No Flat Polyhedral Tori with 7 Vertices

Richard Evan Schwartz *

July 15, 2025

Abstract

We prove that there does not exist a piecewise affine isometric embedding of a flat torus into $\boldsymbol{R}^3$ whose image has 7 veertices.

1 Introduction

1.1 Context

A *flat polyhedral torus* is a piecewise affine isometric embedding

$$\phi : T \to \Omega \subset \boldsymbol{R}^3.$$

Here T is a flat torus, the quotient of $\boldsymbol{R}^2$ by a lattice of translations, and ϕ preserves arc lengths of curves. The map ϕ tells how to build a torus out of finitely many triangles so that the cone angle around each vertex is 2π.

Surprisingly, flat polyhedral tori exist. The 1960 paper of Y. Burago and V. Zalgaller [**BZ1**] gives the first constructions. The 1985 paper [**BZ2**] proves that one can realize every isometry class of flat torus as some flat polyhedral torus. Their construction produces examples with thousands of faces. The 2024 preprint of F. Lazarus and F. Tallerie [**LT**] gives a universal triangulation, with thousands of faces, which does the job simultaneously for all isometry types.

The work (in progress) [**ALM**] of P. Arnoux, S. Lelievre, and A. Malaga gives a different construction which achieves every isometry class of flat polyhedral tori using far fewer faces than in the other cited works. Specifically,

* Supported by N.S.F. Research Grant DMS-2505281

they prove that the simple and elegant flat tori described by Ulrich Brehm [**Br**] in 1978 can be made flat. I will leave it to those authors to tell that story.

In 2025, Vincent Tugayé [**T**], discovered a 9-vertex flat polyhedral torus. (Tugayé is one of Lelievre's housemates, a physics teacher with a PhD in physics and an interest in mathematics.) This beautiful example seems like it might realize the minimum. We conjecture that the minimum number of vertices needed for a flat polyhedral torus is 9.

1.2 The Seven Vertex Case

Let us discuss the 7-vertex case. The minimum number of triangles one needs to triangulate a torus is 7, and there is only one such triangulation up to combinatorial isomorphism. This triangulation is shown in Figure 1. The triangulation gives the famous embedding of the complete graph K_7 in a torus. This is often called the *Moebius Torus*. We are drawing part of the universal cover as well as a fundamental domain consisting of 7 hexagons.

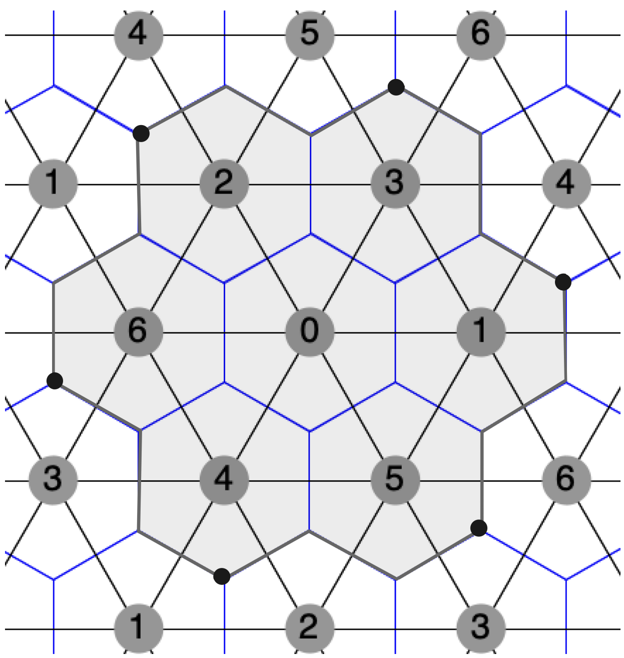

Figure 1: The 7-vertex triangulation of a torus

In 1949, Császár [**Cs**] showed that the Moebius torus has a polyhedral embedding. Császár's polyhedral embedding Ω is specified by giving coordinates for the vertices $P_0, ..., P_6$. With our labeling, the coordinates are

$$(0,0,15), (3,3,0), (-3,-3,1), (-1,-2,3), (1,2,3), (3,3,1), (3,-3,0).$$

This torus has 2-fold rotational symmetry and its convex hull contains 5 of the 7 vertices. See e.g. [**G**], [**HLZ**], and [**L**] for more detail and a discussion of related topics. The Császár torus is not flat. Could some other embedding of the Moebius torus be flat?

The 1991 paper [**BE**] of J. Bokowski and A. Eggert gives an exhaustive study of the combinatorial types of 7-vertex polyhedral tori, but they do not address the question of flatness. In my opinion, had they done so they most likely would have been able to answer it. Anyway, here is the answer:

Theorem 1.1 (Main) *No flat polyhedral torus has 7 vertices.*

A key step in the proof of the Main Theorem is

Theorem 1.2 (Hull) *A 7-vertex polyhedral torus cannot have 7 vertices in its convex hull.*

This is one of the main results in [**BE**], though they did not name the result. See the comment after after [**BE**, Theorem 3.7]. The proof of the Hull Theorem in [**BE**] goes through the authors' complete classification of combinatorial types using oriented matroids. This is an extensive computer-assisted enumeration.

I will give an independent and much lighter (though still computer assisted) proof of the Hull Theorem. The computer part of the proof is combinatorial and uses exact calculations. I include the code in §3. The Java user interface I use to run the code, as well as all the code, can be downloaded at **www.math.brown.edu/∼res/Java/SevenTorus.tar**

1.3 Acknowledgements

I would like to thank the Samuel Lelievre and Alba Malaga-Sabogal for telling me all about flat tori (over a period of some years) and in particular telling me about the flatness problem and supplying me with some historical context. I would like to thank the IHES, the Hamilton Institute, and the Isaac Newton Institute, where I worked on this paper. I would like to thank chat gpt for help evaluating and formatting my computer code.

2 The Main Theorem

2.1 The Proof Modulo the Hull Theorem

Lemma 2.1 *If γ is a spherical polygonal loop of length at most 2π then γ lies in some hemisphere of S^2.*

Proof: This is well known. Let S^2 denote the round unit 2-sphere. The space of oriented great circles in S^2 is canonically bijective with S^2 itself, and inherits a canonical probability measure σ. Given a spherical polygon γ and a great circle σ, we let $\#(\sigma \cap \gamma)$ denote the number of intersection points. (We can ignore the finitely many great circles for which this is infinite.) Crofton's formula [**S**] says that

$$\text{length}(\gamma) = \pi \int_S \#(\sigma \cap \gamma)\ d\sigma. \tag{1}$$

Now suppose that γ is a spherical polygon of length less than 2π. Crofton's formula says that some great circle σ intersects γ at most once. But then γ lies in one of the two hemispheres defined by σ. When γ has length exactly 2π we can shorten γ a bit by cutting a small corner off. The shortened curve then lies in a hemisphere. Taking a limit as the cut corner tends to 0 in length, we see that γ itself lies in a hemisphere. ♠

Call a vertex of Ω *interior* if it does not lie in the convex hull boundary. By the Hull Theorem, each 7-vertex polyhedral torus has at least one interior vertex P. We show that the cone angle θ at P exceeds 2π.

Let L denote the union of rays which emanate from P and go through the other 6 vertices of Ω. Two rays of L are consecutive if they go through points which are adjacent in the combinatorial link of P. Let $\widehat{L}$ denote the union of acute convex sectors we get by filling in every pair of consecutive rays by the acute sector they bound. The intersection of $\widehat{L}$ with the unit S^2 centered at P is a spherical polygon γ whose length equals θ.

Since P is in the interior of the convex hull of Ω and since $\widehat{L}$ contains all the vertices of this convex hull, we see that $\widehat{L}$ cannot lie in any halfspace bounded by a plane through origin. This means that γ cannot lie in any hemisphere of S^2. By Lemma 2.1, we see that γ has length greater than 2π. But then $\theta > 2\pi$.

This completes the proof of the Main Theorem, modulo the Hull Theorem.

2.2 Proof of the Hull Theorem: Combinatorics

We suppose that Ω is an embedded 7-vertex polyhedral torus having 7 points on its convex hull H. We can perturb so that the points are in general position. This makes H into a triangulated solid polyhedron with 7 vertices, 15 edges, and 10 faces. One property we will use repeatedly and is that Ω is *neighborly*: Every two vertices of Ω are joined by an edge in the triangulation. In particular, every edge of H is an edge of the triangulation of Ω.

Of the 21 edges of Ω, exactly 15 lie in ∂H. We call these the *external edges*. We call the remaining 6 edges *internal edges*. We say that an *internal edge pattern* is a choice of 6 distinguished edges from the 1-skeleton of the triangulation, normalized (by symmetry) so that the first internal edge is (01). There are $\binom{20}{5} = 15504$ different internal edge patterns.

We say that an *external triangle* of Ω is one that lies in ∂H, and an *internal triangle* is one that does not. Each internal edge is incident to two internal triangles.

Lemma 2.2 *An internal triangle cannot be bounded by* 3 *external edges.*

Proof: Suppose that such a triangle exists. Call it τ. If all three edges of τ lie in ∂H and $\tau \not\subset \partial H$ then τ separates H into two components, both of which contain vertices of Ω in their interior. Any path in H connecting two such vertices must intersect τ. On the other hand $\Omega - \tau$ is path connected. This is a contradiction. ♠

By Lemma 2.2, the internal edge pattern determines the set of internal triangles and the set of external triangles. It is worth pointing out a triangle of ∂H is not necessarily a triangle of Ω. However, its boundary is one of the 35 three cycles contained in the triangulation of Ω.

Lemma 2.3 *All* 6 *internal edges cannot be incident to the same vertex.*

Proof: Suppose all the internal edges are incident to (0). Then the union of triangles incident to (0) is an embedded disk Δ which intersects ∂H in an embedded hexagon $\gamma = \partial\Delta$ and the isolated point P_0. But then Ω must be the union of Δ with the side of $\partial H - \gamma$ opposite (0). This is a topological sphere rather than a torus. This is a contradiction. ♠

Now we know that there is at least one external edge incident to each vertex.

Lemma 2.4 *If there are K internal edges incident to vertex P, then there are $6-K$ triangles in ∂H incident to P.*

Proof: Say that a *flag* is a pair (τ, P) where τ is a triangle of ∂H and P is a vertex of Ω. For vertex P_i there are K_i incident internal edges. Hence there are $6-K_i$ external edges incident to P_i. But that means there are at least $6-K_i$ flags involving P_i. Say there are $6-K_i+L_i$ such flags, with $L_i \geq 0$. This gives a total of

$$7 \times 6 - 2\sum K_i + 2\sum L_i = 30 + \sum L_i$$

flags. Since each triangle of ∂H participates in 3 flags this gives us a total of $10 + \sum L_i/3$ triangles. But there are exactly 10 triangles in ∂H. Hence $L_0 = ...L_6 = 0$. This means that there exactly $P_i - K_i$ flags involving P_i, which is equivalent to our claim. ♠

For each vertex (q) we have the list $\{(q, v_i) \mid i = 1, ..., K_q\}$ of K_q external edges incident to q. We write $K = K_q$ and order these $K = K_q$ vertices cyclically according to the link of (q) in Ω. The link of L_q of (q) in ∂H is some permutation of $(w_1, ..., w_K)$ of $(v_1, ..., v_K)$.

Lemma 2.5 (Cycle Rule) *In order to be a viable candidate for the link of (q) in ∂H, the link $(w_1, ..., w_K)$ must satisfy two properties.*

1. *(w_i, w_{i+1}) must be an external edge for all i. Indices are taken cyclically*
2. *The cycle $(w_1, ..., w_K)$ must be a dihedral permutation of $(v_1, ..., v_K)$.*

Proof: The necessity of Condition 1 is obvious. Condition 2 requires some explanation. Consider the picture in ∂H at (q). Since the points are in general position, ∂H is a proper convex cone near (q). We let Π be a plane parallel to a support plane through (q) and consider the intersection $\partial H \cap \Pi$. This is a convex (K)-gon Δ and the cyclic order of Δ is given by $(w_1, ..., w_K)$. At the same time, the link of (q) in Ω is some polygonal loop $\gamma \subset \Delta$ that visits the vertices in the order $(v_1, ..., v_K)$. If the permutation is not dihedral then γ cannot be an embedded loop. ♠

Using the Cycle Rule we eliminate all the possible remaining internal edge patterns except 6. These 6 are all the same up to combinatorial isometry.

2.3 Proof of the Hull Theorem: Geometry

Figure 2 shows one of the 6 remaining internal edge patterns. One of them corresponds to an embedded torus if and only all of them do. So, we just have to rule out this one.

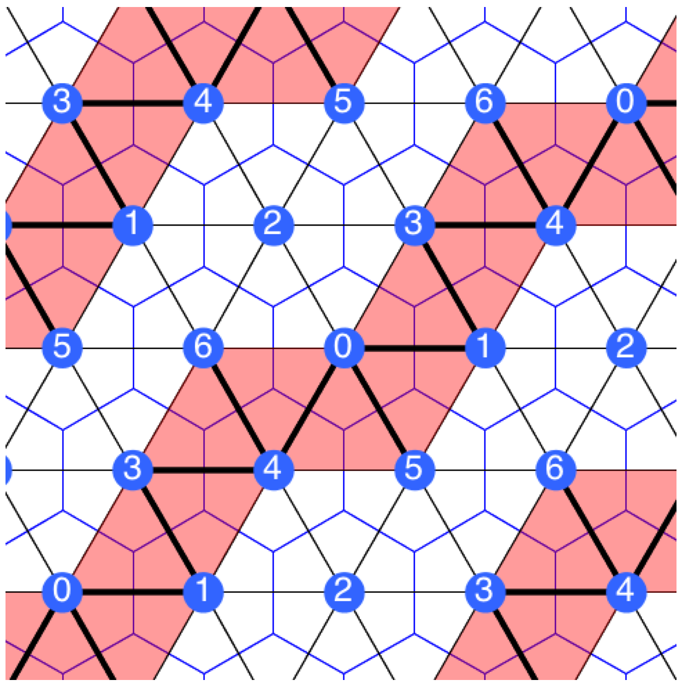

Figure 2: The one remaining pattern

Remark: We have an alternate ending of the proof of the Main Theorem which just uses the combinatorial result in the previous section. The reader who wants to get a full proof of the Main Theorem as fast as possible can skip the rest of this section and read the next section instead.

Suppose for the sake of contradiction that this pattern corresponds to an embedded torus Ω. We think of Ω as a subset of projective space $\boldsymbol{P}^3$.

We can apply a projective transformation so that (2) moves to the point $[0:0:1:0]$ at infinity in $\boldsymbol{P}^3$ in projective space. This point is "infinitely far away" along the Z-axis. We also can arrange that the (now) rays $(2j)$ start at (j) and move downward (rather than upwards) along the Z-axis, limiting on (2). We can do all this by a projective transformation that maps $H - \{(2)\}$ into the affine patch, which we identify with $\boldsymbol{R}^3$. Essentially, our normalization is a limit of examples in $\boldsymbol{R}^3$ in which (2) moves farther and farther down the Z axis.

Now, $H - \{(2)\}$ is still a convex subset in $\boldsymbol{R}^3$. It is like a hexagonal prism that has been truncated at one end. Here is the crucial observation: since H is convex, the projection of the hexagon (603541) into the XY-plane is a convex hexagon. We can further normalize so that the projections of (15) and (36) into the XY-plane are parallel line segments. We do this by mapping the line $[215] \cap [236]$ to a line at infinity which contains (2). Here $[abc]$ is the plane containing the points (a) and (b) and (c).

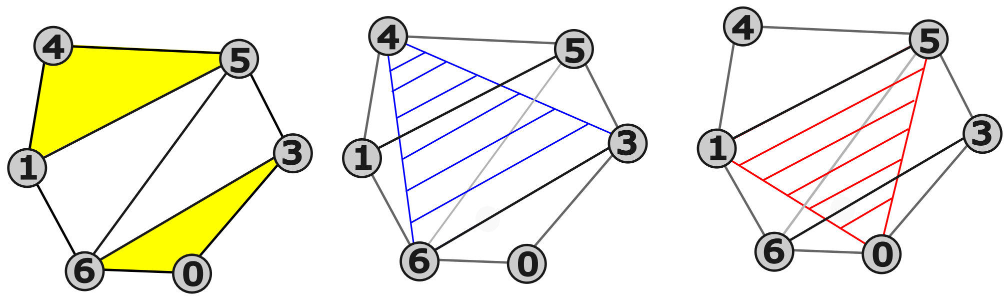

Figure 8: Projection of part of ∂H into the XY-plane

The 3-cycles (036) and (145) are both triangles on ∂H. These are not triangles of Ω. Every other triangle in ∂H does belong to $\partial\Omega$. By convexity, these two 3-cycles are bend downward. What we mean is that the plane containing (036) has the rest of H beneath it. The same goes for the plane containing (145).

The blue triangle in Figure 8 is (346). This is an internal triangle. We can foliate this triangle by parallel line segments as shown in the figure. These segments are parallel in space and they project to parallel segments in $\boldsymbol{R}^2$. The blue triangle in Figure 8 is (015). We make all the same constructions for this internal triangle. We call these foliations *red* and *blue*, as in the figure.

Say that a *special plane* is a plane in $\boldsymbol{R}^3$ whose projection to $\boldsymbol{R}^2$ is a line parallel to the projections of the red and blue foliations. One special plane contains (15). In this plane, the red foliation is above the blue foliation. Another special plane contains (36). In this plane, the blue foliation is above the red foliation. So, by the intermediate value theorem, there is a special plane for which these two foliations coincide. But then (015) and (346) intersect. This contradicts the fact that Ω is embedded.

This completes the proof of the Hull Theorem.

2.4 An Alternate Ending

Our proof of the Main Theorem does not require the full force of the Hull Theorem. All we need is the weaker result that no *flat* embedded 7-vertex polyhedral torus Ω has all 7 vertices in its convex hull boundary, ∂H.

The argument in §2.2 alone establishes the following result: If Ω has all 7 vertices in ∂H then a neighborhood in ∂H of one vertex P contains all 6 triangles of Ω incident to P. In other words, the link of P in Ω coincides with the link of Ω in H. We just have to prove that this property precludes the flatness of Ω.

Suppose that Ω is flat. Since H is also convex at P, we have some serious constraints on the link of P in Ω. First of all, H cannot be strictly convex at P. This means that some support plane through P intersects Ω in (at least) a line segment through P that contains P in its relative interior. But then the intersection of ∂H with a small sphere centered at P is a spherical polygon consisting of two arcs which connect the same pair of antipodal points. Using the fact that the cone angle of Ω at P is 2π we see (by Crofton's formula) that both these spherical arcs must be halves of great circles.

This means that there is a plane Π that contains at least 3 consecutive triangles of Ω incident to P. But then Π contains at least 5 vertices of Ω. Finally, this implies that Π contains an embedded copy of the complete graph K_5. This contradicts the fact that K_5 is not planar.

3 Java Code

The code for the proof of the Hull Theorem is distributed in two files. The first file helps manipulate lists. The second file has the actual tests.

3.1 The ListHelp Class

```java
import java.util.Arrays;

public class ListHelp {

    /**prints out an integer list*/
    public static void printout(int[] list) {
    if(list==null) return;
    for(int i=0;i<list.length;++i) System.out.print(list[i]+" ");
    System.out.println("");
    }

    /**checks if two lists match up to permutation*/
    public static boolean match(int[] a,int[] b) {
    if(a.length!=b.length) return false;
    int[] aa=Arrays.copyOf(a,a.length);
    int[] bb=Arrays.copyOf(b,b.length);
    Arrays.sort(aa);
    Arrays.sort(bb);
    for(int i=0;i<a.length;++i) {
        if(aa[i]!=bb[i]) return false;
    }
    return true;
    }

    /**checks if element a is amongst the
    first k elements of list b*/
    public static boolean onList(int a,int[] b,int k) {
    for(int i=0;i<k;++i) {
        if(a==b[i]) return true;
    }
    return false;
    }
```

```java
/**take an integer list, sorts it, and removes redundancies*/
public static int[] irredundantSortedList(int[] data) {
Arrays.sort(data);
int n=data.length;  int[] temp = new int[n];
int count = 0;
for (int i = 0; i < n; ++i) {
if ((i == 0) ||(data[i] != data[i - 1])) {
        temp[count] = data[i];
        ++count;
      }}
return Arrays.copyOf(temp, count);
}

/**Gives all the 6 element subsets of {0,...,20} having
  0 as the first element*/

public static int[] subsetGenerator(int index) {
int[] subset = {0,0,0,0,0,0};
int x = 1;

for (int i=1; i<6 ;++i) {
    while(choose(20-x,5-i)<=index) {
        index = index - choose(20 - x, 5 - i);
        ++x;
    }
    subset[i] = x;
    ++x;
}
return subset;
}

/**This returns n choose k.*/
public static int choose(int n,int k) {
  int x=1;int y=1;
for(int i=1;i<=k;++i) {
    x=x*(n-i+1);
    y=y*i;
}
return x/y;
}
```

```java
/**Gets the kth dihedral permutation of list a*/
public static int[] perDihedral(int[] a,int k) {
int n=a.length;
if(k<n) return cycle(a,k);
return reverse(cycle(a,k));
}

/**reverses list a*/
public static int[] reverse(int[] a) {
int n=a.length;
int[] b=new int[n];
for(int i=0;i<n;++i) b[i]=a[n-i-1];
return b;
}

/**cycles list a by k clicks*/
public static int[] cycle(int[] a,int k) {
int n=a.length;
int[] b=new int[n];
for(int i=0;i<n;++i) b[i]=a[(i+k)%n];
return b;
}
```

3.2 The LinkAnalyzer Class

```
import java.util.Arrays;

public class LinkAnalyzer {

/**This class performs all the tests for the proof of the
   Hull Theorem in our paper. The file works with ListHelp.java
   to manipulate lists*/

/**Returns kth edge of the complete graph K7. This is also the
   1-skeleton of the 7-vertex triangulation of the torus*/
public static int[] edge(int k) {
  int[][] f={{0,1},{0,2},{0,3},{0,4},{0,5},{0,6},
             {1,2},{1,3},{1,4},{1,5},{1,6},
             {2,3},{2,4},{2,5},{2,6},
             {3,4},{3,5},{3,6},
             {4,5},{4,6},{5,6}};
    return f[k];
}

/**Returns kth face in the 7-vertex triangulation of the torus.*/
public static int[] face(int k) {
    int[][] f={{0,1,3},{0,5,1},{0,3,2},{0,2,6},{0,4,5},{0,6,4},
             {1,2,4},{1,4,3},{1,6,2},{1,5,6},{2,3,5},{2,5,4},
             {3,4,6},{3,6,5}};
    return f[k];
}

/**This gets the links of each vertex in the torus*/

    public static int[] torusLink(int k) {
      int[][] L={{1,3,2,6,4,5},{0,5,6,2,4,3},{0,3,5,4,1,6},
      {0,1,4,6,5,2},{0,6,3,1,2,5},{0,4,2,3,6,1},{0,2,1,5,3,4}};
    return L[k];
      }
```

```java
/**This gets the kth choice of 6 element subset of the edges and
   then returns the corresponding edges.*/
public static int[][] internalEdges(int k) {
    int[] t=ListHelp.subsetGenerator(k);
    int[][] list=new int[6][2];
    for(int i=0;i<6;++i) {
        list[i]=edge(t[i]);
    }
    return list;
}

/**This gets the triangles incident to the edge
list from the previous routine.*/
public static int[][] internalFaces(int k) {
    int[] t=ListHelp.subsetGenerator(k);
    int[] list1=new int[12];
    int count=0;
    for(int i=0;i<6;++i) {
        int[] ee=edge(t[i]);
        for(int j=0;j<14;++j) {
            if(incident(ee,face(j))==true) {
                list1[count]=j;
                ++count;
            }
        }
    }
    list1=ListHelp.irredundantSortedList(list1);
    int[][] list2=new int[list1.length][3];
    for(int i=0;i<list1.length;++i) list2[i]=face(list1[i]);
    return list2;}

/**Returns true if edge e is incident to face f.*/
public static boolean incident(int[] e,int[] f) {
    if(ListHelp.onList(e[0],f,3)==false) return false;
    if(ListHelp.onList(e[1],f,3)==false) return false;
    return true;}
```

```java
/**This routine picks a vertex k and returns all the
external edges that are incident to it.  We are
careful to maintain the correct cyclic order*/

      public static int[] convexLink(int[][] e,int k) {
      int[] L=torusLink(k);
    int[] list=new int[6];
    int count=0;
    for(int i=0;i<6;++i) {
        int[] ee={k,L[i]};
        boolean test=false;
        for(int j=0;j<6;++j) {
        if(ListHelp.match(ee,e[j])==true) {
          test=true;
          break;
        }
        }
        if(test==false) {
           list[count]=L[i];
           ++count;
        }
    }
    return Arrays.copyOf(list,count);
    }

/**Make sure that consecutive elements are not internal edges.*/
public static boolean onlyAllowedConnections(int[][] e,int[]
    ↪ cycle) {
    for(int i=0;i<cycle.length;++i) {
        int ii=(i+1)%cycle.length;
        int[] L={cycle[i],cycle[ii]};
        for(int j=0;j<6;++j) {
            if(ListHelp.match(L,e[j])==true) return false;
        }
    }
    return true;}
```

```java
/**This returns the link if it is viable
and otherwise returns null*/

    public static int[] getViableCycle(int k,int i) {
    int[][] edge=internalEdges(k);
    int count=0;
    int[] link=LinkAnalyzer.convexLink(edge,i);
    if(link.length<3) return null;
    if(onlyAllowedConnections(edge,link)==false) return null;
    return link;}

    /**Tests all the cycles associated to the kth internal
    edge pattern.  Returns true if they are all viable. */

    public static boolean mainTest(int[] filter,int k) {
    for(int i=0;i<7;++i) {
        int[] cyc=getViableCycle(k,i);
        if((filter[i]==1)&&(cyc==null)) return false;
    }
    return true;}

/**This final  test. This is what we run.*/
    public void bigTest() {
    int count=0;
    int[] f={1,1,1,1,1,1,1};
    for(int i=0;i<15504;++i) {
          if(mainTest(f,i)==true)  ++count;
    }
    System.out.println("count (should be 0) "+count);}

      }
```

4 References

[**ALM**] P. Arnoux, S. Lelievre, A Malaga, *Diplotori: a family of polyhedral flat tori.* in preparation (2021)

[**Br**], U. Brehm, Oberwolfach report (1978)

[**BE**] J. Bokowski and A. Eggert, *All realizations of Moebius' Torus with 7 Vertices*, Topologie Structurale, (1991)

[**BZ1**] Y. D. Burago and V. A. Zalgaller, *Polyhedral realizations of developments* (in Russian) Vestnik Leningrad Univ. 15, pp 66–80 (1960)

[**BZ2**] Y. D. Burago and V. A. Zalgaller, *Isometric Embeddings of Two Dimensional Manifolds with a polyhedral metric into* $\boldsymbol{R}^3$, Algebra i analiz 7(3) pp 76-95 (1995) Translation in St. Petersburg Math Journal (3)3, pp 369–385

[**Cs**] Á. Császár, "A polyhedron without diagonals," *Acta Sci. Math. (Szeged)*, vol. 13, pp. 140–142, 1949.

[**G**], M. Gardner, *Mathematical Games: On the remarkable Császár polyhedron and its applications in problem solving*, Scientific American **232**, 5 (1975) pp 102-107.

[**HLZ**], S. Hougardy, F H. Lutz, and M. Zelke, *Polyhedral Tori with Minimal Coordinates*, arXiv 0709.2794

[**LT**] F. Lazarus, F. Tallerie, *A Universal Triangulation for Flat Tori*, CS arXiv 2203.05496 (2024)

[**L**] F. H. Lutz, *Császár's Torus*, Electronic Geometry models (2002) 2001.02.069

[**S**] L. A. Santalo, *Introduction to Integral Geometry* (1953)

[**T**] V. Tugayé, personal communication (2025)