Filomat 38:9 (2024), 3041–3053
https://doi.org/10.2298/FIL2409041M

Published by Faculty of Sciences and Mathematics,
University of Niš, Serbia
Available at: http://www.pmf.ni.ac.rs/filomat

Connection graphs for 3-triangulations of toroids

Nada Mladenović[a], Milica Stojanović[a]

[a] *University of Belgrade, Faculty of Organizational Sciences,
154 Jove Ilića Street, 11040 Belgrade, Serbia*

Abstract. It is known that all convex polytopes admit a triangulation without additional vertices, however this does not hold in general for non-convex polyhedra. If 3-triangulation of some polyhedron is possible, then a connection graph is introduced in such a way that convex pieces of that polyhedron are represented by graph nodes.

A method for constructing a polyhedron P based on a given connected undirected graph is given and the properties of P are investigated. The algorithms for computing the numbers of vertices, edges, faces and handles of P are also given.

1. Introduction

By definition a 3-triangulation of a (generally non-convex) polyhedron in the 3-space is its subdivision into tetrahedra, which uses only the original vertices. It is generalization of classical triangulation where a polygon with n vertices is divided by $n-3$ diagonals into $n-2$ triangles. Analogously, it is also possible to define a d-triangulation in higher ($d \geq 4$) dimensions as a partition of a d-dimensional polytope into d-dimensional simplices.

As in the case of 2-dimensional space, different types of polyhedra decompositions have significant applications in engineering and other fields of research. For example, [24] is devoted to an application of 3-triangulation, while [22] and [23] shows an application of another polyhedron decomposition and space modeling.

But, in higher dimensions ($d \geq 3$) the triangulation process is more complicated and new problems arise. Thus, different 3-triangulations of the same polyhedron can have different numbers of tetrahedra [4], [8], [10], [15], [16].

It is shown (Theorems 2.3 - 2.6 in [4]) that the smallest and largest number of tetrahedra in a 3-triangulation (the minimal and the maximal 3-triangulation) depend linearly, i.e. squarely on the number of vertices.

The next problem concerns the possibility of 3-triangulation. Similar to 2-triangulation we can triangulate any convex polyhedron, but, contrary to the 2-dimensional case, there are exceptions for some non-convex polytopes. The most famous non-triangulable polyhedron is the example of Schönhardt [7]. However, equally well-known Császár non-convex polyhedron [2] is 3-triangulable. That is an example of

2020 *Mathematics Subject Classification*. Primary 52C17; Secondary 68R10, 52B05, 05C85, 68P05, 68Q65

Keywords. triangulation of polyhedra, toroids, piecewise convex polyhedra, connection graph, graph algorithms, abstract data type of graph

Received: 22 June 2023; Revised: 12 September 2023; Accepted: 29 September 2023

Communicated by Ljubica Velimirović

Email addresses: nada.mladenovic@fon.bg.ac.rs (Nada Mladenović), milica.stojanovic@fon.bg.ac.rs (Milica Stojanović)

a so-called 1-toroid and is also discussed as a polyhedron without diagonals [2], [17], [18]. This polyhedron has 7 vertices and it was shown in [21] that it is 3-triangulable with 7 tetrahedra.

Namely, the toroid (here called 1-toroid) defined by Szilassi [19], [20] is a polyhedron topologically equivalent to a torus. Similarly, the term p-toroid ($p \in \mathbb{N}$) is introduced for a polyhedron topologically equivalent to a p-torus-solid (a ball with p handles). In topology this object is known as a 3-handlebody of genus p. Here we use the common name 'toroid' for all p-toroids, independently of p. Minimal 3-triangulation and other properties of toroids are considered in [11], [12], [13], [14]. A connection graph is introduced to investigate whether some toroid is 3-triangulable.

In Section 2, the necessary terms from graph theory are given, as well as some cases, definitions and properties of 3-triangulation of polyhedra, especially p-toroids. Then, in Section 3 we give the construction of a polyhedron P based on a given connected graph and discuss the properties of P in Theorems 3.1 - 3.4. Algorithms are given in Section 4, for computing the numbers of vertices, edges, faces and handles of P, determined in Theorem 3.4.

2. Basic concepts and results

2.1. Preliminaries about graphs

A *graph* is a couple $G = (V, E)$ where V is a set of *nodes* (or vertices, rather used in computer science) and E is a set of *edges* connecting nodes from V. *Degree* of a node u is the number of edges with u as one of its endpoints.

Here, we shall only use *undirected* graphs, that is, those with edges that have no direction, and shall call them 'graphs' for short. A *Path*, a *cycle*, a *loop* are defined as usual in graph theory, while a *chain* is a sequence of edges and nodes of order two, with end nodes of order different from two. A cycle with all nodes of order two can also be considered as a chain.

A graph is *connected* if for each two distinct nodes u and v there exists at least one path joining them. Otherwise, graph is *disconnected* with two or more connected components.

A *tree* is a graph in which any two nodes are connected by exactly one path, which means that it is connected and has no cycles. A *spanning tree* is a subset of connected graph G, which has all the nodes covered with minimum possible number of edges. Each connected graph has a spanning tree.

For a spanning tree, we have to look for all edges which are present in the graph but not in the tree. Adding one of the missing edges to the tree will form a cycle which is called *fundamental cycle* ([6]). All fundamental cycles form a *cycle basis*. Note that a graph can have more different spanning trees. Consequently, each spanning tree constructs its own cycle basis. We can also form a cycle basis for any disconnected graph, using one spanning tree for each connected component.

However, the number of fundamental cycles is always the same and can be easily calculated: For any given graph having V nodes, E edges and r connected components, the number of fundamental cycles N_{FC}, called *cycle rank* or *circuit rank* is:

$$N_{FC} = E - V + r.$$

A *subdivision* of a graph G is a graph obtained from G by replacing some edges of G with internally node-disjoint paths ([9]). In other words, we can make a subdivision of G by dividing the edges of G and inserting new nodes between the edge-parts.

Before using the graphs in this paper, we have to make their geometric representation, and we shall consider it realized in 3-space.

2.2. Graph representation

Graphs can be used in different ways to examine properties of polyhedra and 3D modeling [1], [15], [16], [22]. We use the *abstract data type* (ADT) to work with graphs. That is a mathematical model of a data structure that specifies a type of data stored, the operations supported on them, and the types of parameters of the operations. The ADT specifies what each operation does, but it does not describe the way it is done.

As an abstract data type, a graph is a positional container whose positions are its vertices and its edges. Hence, the graph ADT stores elements at either its edges or vertices (or both). A position in the graph is always defined relatively, that is, in terms of its neighbors.

To make the ways of storing elements abstract and unified, in various implementations of the graph, we introduce a concept of 'a *position*' in the graph, which makes the intuitive notion of the 'place' element formal, relative to others in the graph.

A position itself is an abstract data type that supports a simple *element()* method which returns the element that is stored at this position. We also use specialized iterators for vertices and edges. An iterator is an enumeration with traversal order which can be guaranteed in some way.

In order to perform graph algorithms in a computer, we have to decide how to store a graph. There are several ways to realize the graph ADT with a concrete data structure. In this section, we discuss two popular approaches, usually referred to as the *adjacency list* structure and the *adjacency matrix* [3], [5].

There is a fundamental difference between the adjacency list and the adjacency matrix. The adjacency list structure only stores the edges actually present in the graph, while the adjacency matrix stores a placeholder for every pair of vertices (whether there is an edge between them or not). This difference implies that, for a graph G with n vertices and m edges, the edge list or adjacency list representation uses $O(n + m)$ space, whereas the adjacency matrix representation uses $O(n^2)$ space.

In modern object-oriented program languages, (such as C++, C# and Java) the ADT can be expressed by an interface, which is simply a list of method declarations. The ADT is realized by a concrete data structure, which is modeled in object-oriented program languages by a class. A class defines the data stored and the operations supported by the objects which are instances of the class. Also, unlike interfaces, classes specify how the operations are performed. A program language class is said to implement an interface if its methods give life to all of those of the interface.

2.3. *Some important cases of 3-triangulation*

The smallest number of tetrahedra in a 3-triangulation of a polyhedron with n vertices is $n - 3$. For example, such a polyhedron is a pyramid V_{n-1} with $n - 1$ vertices at the basis and the apex, which means a total of n vertices. We can 3-triangulate it as follows: do any 2-triangulation of the basis into $(n-1)-2 = n-3$ triangles. The apex together with each of such triangles forms one of the tetrahedra in 3-triangulation (Figure 1). Also, the triangular prism Π with bases $A_1B_1C_1$ and $A_2B_2C_2$ has 6 vertices and is 3-triangulable with 3 tetrahedra. Actually, the triangular prism Π can be considered as a 'pyramid' with the apex A_2 and the spatial pentagon $A_1B_1B_2C_2C_1$ as the basis (Figure 2).

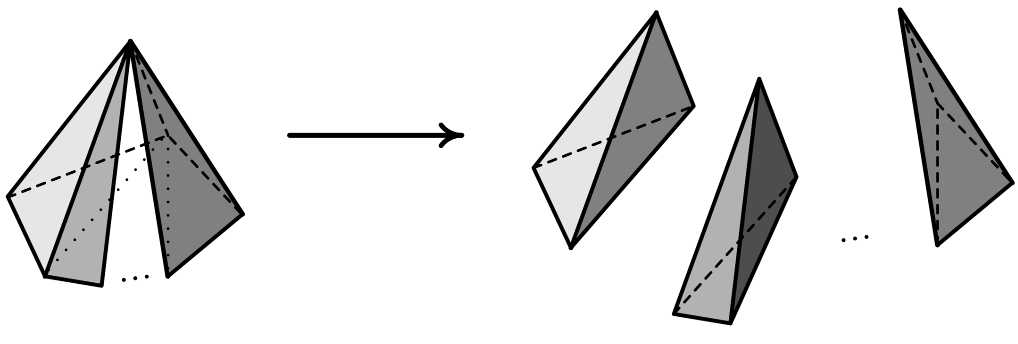

Figure 1: Triangulation of the pyramid V_{n-1} with n vertices

Here we consider 3-triangulation of p-toroids, a special class of polyhedra. Namely, the term 'polyhedron' usually means a simple polyhedron solid, topologically equivalent to a ball. On the other hand, there are classes of polyhedra topologically equivalent to a p-torus-solid (ball with p handles), $p \in \mathbb{N}$.

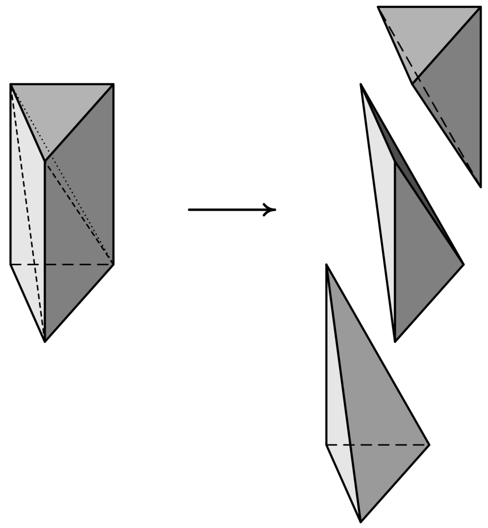

Figure 2: Triangulation of the triangular prism Π

An orientable surface of genus p (we call it a p-torus) is described as the boundary of the solid (p-handlebody, p-torus-solid) obtained by attaching p handles to a 3-ball. It can be effectively constructed from a ($2p$)-polygon by pairing (and gluing together) corresponding sides. Any side s and its pair S are oppositely directed related to the fixed orientation of the polygon, and then glued together. By standard combinatorial procedure, the polygon can be divided and glued into the cyclic normal form $a_1b_1A_1B_1a_2b_2A_2B_2...a_pb_pA_pB_p$, as *p-torus*. This combinatorial procedure is independent of the future spatial placement of the surface. Thus, we can form a p-torus from any spatial knot, as a topological circle in space. Afterwards, its surface can be 2-triangulated, to be the surface of a polyhedron.

The term p-toroid was introduced in [13], [14] based on Szilassi's [19] definition of 'toroid'.

Definition 2.1. *A polyhedron as a solid is called* p-toroid, $p \in \mathbb{N}$, *if it is topologically equivalent to a solid p-torus (ball with p handles), i.e. as a solid, it can be converted to a solid p-torus by continuous deformation.*

We use term toroid *as a common name for all p-toroids.*

2.4. Decomposition of polyhedra to convex pieces

To investigate possible 3-triangulations of non-convex polyhedra, we consider their decomposition into convex pieces and then form a graph representing such decomposition. For this purpose we need the following definitions.

Definition 2.2. *A polyhedron is* piecewise convex *if it can be divided into finitely many convex polyhedra* P_i, $i = 1, \dots, m$, *with disjoint interiors. A pair of above polyhedra* P_i, P_j *is said to be* neighbouring *if they have a common face called* contact face.

If the above polyhedra P_i and P_j are not neighbouring, they may have a common edge e or a common vertex v. That is possible iff there is a sequence of neighbouring polyhedra $P_i, P_{i+1}, \dots, P_{i+k} \equiv P_j$ such that the edge e, or the vertex v belongs to each contact face f_l common to P_l and P_{l+1}, $l \in \{i, \dots, i+k-1\}$. Otherwise, polyhedra P_i and P_j do not have common points.

Remark 2.3. *As convex pieces are 3-triangulable, the same property applies to piecewise convex polyhedra. Namely, we can first make a common 2-triangulation of the contact faces, and then, taking into account the new triangular faces, 3-triangulate convex pieces.*

We can also notice that every 3-triangulable polyhedron is a collection of connected tetrahedra which means that it is piecewise convex.

Definition 2.4. *If a polyhedron P is piecewise convex, its* connection graph *(or its* graph of connection*) is a graph whose* nodes *represent convex polyhedra P_i, $i = 1, \dots, m$, the pieces of P, and* edges *represent contact faces between them.*

In order to have the same number of handles for the considered toroid P and number of basic cycles of the corresponding connection graph G we introduce term *optimized graph* of connection. If G has some of the cycles which do not correspond to some handle of P such a cycle is called *false*. An example of a toroid P with connection graph that has false cycle is given in [13].

Let us consider a toroid P and its connection graph G that have one or more false cycles. For each of the false cycles, note all the nodes that belong to it and the corresponding convex pieces of P. The union of such convex pieces for each false cycle builds a new node of optimized graph $\hat{G}$. The other nodes of the graph G remain in $\hat{G}$ and we call them the old ones. The edges between the old nodes remain in $\hat{G}$. The edges of G between some old node and some node belonging to a false cycle are converted to the edge of $\hat{G}$ between that old node and the new one.

The optimized graph $\hat{G}$ has the same number of basic cycles as the number of handles of the starting toroid P. Note that it is not necessary that the new nodes of the optimized graph to correspond to convex polyhedra, they only correspond to simple piecewise convex polyhedra.

In [11], [12] theorems for 1-toroids and 2-toroids about the minimal number of tetrahedra necessary for their 3-triangulation were proved. The following is the corresponding theorem for p-toroids given in [13].

Theorem 2.5. *If a p-toroid with n vertices can be 3-triangulated, then the minimal number of tetrahedra necessary for its 3-triangulation is $T_{min} \geq n + 3(p-1)$.*

3. Construction of a polyhedron from a given connection graph

In [14], the polyhedron P is constructed based on the subdivided graph G' of the given graph G which is the skeleton of some polyhedron. Here, for the more general case of a given arbitrary connected graph G, the construction of the polyhedron P is given and its properties are examined. In this general case, nodes of order one and loops can appear, and the generalized graph can be a tree, which was not the case with the previously considered graphs. Also, there are no nodes of order two in the graphs - skeletons of polyhedra, which simplifies obtaining a subdivision of such graphs. In the simpler case, in such a process, we need to add exactly one node to each edge.

Here, for a given arbitrary connected graph G, we shall form a subdivided graph G', taking into account the requirements for the number of nodes added to each edge. The number of nodes added to each edge of G with both endpoints of order different than two must be at least one. Otherwise, we may omit adding new nodes to an edge. Additionally, the requirement is that the number of added nodes be sufficient to allow the new edges of G' to be straight lines. Since the number of added nodes can vary, we prefer to consider the graph G' as the original.

In he process of constructing the polyhedron P, we first mark nodes with degree $k \geq 3$ in gray, those with degree 2 in black, and those with degree 1 (leaves) in white. Each of the gray nodes in P represents subpolyhedra of type V_k, that is, a pyramid with k vertices at the basis (Figure 1). Each of the black nodes of G' represents subpolyhedra of type Π, i.e. prism (Figure 2) and each of the white nodes represents tetrahedron T. If necessary, Π will be slightly deformed to enable contact. So, after deformation, Π will have skew - non-parallel bases. Neighbours of V_k or T are always polyhedra of type Π. For Π, we use its two bases as contact faces with neighbours. For V_k with vertices $A_1, A_2, \dots, A_k$ in the basis and V as the apex, its k contact faces are the lateral faces $A_iA_{i+1}V$, $i \in \{1, \dots, k-1\}$ and A_kA_1V, while for T one of its faces is its only contact face.

The graph G' is visually similar to the polyhedron P, because the prisms Π looks like strings. Sequences of such strings give the impression of a sequence of (straight) edges from G', although in each of the non-cyclic chains there are one more edge than there are black nodes. Pyramids V_k and tetrahedra T represent other nodes of G'. In this construction, the prisms Π allow the pyramids V_k and the tetrahedra T to be far enough apart to form a handle for each of the basic cycles of G'. This means that G' is an optimized graph for P. Moreover, if G' has p ($p \in \mathbb{N}$) basic cycles, it is an optimized connection graph for the p-toroid P, and if G' is a tree, P is a simple polyhedron (we can also think of it as an 0-toroid). Since the nodes of the optimized graph do not have to represent convex polyhedra, but only piecewise convex ones, it follows that the grouping of prisms originating from the added black nodes with pyramids and tetrahedra gives the graph G as a optimized graph for P. The only exception to this grouping is in the case of subdividing loops of G.

Note that V_k has $k + 1$ vertices and is 3-triangulable with $k - 2$ tetrahedra, Π has 6 vertices and has 3 tetrahedra in a 3-triangulation. This means that the simple polyhedral pieces of P are 3-triangulable with the minimal number of tetrahedra guaranteed by Theorem 2.5 (which is also applicable in the case of $p = 0$), so this must also be true for the whole polyhedron P. Thus, the next statement holds.

Theorem 3.1. *For any connected, undirected graph G there exist a polyhedron P with subdivided graph G' of G, as an optimized connection graph. If G has no loops, it is also an optimized graph for P. Moreover, for the polyhedron P obtained by this construction, the minimal number T_{min} of tetrahedra in the subdivision is equal to the minimal possible number, guaranteed by the Theorem 2.5.*

For the special type of graph G, which is the skeleton of some simple polyhedron or h-toroid π, the corresponding polyhedron P obtained by the described construction is also itself a toroid with p handles, determined by the following theorem.

Theorem 3.2. *If the graph G is the skeleton of some h-toroid π ($h \in \mathbb{N} \cup \{0\}$) with F faces, then the p-toroid P constructed as before have p handles, where*

$$p = F + 2h - 1.$$

Proof. As we mentioned before, the number of handles p of P is equal to the cycle rank of G. Since the graph G is connected, it holds

$$p = E - V + 1,$$

where V and E are the nodes and edges numbers of G. When forming a subdivided graph G' of G, the number of nodes will increase by the same amount as the number of edges, so the number p will remain unchanged.

On the other hand, as G represents the skeleton of some h-toroid π, the numbers of vertices and edges of π are also V and E. Therefore, according to the Euler-Poincaré Theorem (or according to Euler's if $h = 0$) it follows

$$V - E + F = 2 - 2h.$$

Consequently

$$p = F - 2 + 2h + 1 = F + 2h - 1,$$

which had to be proved. □

For the same type of graph G, besides the number of the handles p of the corresponding toroid P, we can also calculate the number n of vertices, e of edges and f of faces. These numbers will be expressed by the numbers V and E of the nodes and edges of G.

Theorem 3.3. *If the graph G with V nodes and E edges is the skeleton of some h-toroid π ($h \in \mathbb{N} \cup \{0\}$), then the p-toroid P constructed as before has n vertices, e edges and f faces, where*

$$n = 2E + V, \quad e = 7E, \quad f = 3E + V.$$

Proof. We assume that a subdivided graph G' of G is obtained by dividing each edge of G into two parts and adding only one new black node between obtained parts. Then, the graph G' has $2E$ edges, V gray and E black nodes. Of course, gray nodes lead to pyramids V_{k_i}, $i \in \{1, \ldots, V\}$ in P, while black nodes lead to prisms Π.

When counting the vertices of P, only the vertices of V_{k_i} (with k_i vertices at the basis and the apex) should be taken into account, because each vertex of Π belongs to some contact face, and accordingly to some of the pyramids. So, the number of vertices of P is

$$n = \sum_{i=1}^{V}(k_i + 1) = \sum_{i=1}^{V} k_i + V.$$

The sum of the degrees for the gray nodes of G is the same as the sum of the the degrees for vertices of π. It is twice the number of edged E, since every edge of π is incident to two vertices of π. That means

$$\sum_{i=1}^{V} k_i = 2E$$

and it holds that

$$n = 2E + V.$$

When considering the number e, we take into account that each V_{k_i} has $2k_i$ edges (k_i at the basis and also k_i connecting the vertices of the basis to the apex). On the other hand, each Π has 9 edges, but 6 of them belong to contact faces and thus to some of the pyramids. Then we can conclude

$$e = \sum_{i=1}^{V} 2k_i + 3E = 4E + 3E = 7E.$$

Note that only the basis of the pyramids V_{k_i} and the lateral faces of the prisms Π are external, not-contact faces. It means

$$f = V + 3E.$$

Of course, applying the Euler-Poincaré theorem to calculate the number of faces will give the same result

$$f = e - n + 2 - 2p = 7E - (2E + V) + 2 - 2(E - V + 1) = 3E + V.$$

Thus we prove the Theorem. □

In the general case, the number of added vertices in the subdivided graph G' of G, may vary and it would not be sufficient to know the numbers V and E of the graph G. In the following Theorem we determine the numbers n, e, f, p and T_{min} based on the numbers of nodes of G' of the same order. More precisely:

Theorem 3.4. *Let G be a connected graph, and let G' and P be the subdivided graph and corresponding polyhedron of G, constructed as before. If ν_k are the numbers of nodes of G' of order k, then for P the numbers of are equal:*

1. *The number of vertices n is* $n = \sum_{k\geq 3}\left(1 - \frac{k}{2}\right)\nu_k + 3\nu_2 + \frac{5}{2}\nu_1$;
2. *The number of edges e is* $e = \frac{1}{2}\sum_{k\geq 3} k\nu_k + 6\nu_2 + \frac{9}{2}\nu_1$;
3. *The number of faces f is* $f = \sum_{k\geq 3}\nu_k + 3\left(\nu_2 + \nu_1\right)$;
4. *The number of handles p is* $p = \frac{1}{2}\sum_{k\geq 3}\left(k-2\right)\nu_k - \frac{1}{2}\nu_1 + 1$;
5. *The number of tetrahedra in the minimal triangulation is* $T_{min} = \sum_{k\geq 3}\left(k-2\right)\nu_k + 3\nu_2 + \nu_1$.

Proof. If the graph G' is a cyclic graph with all black nodes, the statement of the Theorem is obvious. Namely, there are ν_2 nodes in G', and therefore ν_2 pieces of Π with ν_2 contact faces in P. This means that in P there are $n = 3\nu_2$ vertices, $e = 6\nu_2$ edges and $f = 3\nu_2$ faces. It is necessary $3\nu_2$ terahedra to triangulate P and there is one handle in P.

Otherwise, the graph G' has only the chains of the first type, that is, those with gray and white end nodes. Moreover, each chain has l_i ($l_i \geq 1$) black nodes and $l_i + 1$ edges. Note that each black node belongs to one of the chains. Since each chain has two end nodes, the number L of chains can be calculated as the half sum of orders for all gray and white nodes. Using the notations ν_k for the node numbers of order k, L can be expressed as

$$L = \frac{1}{2}\left(\sum_{k\neq 2} k\nu_k\right) = \frac{1}{2}\left(\sum_{k\geq 3} k\nu_k + \nu_1\right).$$

1. When calculating the number of vertices n of the polyhedron P, we shall consider gray, black and white nodes separately. Since the gray nodes correspond to pyramids V_{k_i} with $k_i + 1$ nodes, the number of vertices of P corresponding to such nodes can be calculated as $\sum_{k\geq 3}(k+1)\nu_k$. The number of vertices corresponding to white nodes is $4\nu_1$ since these nodes lead to tetrahedra. When calculating the number of vertices corresponding to chains and thus black nodes, it should be taken into account that all the vertices of Π belong to contact faces. Therefore, in this calculation, we do not consider the vertices belonging to the bases of Π neighbouring pyramids or tetrahedra, and for other bases we take half of the sum of the number of their vertices. This means that for each chain with l_i black nodes we have $3(l_i - 1)$ vertices. For the sum of all such vertices, we get $3(\nu_2 - L)$. So, the whole number n is

$$\begin{aligned} n &= \sum_{k\geq 3}(k+1)\nu_k + 3(\nu_2 - L) + 4\nu_1 = \\ &= \sum_{k\geq 3}(k+1)\nu_k + 3\nu_2 - \frac{3}{2}\left(\sum_{k\geq 3} k\nu_k + \nu_1\right) + 4\nu_1 = \\ &= \sum_{k\geq 3}\left(1 - \frac{k}{2}\right)\nu_k + 3\nu_2 + \frac{5}{2}\nu_1. \end{aligned}$$

2. The number of edges in V_{k_i} is $2k_i$ (k_i edges are at the basis and k_i are 'lateral') and the number of edges in the tetrahedron is 6. Each prism Π has 3 'lateral' edges and 3 edges in each base. But the bases serve as contact faces, so in P there are $3\nu_2 + 3(\nu_2 - L)$ edges belonging to all prisms. The total number of edges of P is

$$\begin{aligned} e &= 2\sum_{k\geq 3} k\nu_k + 3\nu_2 + 3(\nu_2 - L) + 6\nu_1 = \\ &= 2\sum_{k\geq 3} k\nu_k + 6\nu_2 - \frac{3}{2}\left(\sum_{k\geq 3} k\nu_k + \nu_1\right) + 6\nu_1 = \\ &= \frac{1}{2}\sum_{k\geq 3} k\nu_k + 6\nu_2 + \frac{9}{2}\nu_1. \end{aligned}$$

3. Only the bases of pyramids and the lateral faces of prisms and tetrahedra are external, not-contact faces of P. This means that the number of faces

$$f = \sum_{k\geq 3} \nu_k + 3\left(\nu_2 + \nu_1\right).$$

4. Since the numbers of the basic cycles of G' and of the handles of P are the same, we can calculate p in two ways. For the number of basic cycles $p = E - V + 1$ holds, where E and V are the numbers of edges and nodes of G'. We can get the number of chains L as $E - \nu_2$, therefore

$$\begin{aligned} p &= L - \left(\sum_{k\geq 3} \nu_k + \nu_1\right) + 1 = \\ &= \frac{1}{2}\left(\sum_{k\geq 3} k\nu_k + \nu_1\right) - \left(\sum_{k\geq 3} \nu_k + \nu_1\right) + 1 \\ &= \frac{1}{2}\sum_{k\geq 3} (k-2)\,\nu_k - \frac{1}{2}\nu_1 + 1. \end{aligned}$$

On the other hand, the Euler-Poincaré Theorem provides us with the possibility to calculate the number of holes, because $f - e + n = 2 - 2p$. Note that the special case of this Theorem is the Euler's, when $p = 0$. Thus,

$$\begin{aligned} p &= \frac{1}{2}(-f + e - n) + 1 \\ &= \frac{1}{2}\left(-\left(\sum_{k\geq 3} \nu_k + 3(\nu_2 + \nu_1)\right) + \left(\frac{1}{2}\sum_{k\geq 3} k\nu_k + 6\nu_2 + \frac{9}{2}\nu_1\right) - \right. \\ &\quad \left. - \left(\sum_{k\geq 3}\left(1 - \frac{k}{2}\right)\nu_k + 3\nu_2 + \frac{5}{2}\nu_1\right)\right) + 1 \\ &= \frac{1}{2}\sum_{k\geq 3} (k-2)\,\nu_k - \frac{1}{2}\nu_1 + 1. \end{aligned}$$

5. The number of tetrahedra in the triangulation of P is the sum of tetrahedra in the triangulations of its pieces, i.e. pyramids and prisms Π, taking into account also the separate tetrahedra corresponding to the white nodes of G'. As we mentioned earlier, these numbers are $k_i - 2$, 3 and 1, therefore the statement is true.

Note that based on the Theorem 2.5 the minimal number of tetrahedra is the same, as we obtained here:

$$\begin{aligned} T_{min} &= n + 3(p-1) = \\ &= \sum_{k\geq 3}\left(1 - \frac{k}{2}\right)\nu_k + 3\nu_2 + \frac{5}{2}\nu_1 + \frac{3}{2}\left(\sum_{k\geq 3}(k-2)\,\nu_k - \nu_1\right) = \\ &= \sum_{k\geq 3}(k-2)\,\nu_k + 3\nu_2 + \nu_1. \end{aligned}$$

□

4. The algorithms for determining the properties of the polyhedron P

As in [15], [16] it is possible to create algorithms for calculating the numerical properties of the constructed polyhedron P. So, below are the algorithms for determining the numbers ν_k, n, e, f, p and T_{min}.

First, in Algorithm 1 we need to count the number of vertices of the same degree. As input, we use the adjacency matrix of a given undirected connected graph G with n vertices due to the simplicity of working with such kind of stored data. Then, when passing through the row of a certain vertex, we find its degree. Finally, we count the number $Ni[i]$ of vertices of the same degree i ($1 \leq i \leq n$), where $i \geq 1$ because the graph is connected, and i can be equal to n if a vertex is connected by edges to all other vertices and also to itself by a loop.

The following is the main Algorithm 2 for calculating n, e, f, p and T_{min} in which we use the results of the previous algorithm, that is, the numbers $Ni[1], Ni[2]..., Ni[n]$ of vertices of a certain order.

Algorithm 1 Finding order for each vertex of a graph G and counting the number of vertices of the same degree.

INPUT: Adjacency matrix A for undirected graph G with n vertices.
OUTPUT: The array $Ni[1], Ni[2], ..., Ni[n]$

```
Let Ni be an array of n elements.
Initially all elements of array be zero.
for (i = 0 to n-1) do
{  int d = 0;
    for (j = 0 to n-1) do
    {   if (A[i][j] != null) then
            d++;
    }
    Ni[d]++;
}
for (i = 1 to n) do
  write ("Ni[{0}] = {1} ", i, Ni[i]);
```

Algorithm 2 Calculating the numbers of vertices n, edges e, faces f and handles p and T_{min}

INPUT: The array $Ni[1], Ni[2], ..., Ni[n]$
OUTPUT: n, e, f, p and T_{min}

```
ni_1 = Ni[1];
ni_2 = Ni[2];
n = 0;
e = 0;
f = 0;
p = 0;
T_min = 0;

For (i = 3 to n)
{
        current = Ni[i]
        If (current > 0) then
        {
          n + = (1-i/2)*current;
          e += i*current;
          f += current;
          p += (i-2)*current;
          T_min += (i-2)*current;
        }
}
n += 3*ni_2 + 5*ni_1/2;
e = e/2 + 6*ni_2 + 9*ni_1/2;
f += 3*(ni_1 +ni_2);
p = p/2 - ni_1 +1;
T_min += 3*ni_2 + ni_1;
writeLine ("n = {0}; e = {1}; f = {2};", n, e, f, );
writeLine ("p = {0}; T_min = {1}. ", p, T_min);
```

Example 4.1. *Let us consider a graph G which is a single cycle with 5 nodes (Fig. 3). Since the edges of G can be straight lines, in this case there is no need to make subdivision of G, and the graph G' can be the same as G. Therefore, the polyhedron P is as it is shown on Fig. 3. Note that such P is a special case of the Szilassi's regular toroid [20].*

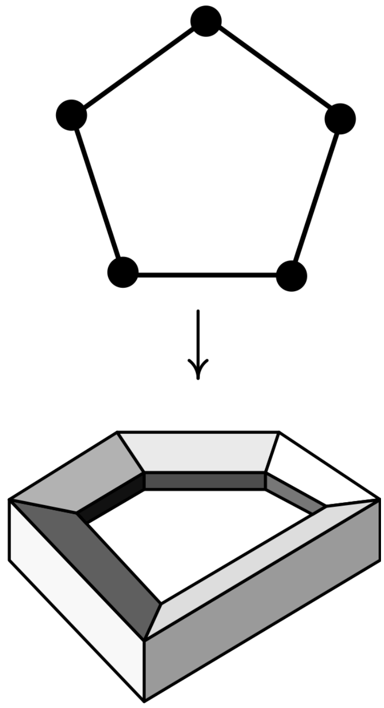

Figure 3: The graph G and polyhedron P

The input to the Algorithm 1 is: $0,1,0,0,1;\ 1,0,1,0,0;\ 0,1,0,1,0;\ 0,0,1,0,1;\ 1,0,0,1,0.$ *After applying our Algorithms 1 and 2 the results are the array* 0 5 0 0 0 *and* $n = 15$; $e = 30$; $f = 15$; $p = 1$; $T_{min} = 15$.

Example 4.2. *The graphs G and G' are given in Fig. 4 while the corresponding polyhedron P is given on Fig. 5. We note that the graph G with 5 nodes, consists of a loop, a three-node cycle, and a white node. Furthermore, two edges connect the cycle with the white node and the loop. In G' we added 0-3 black nodes to each edge, 8 in total. Thus, the graph G' have 13 nodes, one of order 1, nine of order 2 and three of order 3.*

Here, the input to the Algorithm 1 is:
0,1,0,0,0,0,0,0,0,0,0,0,0;1,0,1,0,0,0,0,0,0,0,0,0,0;0,1,0,1,0,0,0,0,0,0,0,0,0;0,0,1,0,1,0,1,0,0,0,0,0,0;
0,0,0,1,0,1,0,0,0,0,0,0,0;0,0,0,0,1,0,0,1,1,0,0,0,0;0,0,0,1,0,0,0,1,0,0,0,0,0;0,0,0,0,0,1,1,0,0,0,0,0,0;
0,0,0,0,0,1,0,0,0,1,0,0,0;0,0,0,0,0,0,0,0,1,0,1,0,1;0,0,0,0,0,0,0,0,0,1,0,1,0;0,0,0,0,0,0,0,0,0,0,1,0,1;
0,0,0,0,0,0,0,0,0,1,0,1,0.

The output of the Algorithm 1 is the array 1 9 3 0 0 0 0 0 0 0 0 0 0 *which we also use as the input to the Algorithm 2. The final result of Algorithm 2 is* $n = 28$; $e = 63$; $f = 33$; $p = 2$; $T_{min} = 31$.

5. Conclusions

In previous papers using the concepts of piecewise convex polyhedra and connection graphs, the properties of 3-triangulation of non-convex polyhedra, if any, were investigated. Here, for further investigation, based on the given graph as its connection graph, a p-toroid ($p \in \mathbb{N} \cup \{0\}$) is constructed. Algorithms for calculating the numbers of vertices, edges, faces and handles of P are then given. Furthermore, two examples of graphs and corresponding toroids are given, and algorithms are tested on them.

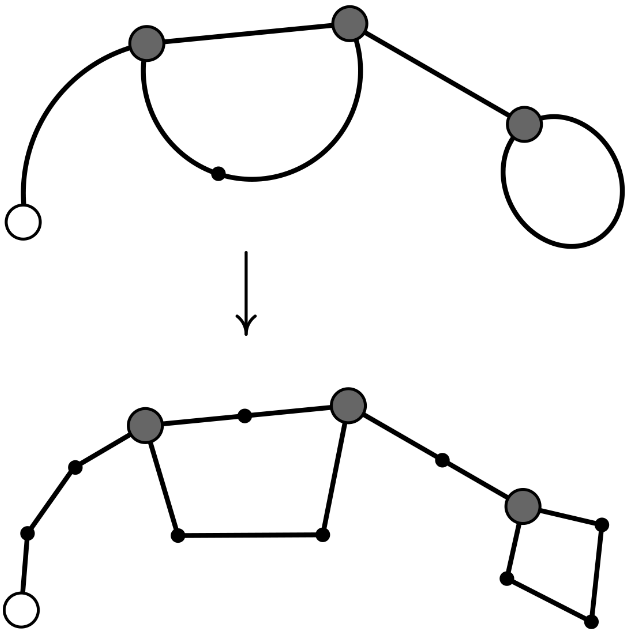

Figure 4: The graphs G and G'

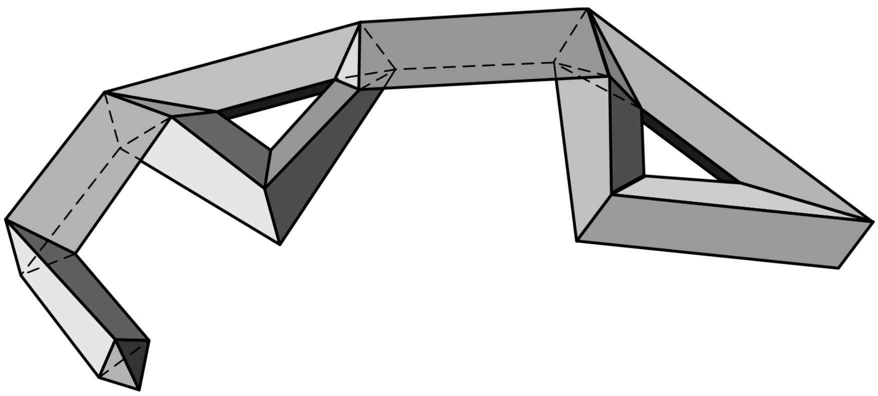

Figure 5: The polyhedron P

For p-toroids, the property of the minimal required number of tetrahedra T_{min} for 3-triangulation is important. That was the reason to examine also the number T_{min} for the constructed polyhedron. The result is that the lower limit obtained in previous papers is reached.

The algorithms described here can be implemented to create programs in, for example C++, C# or Java, so that further investigation of the properties of 3-triangulation of polyhedra would be possible by computer.

A connection graph can also be used to explore other 3-triangulation properties of non-convex polyhedra, such as an upper bound of the minimal required number of tetrahedra or the maximal number of tetrahedra. For these purposes, polyhedra could be constructed in a different way than described here.

References

[1] B. G. Baumgart, A polyhedron representation for computer vision, Proceedings of the 1975 National Computer Conference, AFIPS Conference Proceedings, AFIPS Press, Reston, Va. 44 (1975) 589–596.

[2] A. Császár, A polyhedron without diagonals, Acta Sci. Math. Universitatis Szegediensis 13 (1949) 140–142.

[3] J. McConnell, Analysis of Algorithms: An Active Learning Approach, (2nd Edition), Jones and Bartlett Publishers, 2007.

[4] H. Edelsbrunner, F. P. Preparata, D. B. West, Tetrahedrizing point sets in three dimensions, J. Symbolic Computation 10 (1990) 335–347.

[5] M. Goodrich , R. Tamassia, M. Goldwasser, Data Structures and Algorithms in Java^{TM}, (6th Edition), John Wiley & Sons, 2014.

[6] P. Sch, Enumerating All Cycles in an Undirected Graph, The Code Project Open Licence (2018) https://www.codeproject.com/Articles/1158232/Enumerating-All-Cycles-in-an-Undirected-Graph.

[7] E. Schönhardt, Über die Zerlegung von Dreieckspolyedern in Tetraeder, Math. Ann. 98 (1928) 309–312.

[8] D. D. Sleator, R. E. Tarjan, W. P. Thurston, Rotation distance, triangulations, and hyperbolic geometry, J. of the Am. Math. Soc. 1 (3) (1988) 647–681.

[9] Ch. Sobhan Babu, Ajit A. Diwan, Subdivisions of Graphs: A Generalization of Paths and Cycles, Discrete Mathematics, 308 (2008) 4479–4486.

[10] M. Stojanović, Algorithms for triangulating polyhedra with a small number of tetrahedra, Mat. Vesnik 57 (2005) 1–9.

[11] M. Stojanović, On 3-triangulation of toroids, Filomat 29 (10) (2015) 2393–2401 DOI:10.2298/FIL1510403H.

[12] M. Stojanović, 2-toroids and their 3-triangulation, Krag. J. Math. 41 (2) (2017) 203–217.

[13] M. Stojanović, Minimal 3-triangulations of p-toroids, Filomat 37 (1) (2023) 115–125 DOI:10.2298/FIL2301115S.

[14] M. Stojanović, Properties of 3-Triangulations for p-Toroid, Athens Journal of Sciences 10 (1) (2023) 31–40 DOI:10.30958/ajs.10-1-2.

[15] M. Stojanović, M. Vučković, Algorithms for investigating optimality of use cone triangulation for a given polyhedron, Krag. J. Math. 30 (2007) 327–342.

[16] M. Stojanović, M. Vučković, Convex polyhedra with triangular faces and cone triangulation, YUJOR 21 (1) (2011) 79–92 DOI:10.2298/YJOR1101079S.

[17] S. Szabó, Polyhedra without diagonals, Period. Math. Hung. 15 (1984) 41–49.

[18] S. Szabó, Polyhedra without diagonals II, Period. Math. Hung. 58 (2) (2009) 181–187 DOI:10.1007/s10998-009-10181-x.

[19] L. Szilassi, Regular toroids, Structural Topology 13 (1986) 69–80.

[20] L. Szilassi, On some regular toroids, Visual Mathematics (2005) http://www.mi.sanu.ac.rs/vismath/visbook/szilassi/.

[21] L. Szilassi, The Császár polyhedron subdivided into tetrahedra, Wolfram Demonstrations Project (2012) http://demonstrations.wolfram.com/TheCsaszarPolyhedronSubdividedIntoTetrahedra/.

[22] Q. Zeng, W. Ming, Z. Zhang, Z. Du, Z. Liu, C. Zhou, Construction of a 3D Stratum Model Based on a Solid Model, IEEE Access 9 (2021) 20760–20767 DOI: 10.1109/ACCESS.2021.3053628.

[23] H. Zhang, J. Zhang, W. Wu, X. Wang, H. Zhu, J. Lin, J. Chen, Angle-Based Contact Detection in Discontinuous Deformation Analysis, Rock Mechanics and Rock Engineering 53 (12) (2020) 5545–5569 DOI:10.1007/s00603-020-02206-z.

[24] Q. Zhang, S. Lin, X. Ding, A. Wu, Triangulation of simple arbitrarily shaped polyhedra by cutting off one vertex at a time, Int. J. Numer. Methods. Eng. 114 (5) (2018) 517–534 DOI:10.1002/nme.5752.