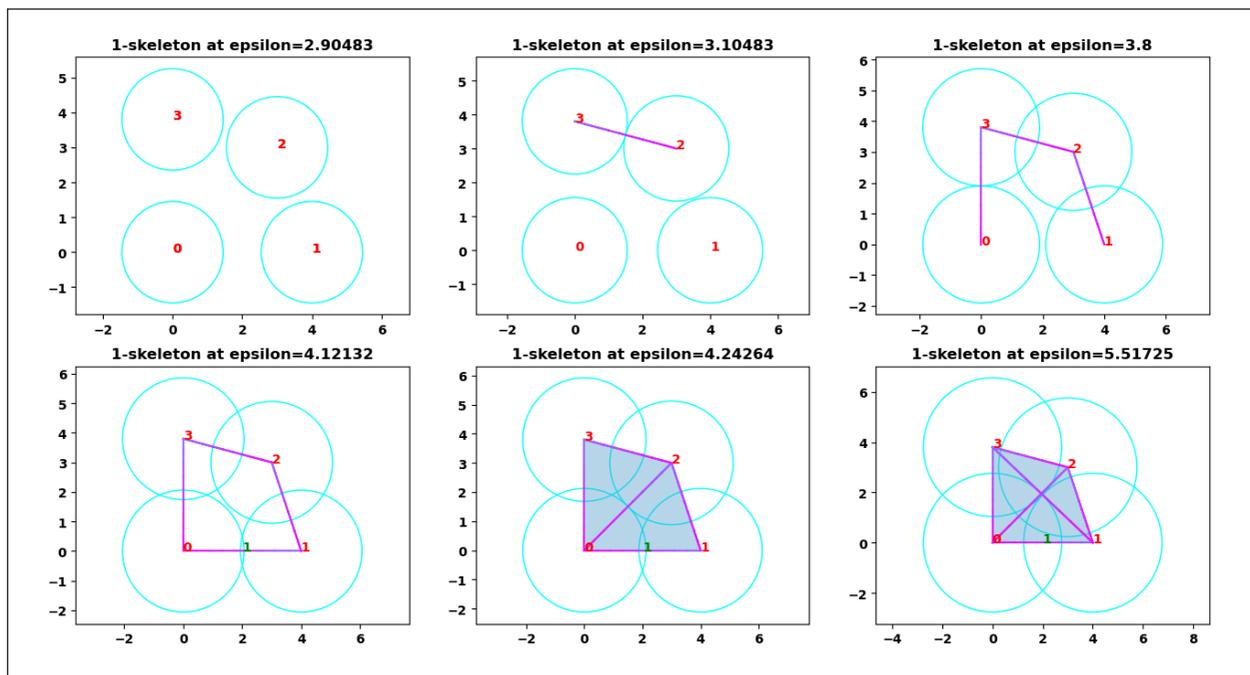


Figure 7. Building a simplicial complex of four points at different radii of mini balls  $\epsilon$ .

Step 7: To validate our generated data we have used persistent homology to track changes of data in each step of the generation process.

### Results and Discussion

Ripser is one of the most popular and efficient Python packages which is capable of computing persistent homology of 5 types of data files (distance matrix, lower distance matrix, upper distance matrix, point cloud, and DIPHA) using Rips complexes [43]. In this study, Ripser has been used to calculate persistent homology and its diagrams using point cloud data only.

To explain the construction of simplicial complexes, 1-skeleton simplicial complexes of a set containing 4 points have been shown with increasing radius of mini balls or thresh denoted as  $\epsilon$  centered at each of the points as shown in Figure 7. Two points have been connected by a line if their  $\epsilon$ -balls intersect each other. In this way, simplicial complexes have been created at different values of  $\epsilon$ . At a certain value of  $\epsilon$  ( $\epsilon = 4.12132$ ), a cycle has been generated on it denoted by 1. The value of  $\epsilon$  at which the cycle has been generated is called the birth of the cycle and after a certain period, it dies because of appearing another line connecting any two points of the cycle. That particular value of  $\epsilon$  is called the death of the cycle. This is how the birth and death of each cycle have been calculated for a

point data set which is stored as persistence data for each cycle.

Here cycles have been created from 1-skeleton simplicial complexes of sample data of the generated torus taking 20 points of them that are indicating 1D holes as shown in Figure 8. Thus, 2D holes have been calculated by creating 2-skeleton simplicial complexes. After calculating the persistence data of the generated 3D and 4D torus, it has been visualized in two ways by plotting death against birth as a PD and by plotting lifetime as a line against  $H_0, H_1$ , and  $H_2$  as a barcode are shown in Figures 9 and 10.

As mentioned in the "Other Tools to Validate this Study" section some topological invariants may lose to lower dimensional reduction of a torus [39]. To check the homology of 3D torus compared with the homology of 4D torus, a 3D torus data of 400 points has been generated using the following parametric equation:

$$\begin{aligned} x(\vartheta, \varphi) &= (R + r \cos \vartheta) \cos \varphi, \\ y(\vartheta, \varphi) &= (R + r \cos \vartheta) \sin \varphi, \\ z(\vartheta, \varphi) &= r \sin \vartheta \end{aligned} \quad (1)$$

with  $\vartheta, \varphi \in [0, 2\pi)$ .

Then, the PD and barcode have been calculated (Figure 9) of the generated torus points using

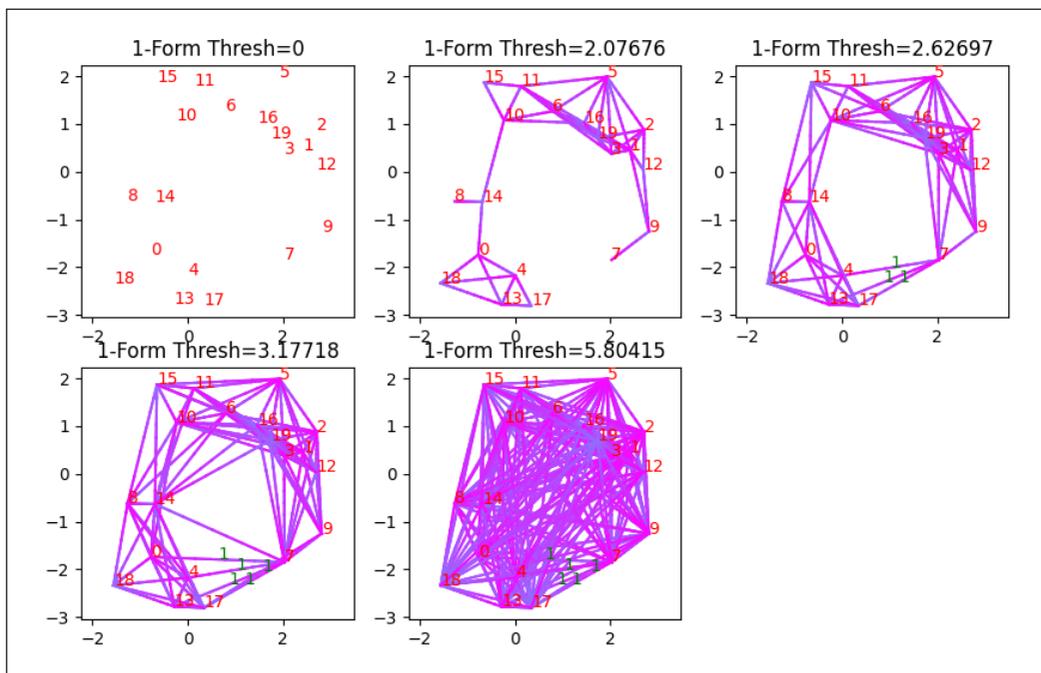


Figure 8. Simplicial complexes of 20 points of 4D torus.

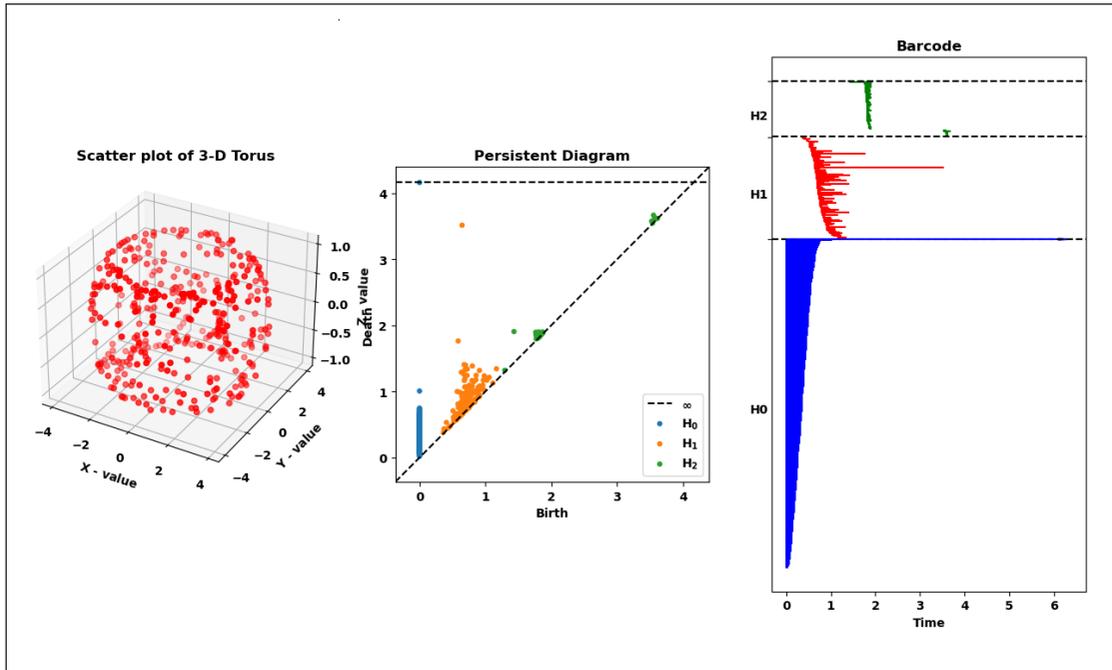


Figure 9. PD and barcode of the 3D torus point cloud.

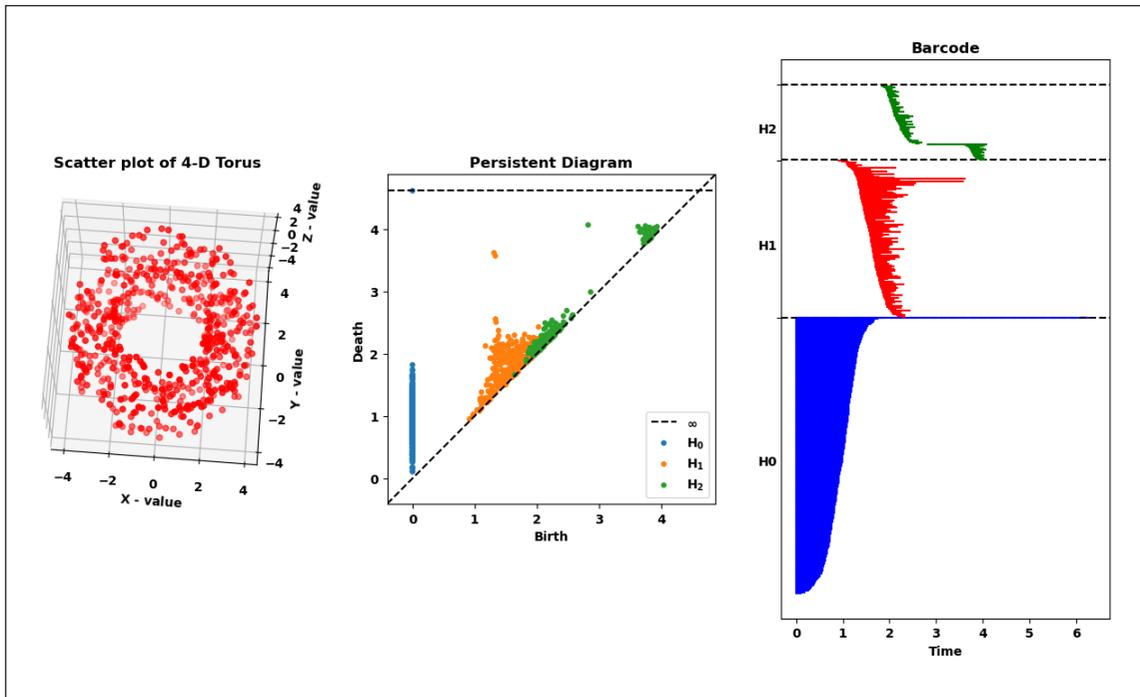
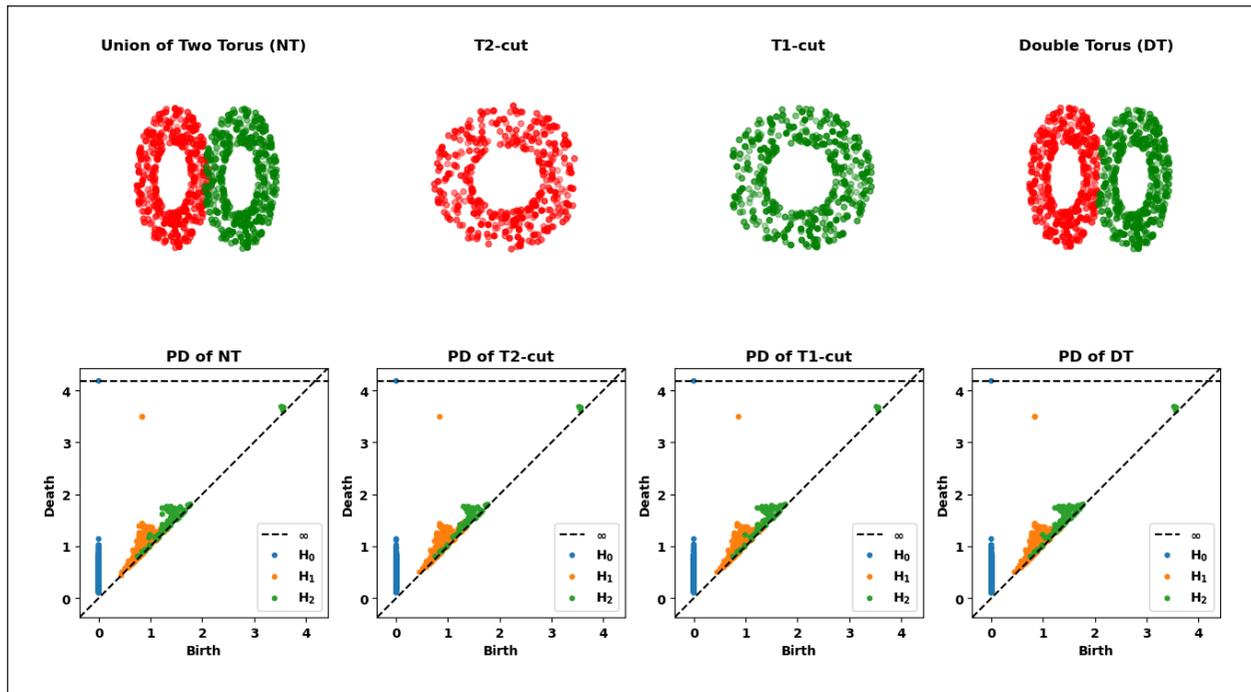


Figure 10. PD and barcode of the 4D torus point cloud.

Ripsier considering  $R=3$  and  $r=1$  to compare the result with Figure 10. From Figure 9,  $H_1$  has one point that persists significantly but one more was expected and  $H_2$  hardly persists. On the other hand, 400 data points of 4D torus have been generated considering  $R=3$  and  $r=1$  using the following parametric equation:

$$s(\varphi, \theta, \psi) = [(R+r \cdot \cos(\varphi)) \cos(\vartheta), (R+r \cdot \cos(\varphi)) \sin(\vartheta), (R+r \cdot \sin(\varphi)) \cos(\psi), (R+r \cdot \sin(\varphi)) \sin(\psi)], \quad (2)$$

with  $\vartheta, \varphi, \psi = [0, 2\pi)$ .



**Figure 11.** Steps of double torus data generation and PD of  $NT$ ,  $T_2$  cut,  $T_1$  cut, and double torus.

Then Figure 10 has been made following the same approach of the 3D torus. Since Figure 10 has a clear recognition of persisting one  $H_0$ , two  $H_1$ , and one  $H_2$  points significantly which is exactly equal to our known knowledge of the shape of a torus. Thus some influence of lower dimensional reduction of the torus has been issued. Therefore it is recommended to generate 4D torus data without taking any lower dimensional reduction, especially for preserving its topology. Naturally, 3D torus data can be used for the entire process of creating double torus data that is suggested in this work. In this experiment, we opt to use 4D torus data since we want to verify if the created double torus data is hollow inside.

To construct a double torus as found in the “Other Tools to Validate this Study” section theoretically, at first, two 4D torus named as  $T_1$  and  $T_2$  centered at  $(0,0,0,0)$  and  $(2R+2r-r/2,0,0,0)$ , respectively, containing 400 points each have been generated by using (2) considering  $R=3$  and  $r=1$  as mentioned in the “Methodology” section. Then,  $T_1$  and  $T_2$  have been joined with the set union, that is, newly joined torus  $NT$  can be written as  $NT=T_1 \cup T_2$  which contains 800 points. Then, Euclidean distances

between any two points of  $NT$  have been introduced. As defined in the “Other Tools to Validate this Study” section,  $T_1$  cut has been calculated as the set of all those points of  $NT$  whose distance from the center of  $T_1$  is greater than or equal to the radius of the approximated 3D sphere of  $T_1$  torus  $R+r=3+1=4$ . Thus,  $T_1$  cut consisting of 375 points has been computed which is equal to  $T_2 \setminus T_1$  considering 3D sphere  $S^3(T_1(R,r))$  as an approximation of  $T_1$  torus. Similarly, a  $T_2$  cut consisting of 383 points has been computed which is equal to  $T_1 \setminus T_2$  considering 3D sphere  $S^3(T_2(R,r))$  as an approximation of  $T_2$  torus. Finally, the union of  $T_2$  cut and  $T_1$  cut construct  $DT$  which contains 758 points. The generated  $NT$ ,  $T_1$  cut,  $T_2$  cut, and  $DT$  have been shown in Figure 11.

Table 1 shows homologies of  $DT$  that have two significant overlapped points of  $H_1$  and two very close points of  $H_2$ . Obviously, these overlapping cannot be detected from the PD of  $DT$  in Figure 11. So,  $H_0, H_1$ , and  $H_2$  of  $DT$  have been ordered decreasingly considering each point’s lifetime (i.e., death-birth) and the first 10 rows of the data set

**Table 1.** Significant homologies of *DT*.

$H_0$			$H_1$			$H_2$		
Birth	Death	Lifetime	Birth	Death	Lifetime	Birth	Death	Lifetime
0	inf	inf	1.50364	3.7196	2.21596	2.8048	4.20771	1.40291
0	1.94059	1.94059	1.55584	3.65461	2.09877	2.81897	4.20771	1.38874
0	1.86374	1.86374	1.55584	3.65461	2.09877	2.43653	3.19298	0.75645
0	1.84732	1.84732	1.37491	3.03275	1.65784	2.90905	3.61087	0.70182
0	1.84732	1.84732	1.50364	2.7955	1.29186	3.08696	3.60688	0.51992
0	1.83902	1.83902	1.51649	2.75927	1.24278	3.7196	4.21659	0.49699
0	1.83636	1.83636	1.58048	2.75927	1.17879	3.71847	4.11452	0.39605
0	1.8108	1.8108	1.585	2.75927	1.16713	3.71847	4.11452	0.39605
0	1.8108	1.8108	1.585	2.75927	1.16713	3.75589	4.10228	0.34639
0	1.78019	1.78019	1.46216	2.61354	1.15138	3.75589	4.10228	0.34639

**Table 2.** Significant homologies of *NT*.

$H_0$			$H_1$			$H_2$		
Birth	Death	Lifetime	Birth	Death	Lifetime	Birth	Death	Lifetime
0	inf	inf	1.50364	3.7196	2.21596	2.8048	4.20771	1.40291
0	1.94059	1.94059	1.55584	3.65461	2.09877	2.8048	4.20771	1.40291
0	1.86374	1.86374	1.55584	3.65461	2.09877	2.43653	2.84649	0.40996
0	1.84732	1.84732	1.40604	2.61354	1.2075	3.71847	4.11452	0.39605
0	1.84732	1.84732	1.585	2.75213	1.16713	3.71847	4.11452	0.39605
0	1.8108	1.8108	1.585	2.75213	1.16713	3.7196	4.10322	0.38362
0	1.8108	1.8108	1.47946	2.61354	1.13408	3.75589	4.10228	0.34639
0	1.78019	1.78019	1.50364	2.61354	1.10988	3.75589	4.10228	0.34639
0	1.78019	1.78019	1.37491	2.44595	1.07104	3.73619	4.05671	0.32052
0	1.7709	1.7709	1.58048	2.61352	1.03304	3.73619	4.05671	0.32052

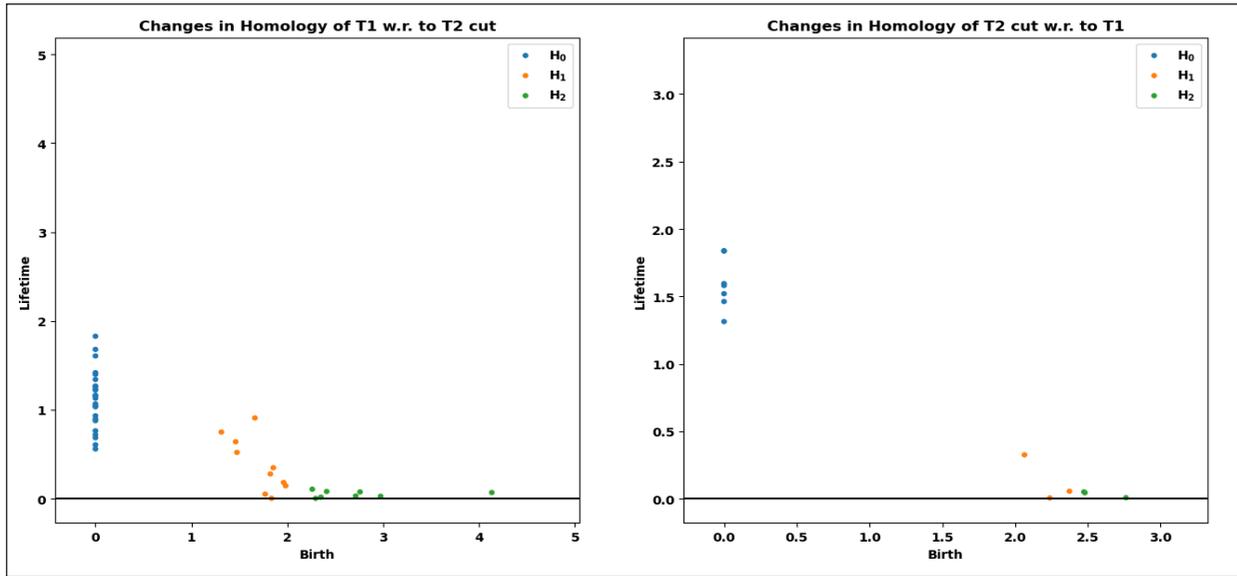
have been shown in Table 1. There is one  $H_0$  point that persists, four significant  $H_1$  points that persist for a certain lifetime greater than 1.5 and one significant  $H_2$  point that persists for a certain lifetime greater than 1.4 for *DT*. On the other hand, there is one  $H_0$  point that persists, and three significant  $H_1$  points that persist for a certain lifetime greater than 1.5, and two significant  $H_2$  points that persist for a certain lifetime greater than 1.4 for *NT* (Table 2).

From the above shreds of evidence overall shape of *DT* has been detected as a double torus with one significant void in common, i.e., it has a hollow inside the intersection and two significant voids of *NT* indicating there is a block inside the intersection that separated the void into two. Though overall estimation has been made, selecting significant points is still unclear. To be honest, the

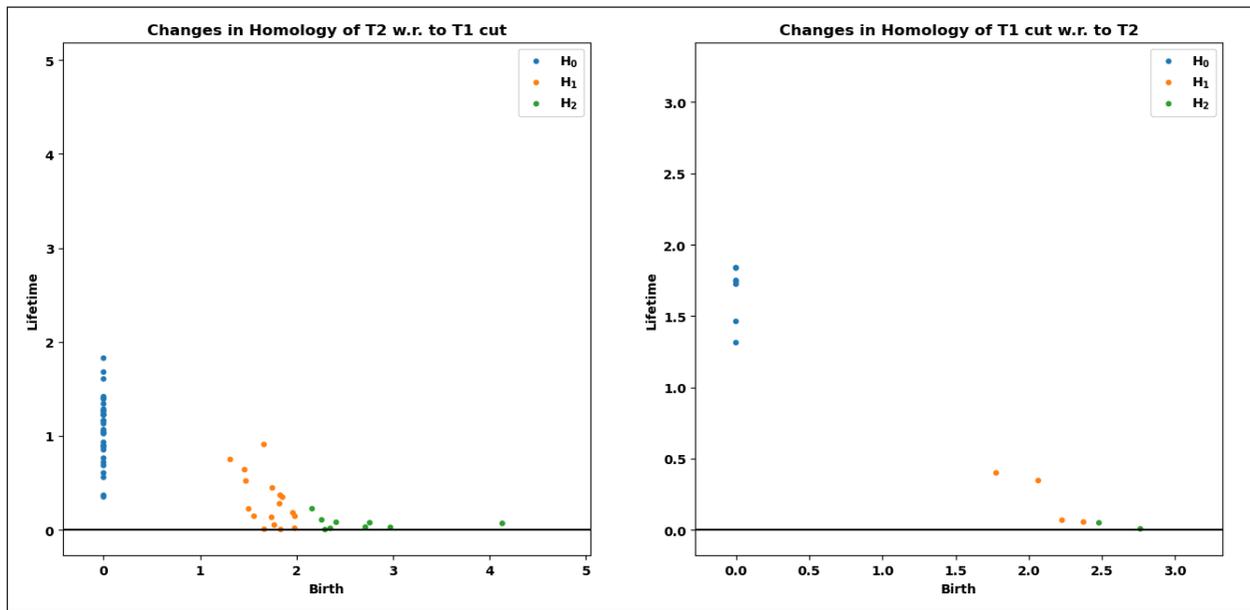
significant thresholds have been chosen considering the known shape of *DT* or *NT*. Therefore more strong evidence is needed to describe the hollowness of *DT*.

The data generation process followed in this study is actually a step-by-step evolution. In this whole process, a small number of points of two given tori have been changed. That is, most of the given points remained unchanged. Since the homology of some specific data will always be the same, the homology of unchanged data points should be the same throughout the generation process. Differences between the two homologies have been introduced to eliminate unchanged data points homology to tract changes of the first homology concerning the second homology. In this study, eight cases have been made which have been discussed below:

**Case 1.**  $T_1 \setminus T_2$  cut and  $T_2$  cut  $\setminus T_1$ .



**Figure 12.** Homologies of  $T_1 \setminus$ homologies of  $T_2 \text{ cut}$  and homologies of  $T_2 \text{ cut} \setminus$ homologies of  $T_1$ .



**Figure 13.** Homologies of  $T_2 \setminus$ homologies of  $T_1 \text{ cut}$  and homologies of  $T_1 \text{ cut} \setminus$ homologies of  $T_2$ .

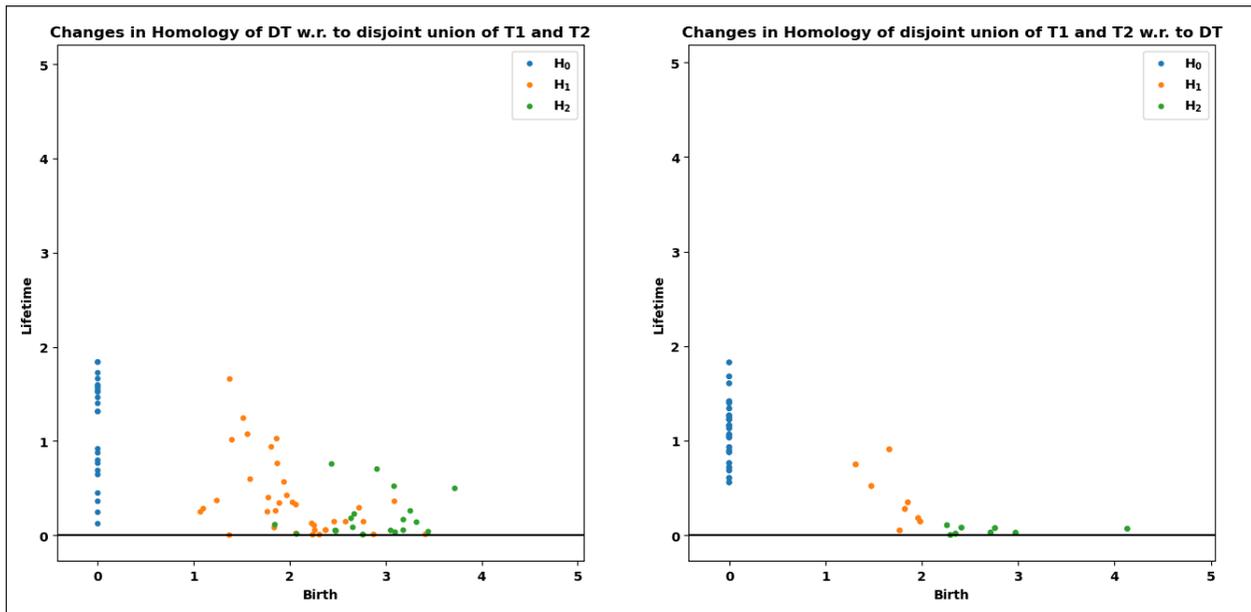
**Case 2.**  $T_2 \setminus T_1 \text{ cut}$  and  $T_1 \text{ cut} \setminus T_2$ .

In these cases, homologies of  $T_1 \setminus$ homologies of  $T_2 \text{ cut}$  produced those homologies of  $T_1$  that have been changed in  $T_2 \text{ cut}$  (Fig. 12). Due to some missing points of  $T_1$  in the  $T_2 \text{ cut}$  some loops connecting those points in  $T_1$  are missing in the homologies of  $T_2 \text{ cut}$ . The loop found at 1.6624 is significant among them. Some nonsignificant voids have been found enclosed by those loops. Similar changes have been noted in the

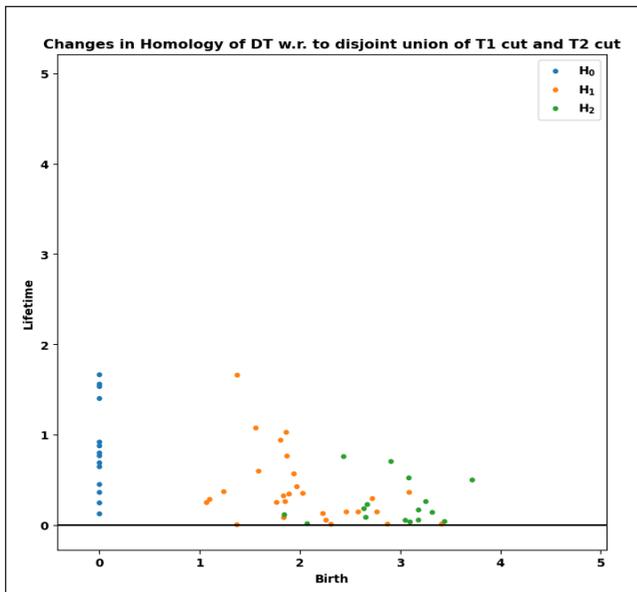
homologies of  $T_2$  with respect to the homologies of the  $T_1 \text{ cut}$  (Fig. 13).

On the other hand, homologies of  $T_2 \text{ cut} \setminus$ homologies of  $T_1$  produced those homologies of  $T_2 \text{ cut}$  that have been changed in  $T_1$  (Fig. 12). Few loops have been found around the cut of  $T_2 \text{ cut}$  since it creates a hole inside. Similar changes have been noted in the homologies of  $T_2$ . The significant loop among them has been found at 1.7776 (Fig. 13).

**Case 3.** (disjoint union of  $T_1$  and  $T_2$ )  $\setminus$  DT and DT  $\setminus$  (disjoint union of  $T_1$  and  $T_2$ ).



**Figure 14.** Homologies of disjoint union of  $T_1$  and  $T_2 \setminus$  homologies of  $DT$  and homologies of  $DT \setminus$  homologies of disjoint union of  $T_1$  and  $T_2$ .



**Figure 15.** Homologies of  $DT \setminus$  homologies of disjoint union of  $T_1 \text{ cut}$  and  $T_2 \text{ cut}$ .

**Case 4.**  $(\text{disjoint union of } T_1 \text{ cut and } T_2 \text{ cut}) \setminus DT$  and  $DT \setminus (\text{disjoint union of } T_1 \text{ cut and } T_2 \text{ cut})$ .

Taking the difference between the homologies of the disjoint union of  $T_1$  and  $T_2$  and the homologies of  $DT$ , few loops have been found due to the absence of some points of  $T_1$  and  $T_2$  in  $DT$  which are similar to **Case 1** and **Case 2**. A significant loop has been found in 1.6624 (Fig. 14). There are no changes

found in the disjoint union of the  $T_1 \text{ cut}$  and  $T_2 \text{ cut}$  concerning  $DT$ .

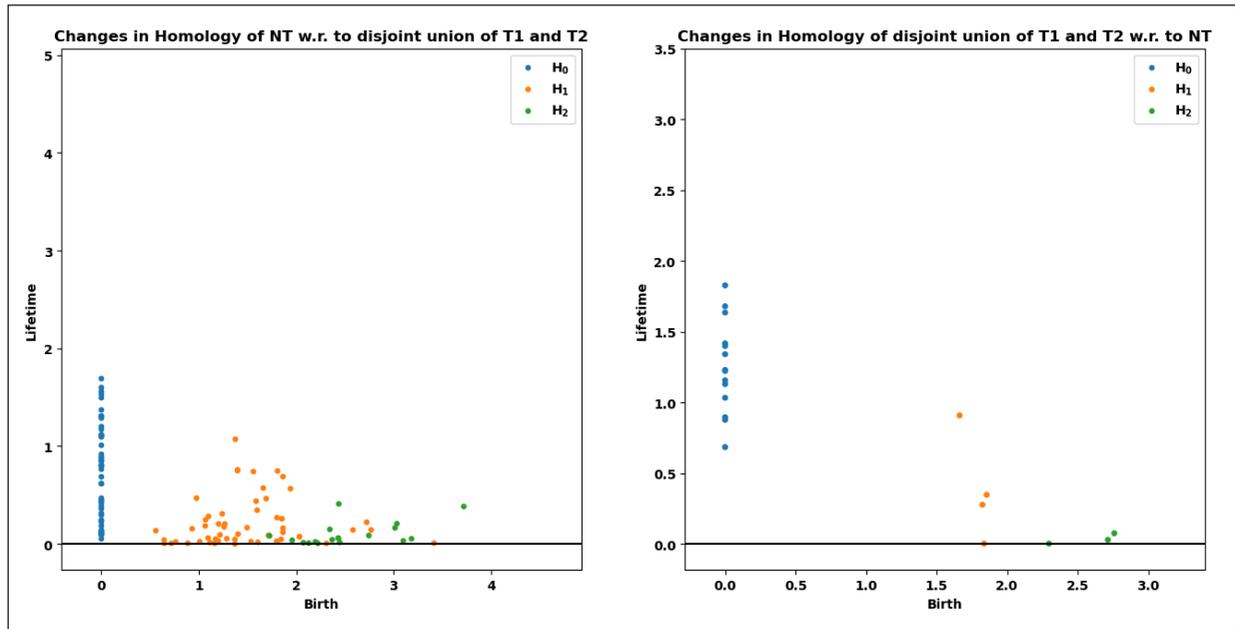
The difference between the homologies of  $DT$  and homologies of the disjoint union of  $T_1$  and  $T_2$  indicates changes in the homologies of  $DT$  concerning homologies  $T_1$  and  $T_2$ . In this case, two significant loops around the intersection of  $T_1$  and  $T_2$  have been found at 1.3749 and 1.8637. Also, two significant voids have been found enclosed by these two loops at 2.4365 and 2.9090 (Fig. 14). A similar result has been found taking the difference between the homologies of  $DT$  and homologies of the disjoint union of the  $T_1 \text{ cut}$  and  $T_2 \text{ cut}$  (Fig. 15).

**Case 5.**  $(\text{disjoint union of } T_1 \text{ and } T_2) \setminus NT$  and  $NT \setminus (\text{disjoint union of } T_1 \text{ and } T_2)$ .

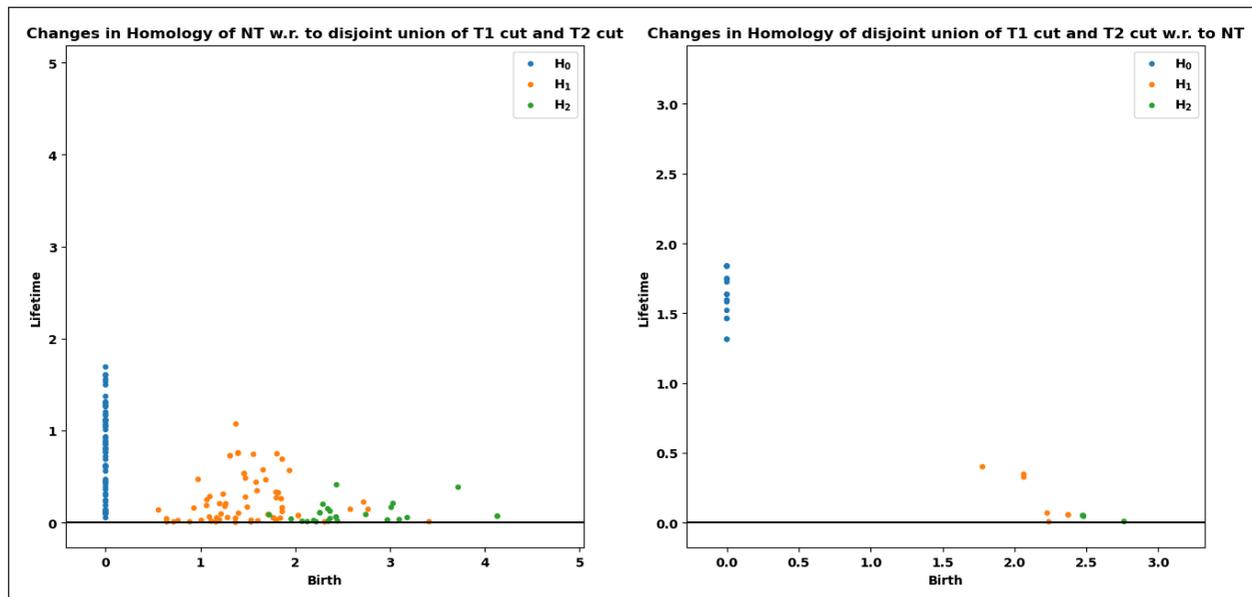
**Case 6.**  $(\text{disjoint union of } T_1 \text{ cut and } T_2 \text{ cut}) \setminus NT$  and  $NT \setminus (\text{disjoint union of } T_1 \text{ cut and } T_2 \text{ cut})$ .

Some loops have been found taking differences between a) homologies of disjoint union of  $T_1$  and  $T_2$  and homologies of  $NT$ ; b) homologies of disjoint union of  $T_1 \text{ cut}$  and  $T_2 \text{ cut}$ , and homologies of  $NT$  which are similar to **Case 1**, **Case 2**, **Case 3**, and **Case 4** as expected.

But differences between a) homologies of  $NT$  and homologies of the disjoint union of  $T_1$  and  $T_2$



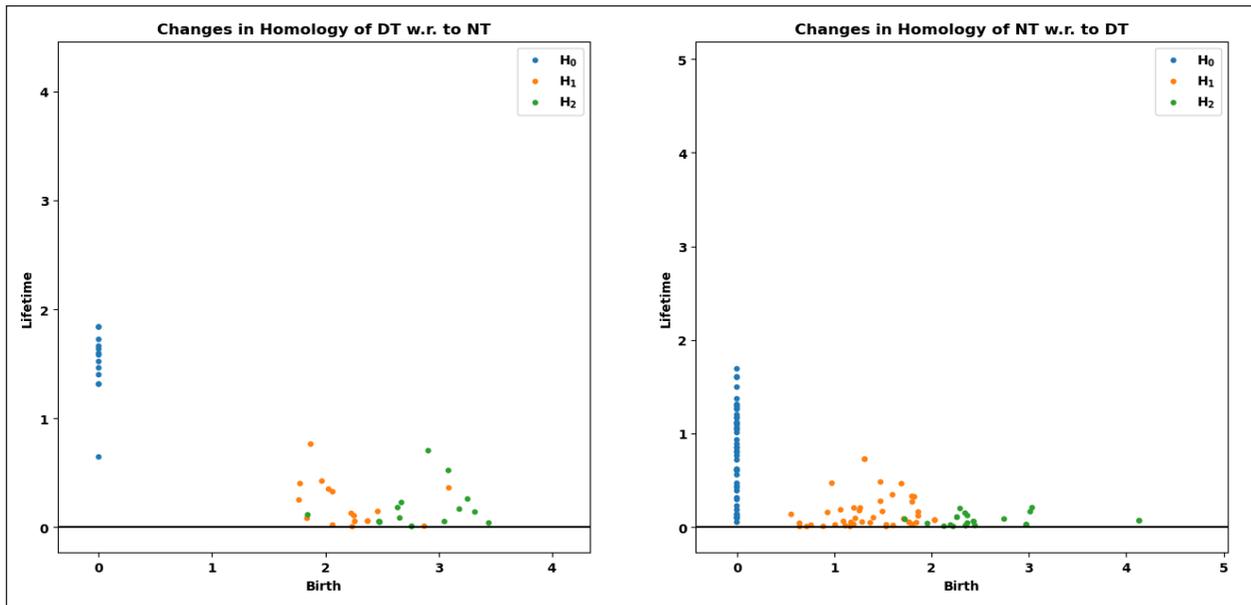
**Figure 16.** Homologies of  $NT \setminus$  homologies of disjoint union of  $T_1$  and  $T_2$ , and homologies of disjoint union of  $T_1$  and  $T_2 \setminus$  homologies of  $NT$ .



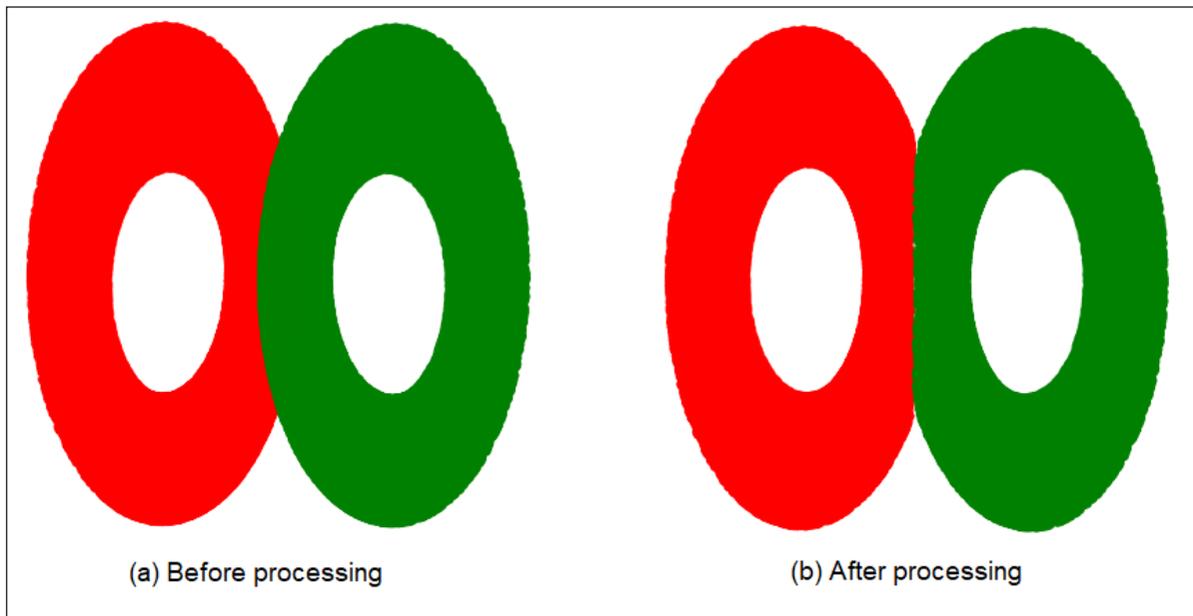
**Figure 17.** Homologies of  $NT \setminus$  homologies of disjoint union of  $T_1 \text{ cut}$  and  $T_2 \text{ cut}$  and homologies of disjoint union of  $T_1 \text{ cut}$  and  $T_2 \text{ cut} \setminus$  homologies of  $NT$ .

(Fig. 16), b) homologies of  $NT$  and homologies of disjoint union of  $T_1 \text{ cut}$  and  $T_2 \text{ cut}$  (Fig. 17) showed a greater number of loops than **Case 3** and **Case 4** since the intersection of  $T_1$  and  $T_2$  is not hollow in  $NT$ . Among all of these loops, four significant loops have been detected. Two of them are similar to  $DT$  around the intersection at 1.3749 and 1.8637. The

remaining two other significant loops are due to the presence of some points of  $T_1$  and  $T_2$  inside their intersection at 0.5586 and 1.0995. There are a few voids that are not significant compared to voids of  $DT$  found in **Case 3** and **Case 4**. A significant loop at 1.6624 has been detected in the homology of the disjoint union of  $T_1$  and  $T_2$  different from the homology of  $NT$ .



**Figure 18.** Homologies of  $DT \setminus$  homologies of  $NT$  and homologies of  $NT \setminus$  homologies of  $DT$ .



**Figure 19.** The generated double torus data considering each torus of 50,000 random uniform points. (a) Before processing. (b) After processing.

**Case 7.**  $NT \setminus DT$ .

**Case 8.**  $DT \setminus NT$ .

In these cases, the difference between the homologies of  $NT$  and homologies of  $DT$  showed two significant loops due to having points on the intersection at 0.5586 and 1.0995 (Fig. 18). The other two significant loops of  $NT$  found in **Case 5** and **Case 6** are similar to  $DT$  and so they have been removed.

However, the difference between the homologies of  $DT$  and the homologies of  $NT$  (Fig. 18) showed one significant loop around the intersection at 1.8712 since other loops found in **Case 5** and **Case 6** is similar to  $NT$ . Also, two significant voids have appeared at 2.9090 and 3.0870 since there are no significant voids in  $NT$ .

Analyzing all eight cases, we have found the following significant results:

1. Presence of some points of  $T_1$  and  $T_2$  inside their intersection created two more significant loops and some non-significant voids discussed in **Case 5** and **Case 6**.
2. Due to the hollowness of the intersection of  $T_1$  and  $T_2$  in  $DT$ , two significant voids have been created with two significant loops around it discussed in **Case 3** and **Case 4**.
3. There are some dissimilar significant voids and one dissimilar significant loop of  $DT$  than  $NT$  has been detected in **Case 3** and **Case 4** which ensure that  $DT$  has hollowness inside comparing with  $NT$ .
4. Changes in connected components of all cases are found helpful to detect significant loops by following gaps among themselves. It seems that there is a relationship between them. Since this is a new phenomenon, the relation among connected components, loops, and voids should be investigated in more details.

Thus the problem of generating a double torus has been encountered. The double torus data has been generated considering 50,000 random uniform points from a torus. The generated double torus data before and after processing has been shown in Figure 19.

### Conclusion and Further Research

In the field of data generation, double-torus data are useful for testing algorithms and models in a variety of applications, such as topology optimization, computer-aided design, computer graphics, image processing, and machine learning. However, generating double torus data needs more care concerning its topological properties. In this study, we proved theoretically that a torus  $T(R,r)$  can be deformed into a 3-sphere  $S^3(R+r)$  centered at the center of the torus  $T(R,r)$  in the **Theorem 1**. As a result, the **Lemma 1** has been derived that shows a 3-sphere  $S^3(T(R+r))$  is the smallest ball to cover the torus  $T(R,r)$ . Thus, a double torus data generation algorithm has been proposed with an approximation of torus by a sphere theoretically in the "Other Tools to Validate this Study" section.

After that, the Ripser [43] package was used to check the number of holes as a topological invariant of a 3D torus and a 4D torus generated from their parametric equations (1) and (2), respectively, to compare with the known knowledge of the shape of a torus. Following a cross-check of

their PDs, we decided to employ 4D torus data for this investigation. Then, following the theoretical findings, a routine has been developed in Python to approximate the torus-torus intersection into a torus-sphere intersection. Thus, torus-torus intersection points have been calculated from the union of two 4D torus data to extract it by following the procedure mentioned in the "Methodology" section. Finally, double torus data has been calculated by separating intersection points from the union of two 4D torus. After that, the generated data was validated by following persistence in the calculated number of zero, one, and two-dimensional holes using Ripser. Finally, changes in the homology of each step of the generation process have been investigated in "Results and Discussion" to explain the hollowness of the intersection of our generated double torus data.

It is worth noting that this method is an approximation of torus-torus intersection points, as the two tori are represented by a discrete set of points, not a continuous mathematical surface and the sphere representation of the torus is just an approximation of the true surface. Additionally, this method only works if the tori are convex ( $R > r$ ), as for concave tori ( $R < r$ ), the intersection will not be spherical. The generated double torus data must have better quality since it preserves the proper topology of a double torus. The proposed algorithm might be a good alternative to generate topology preserving double torus data that is suitable to use in testing algorithms and models of relevant applications. The generated double torus data can also be used to visualize double torus data after taking a 3D projected view of it. Additionally, conducting the entire process for 3D torus data is not complicated. It is also possible to reconstruct an irregular double torus considering two dissimilar torus in **Step 1** of this generation process discussed in the "Methodology" section.

There are several possible applications of this study:

- (a) **Visualization:** Different mathematical (topological) surfaces can be visualized using a similar approach to this study. The most closely related topological surfaces include the 3-torus, 4-torus, and others.
- (b) **Validation:** The validation approach using persistent diagrams can be employed to verify whether any data structure is topologically correct.

- (c) **Digital art:** Different digital artworks based on the torus can be created using the concept of the proposed data processing approach.
- (d) **Hollow double torus as synthetic data:** The generated double torus data can be used as synthetic data to validate models in various fields of study, particularly in computational geometry and topology.
- (e) **Data generation:** In a broader sense, this study serves as an example of data generation concerning topological properties. Studies that require topologically sound synthetic data to validate their models may find this study particularly relevant to their interests.

### Acknowledgments

The authors express utmost thanks to the Almighty Allah first for keeping us in such a condition to work on this investigation and then acknowledge fund providers: The Ministry of Science and Technology, Bangladesh (MoST), and the Dean, Faculty of Science, University of Rajshahi, Bangladesh. The authors acknowledge all those researchers in this field of study for their unpayable efforts.

### Funding

Md. Morshed Bin Shiraj is funded by the National Science and Technology Fellowship issued from the Ministry of Science and Technology, The Government of Bangladesh with GOno.39.00.0000.012.002.06.21.179, Md. Masum Murshed is funded by the Dean of Science faculty, University of Rajshahi, Rajshahi-6205, Bangladesh, with grant no. A-164/5/52/RU./Science-05/2021-2022 and Nasima Akhter is funded by the Dean of Science faculty, University of Rajshahi, Rajshahi-6205, Bangladesh, with grant no. A-160/5/52/RU./Science-04/2021-2022.

### Data availability

The hollow double torus data generated from this study is available at <https://data.mendeley.com/datasets/gxdmbnmbvh/1>.

### Conflict of interest

The authors declare that they have no conflicts of interest related to this research.

### References

- Lorenzo G, Scott MA, Tew K, Hughes TJR, Zhang YJ, Liu L, et al. Tissue-scale, personalized modeling and simulation of prostate cancer growth. *Proc Natl Acad Sci* 2016; 113:E7663–71.
- Nagano T, Yagi K. The affine structures on the real two-torus. i. *Osaka J Math* 1974, 11:181–210.
- Myers D. Homeomorphisms on the solid double torus. *Can J Math* 1975; 27:797–804.
- Kuusalo T, N'at'anen M. Geometric uniformization in genus 2. *Ann Acad Sci Fenn Series A.I. Math* 1995; 20:401–18.
- Kr'al D, Stehl'ik M. Coloring of triangle-free graphs on the double torus. *SIAM J. Discrete Math* 2008; 22:541–53.
- Kocay WL, Gagarin A. Embedding  $K_{3,3}$  and  $K_5$  on the Double Torus. *arXiv:2203.01150* 2022.
- Meijer DKF, Geesink HJH. Consciousness in the universe is scale invariant and implies an event horizon of the human brain. *Neuroquantology* 2017; 15:41–79.
- Meijer D. Processes of science and art modeled as a holoflux of information using toroidal geometry. *Open J Philos* 2018; 8:365–400.
- Patsis PA, Katsanikas M. The phase space of boxy-peanut and Xshaped bulges in galaxies—II. The relation between face-on and edge-on boxiness. *Monthly Notices Royal Astron Soc* 2014; 445(4):3546–56.
- Kramer P, Lorente M. The double torus as a 2d cosmos: groups, geometry and closed geodesics. *J Phys AMath General* 2002; 35:1961–81.
- Haramein N, Rauscher EA. Spinors, twistors, quaternions and the "spacetime" torus topology. In: *International Journal of Computing Anticipatory Systems*, D. Dubois (ed.). Institute of Mathematics, Liege University, Liège, Belgium, 2007. Available via [https://www.researchgate.net/publication/242072127\\_SPINORS\\_TWISTORS\\_QUATERNIONS\\_AND\\_THE\\_SPACETIMETORUS\\_TOPOLOGY](https://www.researchgate.net/publication/242072127_SPINORS_TWISTORS_QUATERNIONS_AND_THE_SPACETIMETORUS_TOPOLOGY)
- L'u J, Chen G. Generating multiscroll chaotic attractors: theories, methods and applications. *Int J Bifurcation Chaos* 2006; 16:775–858.
- Yu S, Lu J, Chen G. Multifolded torus chaotic attractors: design and implementation. *Chaos: An Interdis J of Nonlinear Sci* 2007; 17(1):013118.
- Yu S, Lu J, Chen G. Theoretical design and circuit implementation of multidirectional multi-torus chaotic attractors. *IEEE Trans Circuits Syst I Regular Papers* 2007; 54:2087–98.
- Chae S, Kim J, Hong S, Lee S. Design and analysis of the dual-torus network. *New Gener Comput* 1999; 17:229–54.
- Cheng S, Zhong W, Isaacs KE, Mueller K. Visualizing the topology and data traffic of multi-dimensional torus interconnect networks. *IEEE Access* 2018; 6:57191–204.
- Qin H, Carner C, Jin M, Gu X. Topology-driven surface mappings with robust feature alignment. In:

- VIS 05. IEEE Visualization, Minneapolis, MN, pp 43–50, 2005.
18. Schonsheck S, Chen J, Lai R. Chart auto-encoders for manifold structured data. arXiv: 1912.10094; 2019.
  19. Bourakna AEY, Chung MK, Ombao H. Modeling and simulating dependence in networks using topological data analysis. arXiv:2209.10416 2022.
  20. Silva VD, Vejdemo-Johansson M. Persistent cohomology and circular coordinates. In: Proceedings of the Twenty-Fifth Annual Symposium on Computational Geometry (SCG); pp 227–236, 2009.
  21. Patrikalakis NM, Maekawa T. Chapter 25: Intersection problems. In: Farin G, Hoschek J, Kim MA (eds.). Handbook of computer aided geometric design, 1st edition, North-Holland: Elsevier Science B.V., Amsterdam, The Netherlands, pp 623–50, 2002.
  22. Ugail H. Introduction to geometric design. In: Partial differential equations for geometric design, Springer, London, UK, pp 9–20, 2011, doi: HYPERLINK "[https://doi.org/10.1007/978-0-85729-784-6\\_2](https://doi.org/10.1007/978-0-85729-784-6_2)" 10.1007/978-0-85729-784-6\_2
  23. Abdel-Malek K, Yeh HJ. Determining intersection curves between surfaces of two solids. Computer Aided Design 1996; 28:539–49.
  24. Alsaidi R, Musleh A. Two methods for surface/surface intersection problem comparative study. Int J Comput Appl 2014; 92:1–8.
  25. Bajaj CL, Hoffmann CM, Lynch RE, Hopcroft J. Tracing surface intersections. Comput Aided Geometric Design 1988; 5:285–307.
  26. Farin G, Hoschek J, Kim MS. Handbook of computer aided geometric design. Elsevier Science B.V., Amsterdam, The Netherlands; 2002.
  27. Lee KY, Cho DY, Kim TW. A tracing algorithm for surface-surface intersections on surface boundaries. J Comput Sci Technol 2002; 17:843–50.
  28. Martin S, Watson JP. Non-manifold surface reconstruction from highdimensional point cloud data. Comput Geomet 2011; 44:427–41.
  29. Grandine TA, Klein IV FW. A new approach to the surface intersection problem. Comput Aided Geometric Design 1997; 14:111–34.
  30. Chau S, Galligo A. Topology of the intersection of two parameterized surfaces, using computations in 4D space. In: Dokken, T, Muntingh, G. (eds.) SAGA—advances in shapes, geometry, and algebra. geometry and computing, Springer International Publishing, Cham, 2014; 10:123–45.
  31. Hohmeyer ME. A surface intersection algorithm based on loop detection. In: Proceedings of the First ACM Symposium on Solid Modeling Foundations and CAD/CAM Applications (SMA); pp 197–207, 1991.
  32. Hur S, Oh MJ, Kim TW. Classification and resolution of critical cases in Grandine and Klein’s topology determination using a perturbation method. Computer Aided Geomet Design 2009; 26:243–58.
  33. Ma Y, Lee YS. Detection of loops and singularities of surface intersections. Computer Aided Geometric Design 1998; 30:1059–67.
  34. Sederberg TW, Meyers RJ. Loop detection in surface patch intersections. Computer Aided Geomet Design 1988; 5:161–71.
  35. Wu ST, Al’essio O. Complete and non-overlapping marching along a closed regular intersection curve. Comput Graph 2002; 26:853–64.
  36. Ying-xiu O, Min T, Jun-cheng L, Jin-xiang D. Intersections of two offset parametric surfaces based on topology analysis. J Zhejiang Univ-Sci 2004; 5:259–68.
  37. Liu XM, Liu CY, Yong JH, Paul JC. Torus/torus intersection. Computer-Aided Design Appl 2011; 8:465–77.
  38. Kim KJ. Circles in torus–torus intersections. J Comput Appl Math 2012; 236:2387–97.
  39. Paul R, Chalup SK. A study on validating non-linear dimensionality reduction using persistent homology. Pattern Recogn Lett 2017; 100:160–6.
  40. Zomorodian AJ. Topology for computing. Cambridge Monographs on Applied and Computational Mathematics, Cambridge University Press, Cambridge, UK; 2005.
  41. Hatcher A. Algebraic topology. Cambridge University Press, Cambridge, UK; 2002.
  42. Singh GKC, M’emoli F, Carlsson GE. Topological methods for the analysis of high dimensional data sets and 3d object recognition. In: 4th symposium on point based graphics, PBG@Eurographics, Prague, Czech Republic, 2007.
  43. Tralie C, Saul N, Bar-On R. Ripser.py: A lean persistent homology library for python. J Open Source Software 2018; 3(29):925.