

Weaving patterns inspired by the pentagon snub subdivision scheme

Henriette Lipschütz^a, Ulrich Reitebuch^a, Martin Skrodzki^b, and Konrad Polthier^a

^aFachbereich Mathematik und Informatik, Institut für Mathematik, Freie Universität Berlin, Germany

^bComputer Graphics and Visualization, TU Delft, The Netherlands

ARTICLE HISTORY

Compiled April 4, 2022

ABSTRACT

Various computer simulations regarding, e.g., the weather or structural mechanics, solve complex problems on a two-dimensional domain. They mostly do so by splitting the input domain into a finite set of smaller and simpler elements on which the simulation can be run fast and efficiently. This process of splitting can be automated by using subdivision schemes.

Given the wide range of simulation problems to be tackled, an equally wide range of subdivision schemes is available. They create subdivisions that are (mainly) comprised of triangles, quadrilaterals, or hexagons. Furthermore, they ensure that (almost) all vertices have the same number of neighboring vertices.

This paper illustrates a subdivision scheme that splits the input domain into pentagons. Repeated application of the scheme gives rise to fractal-like structures. Furthermore, the resulting subdivided domain admits to certain weaving patterns. These patterns are subsequently generalized to several other subdivision schemes.

As a final contribution, we provide paper models illustrating the weaving patterns induced by the pentagonal subdivision scheme. Furthermore, we present a jigsaw puzzle illustrating both the subdivision process and the induced weaving pattern. These transform the visual and abstract mathematical algorithms into tactile objects that offer exploration possibilities aside from the visual.

KEYWORDS

mesh subdivision schemes; pentagons; fractals; weaving; paper models; jigsaw puzzles; visual and tactile illustration

1. Introduction

For many people all around the world, checking the weather report is part of their daily routine in the morning. Will it be a sunny day for the planned trip to the beach? Or will some rain finally water the crops on the field? While a weather forecast is easily accessible nowadays, its computation is fairly involved and occupies the largest supercomputers in the world.

The underlying mathematical equations that govern the weather cannot be solved explicitly, i.e., it is not possible to write down an analytic solution for them. A first breakthrough in weather forecast was therefore the development of difference methods that do not solve the entire system of equations, but instead compute an approximate

solution. These solutions are not computed for the entire area of, say, a country, but only for a finite number of points in the country, e.g., at the weather stations where good information is available. For any place in the region that is not sufficiently close to one of the weather stations, the forecast is computed as a mixture of the solutions from the surrounding weather stations. Of course, the precision of the computation will depend on the number of points distributed across the country; a single weather station cannot provide a prediction as accurate as a dense network of stations distributed throughout the country. This was also realized by Lewis Richardson, who first tried (and failed) to predict the weather in 1928 [11, Ch. 3.8.4].

Techniques for the solution of such complex problems as weather predictions came up in the 1950's from a completely different scientific branch: structural mechanics. The equations that govern stresses in mechanical assemblies are also impossible to be solved analytically. In particular complex geometries, like entire airplanes call for optimizations of, e.g., the weight, while having a high demand for structural integrity. In order to achieve both, *finite element methods* are used to compute approximations of the arising stresses. Not the entire hull of the aircraft is considered in the computation, but it is split into a (finite) set of simple elements, such as triangles, quadrilaterals, or other geometric primitives. Again, like in the case of the weather stations, the computations become more accurate with a growing number of elements [11, Ch. 5.6].

One possibility to generate a fine collection of surface elements for a geometry is the use of *subdivision schemes*. These start from a coarse input and iteratively divide each geometric primitive into smaller parts until a limit surface is reached. It is one specific representative from the family of these subdivision schemes that we want to illustrate in this article. In the following, we will first discuss existing subdivision schemes that create triangles, quadrilaterals, and hexagons. This discussion motivates the description of a method creating pentagons, which also produces a fractal boundary curve as well as a collection of fractal-like interior curves.

Naturally, subdivision schemes have been illustrated with static figures, dynamic videos or animations, and interactive computer programs. However, all these are confined to the page of a book or the screen of a computer, tablet, or smartphone, i.e., they relay visual information about the content. None of them provide a haptic sensation of what it means to have a two-dimensional domain and subdivide it into smaller pieces. Starting from the pentagon subdivision scheme, we introduce a coloring that provides a weaving pattern on the subdivided geometries. The pattern consists of a small number of strands which intertwine in pairs and which weave into a complex pattern. Here, the number of pairs depends on the number of refinement steps executed. After exploring properties of the coloring and the related weaving patterns for the pentagonal case, we discuss how they can also be applied to other subdivision schemes.

Finally, this article aims to close the gap between virtual or visual illustrations on the one hand and haptic illustrations on the other hand. To do so, we present two paper models that can be manufactured at home. These paper models transfer the observations made on the weaving patterns into a personal experience by assembling the corresponding subdivisions. Our final contribution is a two-sided jigsaw puzzle that incorporates both the pentagon subdivision scheme as well as two different views on the weaving patterns. Thus, the paper models and the jigsaw puzzle provide haptic means to explore both subdivision schemes and the related weaving patterns.

As will become clear throughout the article, these two physical realizations aim at illustrating different aspects of the pentagon subdivision scheme and its weaving patterns. Here, the paper models emphasize the weaving aspect, which provides the

opportunity of experiencing the assembly of the pattern from its basic parts. It can easily be extended to cover a larger area. Yet, the assembly is somewhat cumbersome and requires a high degree of dexterity. The jigsaw puzzle emphasizes the different steps of the subdivision process. While it remains at a fixed level of the subdivision, its colors indicate how to proceed from one level to the next. Hence, it incorporates the entire process. Furthermore, it is comparably approachable and easy to play with, compared to the paper models.

Throughout the research process for this paper and when creating the illustrations, we encountered several mathematical questions. While we offer insights to some of them, others remain open and have yet to be answered. As an invitation to the mathematical community, we pose these open questions prominently throughout the paper. We hope that this inspires new research, starting from the presented illustrations.

2. Subdivision schemes

As stated in the introduction, a subdivision scheme works on a set of geometric primitives, such as triangles, quadrilaterals, or hexagons, called a *mesh*. This input is iteratively split into a number of smaller primitives. Each of these iterative passes is called a *refinement step*. During this process, new vertices, edges, and faces are inserted following the set of rules of the respective subdivision scheme. In the following, we will denote the respective sets by V , E , and F . For a vertex $v \in V$, we will denote the number of neighbors of v as the *degree* of v . Similarly, for a face $f \in F$, we denote the number of its sides as the *degree* of f . In the following, we will always assume a planar input mesh.

In general, one distinguishes between *interpolating* and *approximating* subdivision schemes: While the first group keeps the vertex positions of the original input mesh intact, the second group does not. Furthermore, a subdivision scheme is called *primal* if a face f of the mesh is replaced by several new faces f_1, \dots, f_n in the refinement step. If the vertices of the mesh are replaced by several new vertices, the scheme is called *dual*. Observe that, depending on the inserted elements, the number of all three components—vertices, edges, and faces—increases.

Throughout the remainder of the paper, we will apply subdivision schemes solely to two-dimensional input geometries. Hence, we do not discriminate between a *remeshing* and a subdivision scheme. Here, the latter usually adds regularity properties of the curved limit surface embedded in three-dimensional space. This is out-of-scope for our considerations, which will be solely planar.

2.1. Triangle meshes

A first example of a subdivision scheme that operates on triangle meshes is the subdivision scheme by Loop. The general idea is to introduce a new vertex for each edge as a weighted combination of the two triangles that share the edge. This new vertex is connected to the two old vertices spanning the edge on which it was created as well as to the four other new vertices created as edge midpoints in its neighboring triangles. Thereby, each newly inserted vertex has degree six. Finally, in a smoothing operation, the old vertices are moved to a convex combination of their new neighbors, see Figure 1.

The *butterfly* subdivision scheme acts based on the same principles [10]. It differs in the selection of points that contribute to the position of the newly inserted vertices.

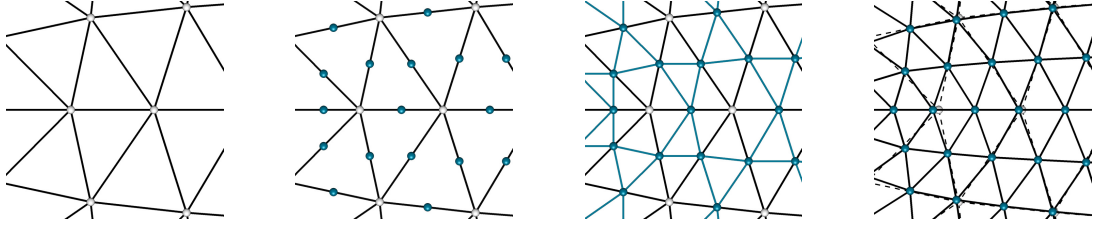


Figure 1.: An illustration of the subdivision scheme by Loop [16]: input mesh, inserting vertices as edge midpoints, connecting the new vertices, and smoothing the mesh to be more regular. White input vertices are moved during smoothing.

See Figure 2 for the result of applying one refinement step of the *butterfly* subdivision scheme to a non-regular triangle mesh.

In contrast to these two schemes, consider the $\sqrt{3}$ subdivision scheme [15]. It operates slightly differently by not creating vertices on the edge midpoints but the triangle barycenters. These are connected to the three triangle vertices and afterwards, the old edges are flipped, see Figure 3. Therefore, applying the scheme twice subdivides every input triangle into nine smaller triangles. Note that all three subdivision schemes listed so far, the *butterfly* and $\sqrt{3}$ subdivision schemes as well as the subdivision scheme by Loop are primal, as they perform face splits. While the $\sqrt{3}$ subdivision scheme and the subdivision scheme by Loop are approximating, the *butterfly* subdivision scheme is interpolating.

2.2. Quad and hexagon meshes

Other schemes have been proposed to work on (or create) different mesh types. The *mid-edge* subdivision scheme was proposed independently by Peters and Reif [21] as well as by Habib and Warren [12]. It connects every edge-midpoint to the four midpoints of the edges that share both a vertex and a face with the current edge, see Figure 4. Thereby, it creates meshes that are *quad-dominant*, i.e., that consist mostly of quadrilateral faces, at least after applying the scheme a minimum of two times. However, vertices and faces of the input mesh, which are of a degree different from four, give rise to faces of the same degree after the refinement step. For instance, subdivision of a triangle via the *mid-edge* subdivision scheme will at all steps contain a triangular face in the center of the refined mesh. This scheme is approximating and dual.

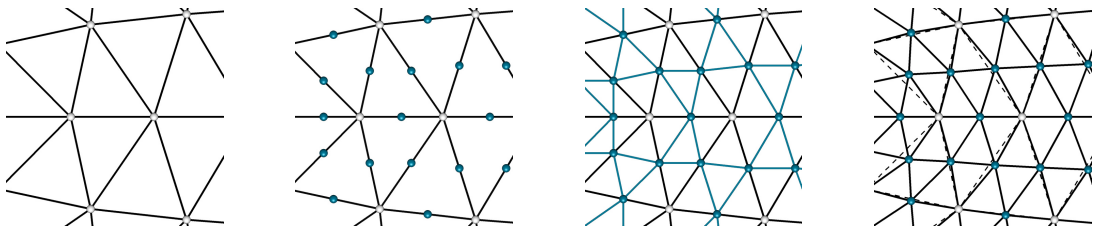


Figure 2.: An illustration of the *butterfly* subdivision scheme [10]: input mesh, inserting vertices as edge midpoints, connecting the new vertices, and smoothing the mesh to be more regular. White input vertices are retained during smoothing.

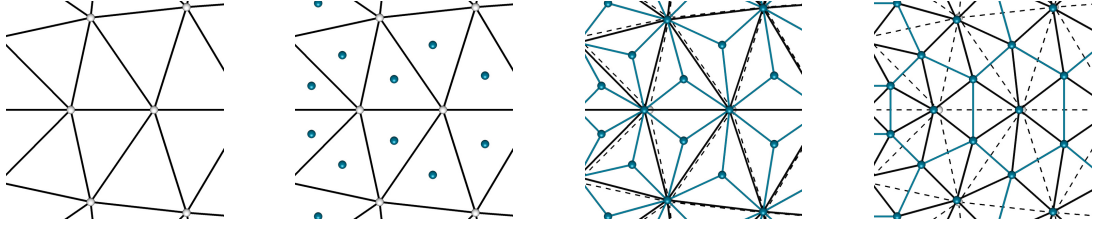


Figure 3.: An illustration of the $\sqrt{3}$ subdivision scheme [15]: input mesh, inserting vertices as face barycenters, connecting and smoothing the new vertices, and flipping the original mesh edges. White input vertices are moved during smoothing.

The triangle subdivision schemes discussed above do indirectly create a whole different class of meshes. Namely, their resulting meshes can easily be converted into hexagon-dominant ones. The procedure to do so simply consists of turning the mesh into its *dual*, i.e., placing a vertex at the center of each face of the mesh, and connect those vertices whose parental faces are connected via an edge. Thereby, the vertices of the original mesh become faces of the dual, which has as many edges as the degree of the respective vertex. As the *butterfly* and the $\sqrt{3}$ subdivision schemes and the subdivision scheme by Loop primarily create vertices with degree six—at least starting from the second refinement step—the dual mesh is dominated by hexagons, as described by Akleman and Srinivasan [1].

It is worth noting at this point that the very first subdivision schemes by Doo-Sabin [8, 9] and Catmull-Clark [6] are applicable to arbitrary meshes. The first creates quad-dominant subdivisions, while the second creates meshes that are solely comprised of quadrilaterals. For a general overview of subdivision schemes see [3, 25] and for a discussion of primal/dual quadrilateral subdivision schemes see [28].

2.3. Regularity and the absence of pentagonal faces

The well-known subdivision schemes presented above produce very regular meshes in which all newly inserted vertices have the same degree and all new faces are of a given type (triangles, quadrilaterals, or hexagons). This regularity does, however, not hold for vertices and faces of the input mesh that are not already of the required type. For instance, if a vertex in the input mesh has a degree different from the vertices inserted in the refinement step, it will continue to have this different degree through the whole

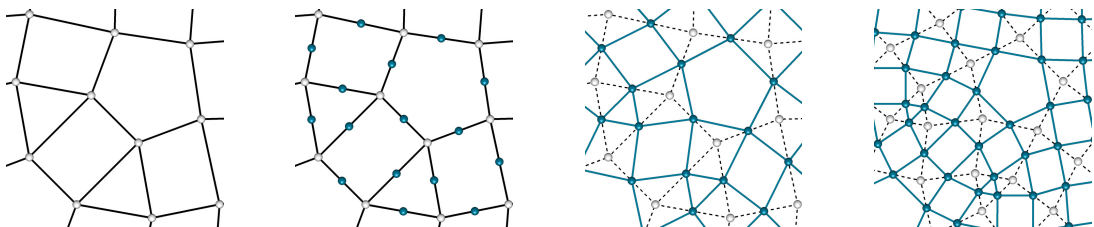


Figure 4.: An illustration of the *mid-edge* subdivision scheme [21]: input mesh, inserting vertices as edge midpoints, and connecting the new vertices. Fourth image shows a second complete refinement step, which illustrates that two steps of the *mid-edge* subdivision scheme are equivalent to one step in the subdivision scheme of Doo-Sabin [9], compare to Figure 5c.

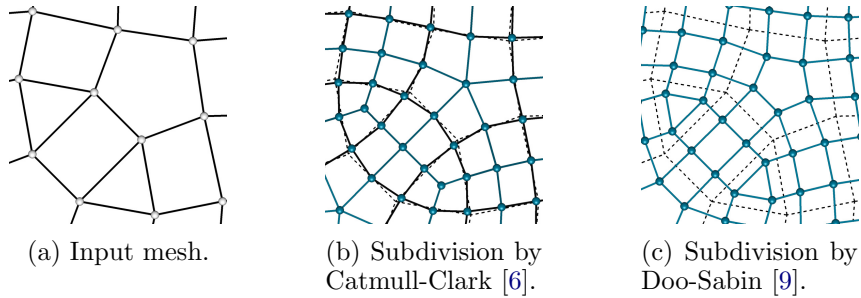


Figure 5.: Irregularities of the input mesh retained during the refinement steps. The subdivision scheme of Catmull-Clark [6] mainly creates vertices of degree four, but has vertices of different degree wherever the input mesh has a non-quadrilateral face. The subdivision scheme of Doo-Sabin [9] mainly creates quadrilaterals, but retains non-quadrilateral faces of the input mesh. Also, it is combinatorially equivalent to applying the *mid-edge* subdivision scheme twice, confer Figure 4.

subdivision procedure or spawn a face of corresponding degree. The same is true for faces of the input mesh that have a degree different from the newly inserted faces, they are either retained in the entire refinement step or spawn a vertex of corresponding degree. Hence, all irregularities present in the final subdivided mesh are already given with the input mesh.

This can be observed in Figure 5 when considering the subdivision schemes of Catmull-Clark [6] and Doo-Sabin [9] respectively. The first always creates quadrilateral faces and all vertices inserted after the first refinement step are of degree four. However, at those places where the input mesh does have a non-quadrilateral face, the scheme creates a vertex with degree different from four. The second scheme always creates vertices of degree four and all faces inserted after the first refinement step are quadrilaterals. However, non-quadrilateral faces of the input mesh are retained across all refinement steps.

Note that the subdivision schemes discussed in this section are capable of generating meshes that are dominated or even completely comprised by triangles, quadrilaterals, or hexagons. In this list, one shape is clearly missing and has not played a role so far: pentagons. Therefore, in the next section, we will consider a subdivision scheme that creates a pure pentagon mesh from a given input mesh.

3. The pentagon snub subdivision scheme

Given an arbitrary input, the subdivision scheme we investigate in the following produces a mesh consisting solely of pentagons. Vertices newly introduced in a refinement step have either degree three or five, while all vertices originally given in the input keep their degrees throughout all refinement steps applied. We originally approached this subdivision scheme from the context of cutting operations executed on Platonic or Archimedean solids. Thus, we call it *pentagon snub subdivision scheme* and will explain this naming later in this section. The version of the pentagon snub subdivision scheme we will consider here consists of four steps which are listed in the following and will be explained in more detail afterwards.

Let \mathcal{M}_0 be a 2-manifold mesh in the plane, consisting of convex faces only and having no self-intersections. For later use, we subdivide both the edge set E and the

vertex set V into two subsets—the outer and inner edges or vertices, respectively. An edge is called an *inner edge* if it is incident to exactly two faces, otherwise it is called an *outer edge*. A vertex is called an *inner vertex* if it is incident to inner edges only, it is called an *outer vertex* otherwise.

In each refinement step $t \in \mathbb{N}$ of our version of the subdivision scheme, the following four operations are applied to \mathcal{M}_t in order to obtain the refined mesh \mathcal{M}_{t+1} :

- (1) **Insertion of vertices and edges:** Each edge e in \mathcal{M}_t is replaced by three edges in the form of a stretched “Z” (henceforth called a *Z-triplet*). Hence, two vertices are added as well. The first and last vertex of the Z-triplet correspond to the two vertices of e .
- (2) **Insertion of face barycenters:** To each face f of \mathcal{M}_t , a new vertex is added at the barycenter of f .
- (3) **Insertion of edges:** Iterate over all edges e of \mathcal{M}_t . If edge e was an inner edge of \mathcal{M}_t , both new vertices (created during operation 1) are connected to the respectively closest barycenter (created during operation 2) of a face incident to e . If e is an outer edge, only one of the newly created vertices is connected to the barycenter of the face incident to e .
- (4) **Smoothing operation:** Simultaneously, the position of each inner vertex is revised using the barycenters of the faces incident to it.

A variant of this subdivision scheme including the first three operations was proposed by Bowers and Stephenson in 1997 [4] and discussed by Akleman et al. [2] while combinatorial aspects are investigated by Yan [27, Sec. 3.2]. Furthermore, a later subsection will relate the first three operations to a notation due to Conway [7], which motivates the naming of the subdivision scheme. In the following, we will discuss and illustrate these operations in more detail. Furthermore, we will justify the necessity of smoothing operation.

3.1. Insertion of vertices and edges

During operation 1, every edge is replaced by a Z-triplet in which all edges have equal lengths, see Figure 6a. The smaller angle enclosed by two edges of the Z-triplet is equal to $\frac{2\pi}{3}$. The newly inserted inner vertices have degree three after each refinement step. Therefore, this choice of the angle allows for an equal partition of 2π among all three angles. However, in general, the other two angles are not necessarily equal to $\frac{2\pi}{3}$. The first and last vertex of each Z-triplet (colored white in Figure 6a) are identical to the vertices of the edge it replaces. Hence, the edge length of the Z-triplet depends on the length of the replaced edge, but is completely determined by the angle choice. Within operation 2, a new vertex is added at the barycenter of every face of \mathcal{M}_t . See Figure 6b for an example of an input mesh \mathcal{M}_0 and Figure 6c for the result of applying operations 1 and 2 on the input once.

The Z-triplets are inserted always in the same way, i.e., running through the boundary of a face, the newly inserted faces appear to the left and to the right alternately. This decision is made in the very beginning for one edge. Propagating the decision along the mesh fixes the setup for the entire procedure.

During operation 3, we iterate over all edges e of the mesh \mathcal{M}_t . For each edge e , the line running through e divides the plane into two half-planes. If e is an inner edge, each half-plane contains exactly one of the faces incident to e . Furthermore—by construction—each half-plane also contains exactly one of the newly created vertices from operation 1. These new vertices are connected to the face midpoint (created

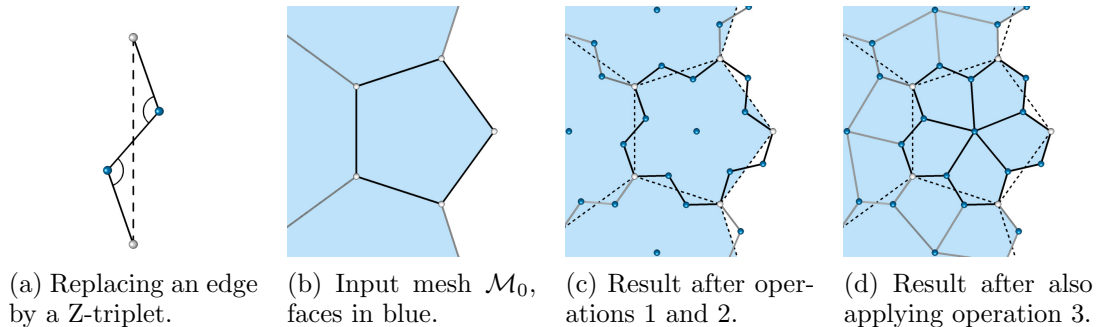


Figure 6.: Illustration of a Z-triplet with its two enclosed $\frac{2\pi}{3}$ angles, where the original line that is replaced by the Z-triplet is shown dashed. Also shown is the input mesh \mathcal{M}_0 with its faces colored blue and a boundary to the right, i.e., two inner and three outer vertices are shown. The last two images show operations 1–3 applied in each refinement step: replacing edges by Z-triplets, inserting face barycenters, and connecting new vertices.

during operation 2) lying in the same half-plane. If e is an outer edge, exactly one of the newly created vertices lies in the same half-plane as the face e is incident to. Then, this vertex and this face midpoint are connected. See Figure 6d for an illustration of the result of applying operation 3.

Note that the scheme creates only pentagonal faces. After operations 1–3 in the first refinement step, vertices from the Z-triplets have degree three, while a vertex that was created as barycenter of a face f has degree n_f , where n_f denotes the number of vertices of the face f . Like in the other subdivision schemes discussed in Section 2, these vertices, stemming from the faces of the original mesh, will always keep their degree and not become regular. All newly created outer vertices always have degree two or three while outer vertices of higher degree that are already present in the input mesh retain this degree throughout the subdivision process. From the second refinement step onward, all newly inserted vertices have degree three or five.

In case of applying the subdivision scheme to a single pentagon as input \mathcal{M}_0 , one gets the following recursion result for the number of vertices, edges, and faces of the mesh \mathcal{M}_{t+1} , after t refinement steps, in terms of the vertex set V_t , the edge set E_t , and the face set F_t of \mathcal{M}_t :

$$|V_{t+1}| = |V_t| + 2|E_t| + |F_t|, \quad |E_{t+1}| = 3|E_t| + 5|F_t|, \quad |F_{t+1}| = 5|F_t|, \quad (1)$$

while the following holds for an arbitrary input mesh:

$$|V_{t+1}| = |V_t| + 2|E_t| + |F_t|, \quad |E_{t+1}| = 3|E_t| + \sum_{f \in F_t} n_f, \quad |F_{t+1}| = \sum_{f \in F_t} n_f. \quad (2)$$

Thus, by iteratively applying operations 1–3, we have created a subdivision scheme that creates pentagon meshes from an arbitrary input. So far, the operations described here match the approach of Bowers and Stephenson [4], Yan [27], and Akleman et al. [2] combinatorially. However, embeddings of the resulting meshes, when following the steps as outlined above, generally lack an important geometric property, which we will address in the following.

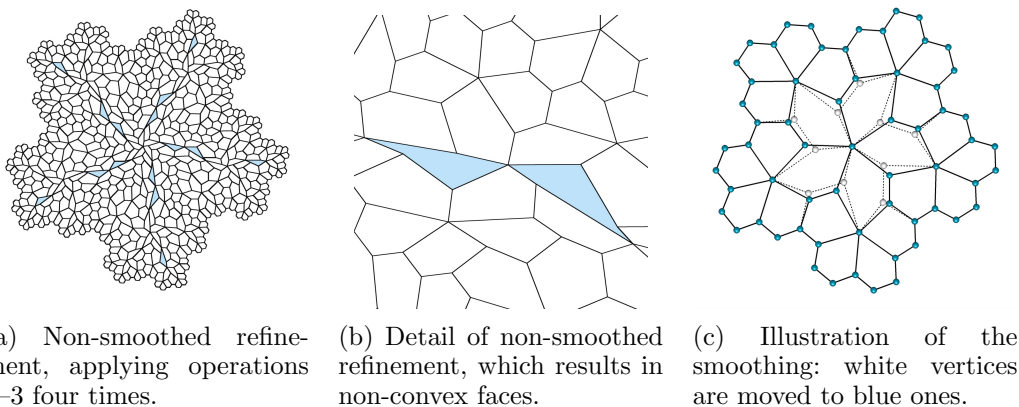


Figure 7.: Simply applying operations 1–3 as given above results in non-convex faces. A smoothing operation prevents this by nudging faces into a more convex position.

3.2. The smoothing operation

As it can be seen in the blue highlighted faces in Figure 7a, after several refinement steps, some of the newly created pentagons contain edges which are—in comparison to the other edges of the same face—relatively long. Also, some of the new faces are not convex any more (see Figure 7b, non-convex faces colored in blue). Because of the focus on combinatorial aspects, both the work of Bowers and Stephenson [4] and that of Yan [27] do not address this. Akleman et al. [2] do propose a second, different procedure that ensures convexity, which we will discuss after presenting our approach to preserving convexity of the faces.

To overcome the creation of non-convex faces and to avoid intersections of new edges, we introduce a smoothing operation that is applied to the vertices as operation 4 in each refinement step. However, this operation is not applied to all vertices. If a vertex is an outer vertex, its position is maintained, while the inner vertices are possibly moved to a new position.

The smoothing is performed as follows. For each face f of the mesh \mathcal{M}_t , as obtained after operation 3, its barycenter b_f is computed. Then, for each inner vertex v , its new position is determined as the average of the face barycenters b_f of those faces f to which v is incident. After determining the new positions, all vertices are updated at once. Since all inner vertices may be moved in the smoothing operation, the scheme is not interpolating. See Figure 7c for an illustration of the smoothing operation.

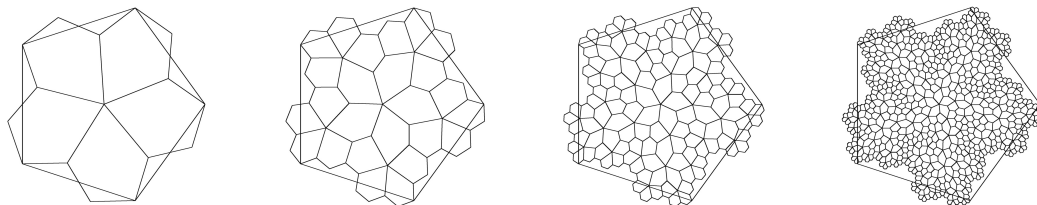


Figure 8.: From left to right: operations 1–4 of pentagon snub refinement with smoothing, applied to an initial pentagon (shown in the background). Note the creation of more regular faces, compared to Figure 7a.

Regular pentagons do not tessellate the plane—therefore, the created pentagons cannot all be regular. Figure 8 shows the first four refinement steps of the presented subdivision scheme, applied to a regular pentagon. Since the input pentagon does not induce a vertex with degree different from three or five, all inner vertices have degree three or five after every refinement step—some inner pentagons have one vertex with degree five and four with degree three, some have two vertices with degree five and three with degree three. In 2D, the presented refinement scheme has the following properties: it is primal (faces split into faces), approximates the vertices of the original mesh because of the smoothing operation, and interpolates the outer vertices of the input mesh.

While the proposed smoothing operation cannot guarantee convex faces and the avoidance of edge-intersections, we have not encountered any such behavior in our experiments with reasonably regular input meshes. As stated in Equation (2), our subdivision scheme is applicable to arbitrary input meshes, see several examples in Figure 9, all of which exhibit convex faces. Our work with various examples motivates the following question, which is posed as a problem for further investigation:

Open Question 1. *Does the smoothing operation always keep tiles convex?*

In the above, we have approached the creation of pentagonal faces from the perspective of subdivision schemes in two dimensions. As briefly stated above, in the context of searching for a “regular” pentagonal tiling of the plane, Bowers and Stephenson introduce a pentagonal subdivision scheme that splits a given pentagon into six smaller ones, see [4, Fig. 4]. The obtained tiling has a direct relation to circle packings and allows for several generalizations. Relaxing one of their parameters yields a result with the same combinatorial structure as the subdivision scheme presented above (compare Figure 6d to [4, Fig. 13(a)]). This combinatorial structure is investigated by Yan [27]. For both works [4, 27], it is unclear how to embed the obtained meshes, while our approach creates a corresponding embedding on the fly.

Our fourth operation was added in order to obtain as-convex-as-possible faces. Note that Akleman et al. present a different approach towards this end, which they denote as “Pentagon preserving algorithm“ [2, Sec. 5.2]. They propose to split all edges at their midpoints and move a copy of these midpoints towards the involved face centers. This results in the creation of scaled, smaller faces within each original face of the mesh. By construction, these are convex¹, however, they are only pentagonal if their corresponding original face was a pentagon. This is in contrast to the scheme presented here, which always creates pentagonal faces from arbitrary input meshes.

Instead of taking a two-dimensional tiling- or subdivision-based approach, faces can also be created from a three-dimensional perspective when considering operations performed on polyhedra. In their book, Conway, Burgiel, and Goodman-Strauss present a naming scheme for Archimedean and Catalan solids [7, Ch. 21]. The first three operations of the scheme discussed here correspond combinatorially to the process of forming the dual of the snub dodecahedron, written as dsD in their notation, and also know as *pentagonal hexacontahedron*, one of the Catalan solids. This relation to polyhedra motivates our naming choice for the subdivision scheme and explains the title of this section. George Hart collects an expansion of the proposed notation on a website [13]. Note that these symmetry-preserving operations can be cast into an

¹The smaller copies of each face are convex by congruency while the other parts cannot include a vertex angle larger or equal to π without violating the property that each point of the smaller part has to be an element of the convex original face.

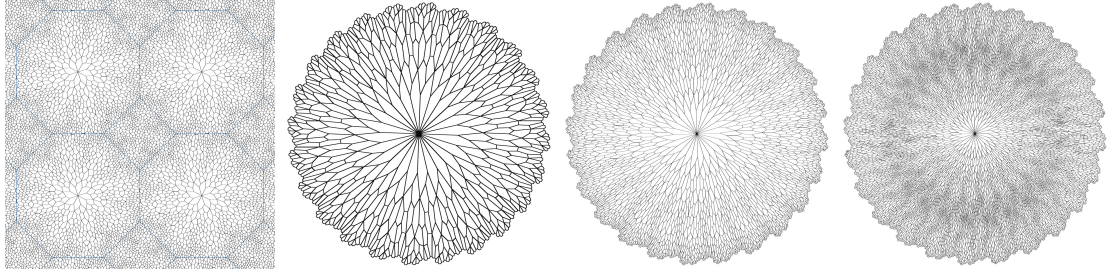


Figure 9.: From left to right: four times subdivided 8-4-mesh, three times subdivided regular 24-gon, four times subdivided regular 24-gon, and three times subdivided triangulated 24-gon.

even more general framework [5]. In the terminology of Hart, the subdivision scheme presented above corresponds to the gyro of the dodecahedron, written as gD .

Thus, the combinatorial aspects of operations 1–3 of the presented subdivision scheme have their roots in literature of both tilings and polyhedra. The additional element we added is of geometric nature, by adding the smoothing operation, aiming at embeddings whose faces are always convex. In the following, we will keep the two-dimensional perspective and further explore aspects of continued refinement steps.

4. Fractal-like

The last image shown in Figure 8 gives the impression that the presented subdivision scheme does not only create meshes composed of pentagons, but—for a single input pentagon—additionally results in an object resembling a fractal. A similar observation has been made by Bowers and Stephenson for their geometric embedding [4, Part III]. Recall that a *fractal* is a naturally or artificially created object characterized by several properties: self-similarity, having fractal dimension, and being produced by an iterative procedure [18, Ch. 3]. Because of the iterative nature of the subdivision scheme discussed in Section 3, attention should be paid to the first two properties.

In terms of self-similarity, the object resulting from the subdivision scheme fulfills this property in a broad manner—after the first refinement step, all faces are pentagons. Therefore, from the second refinement step onward, all faces and their replacements are combinatorially self-similar. However, due to the smoothing operation, this self-similarity does not hold geometrically, i.e., the pentagons cannot be scaled to be pairwise congruent.

While self-similarity cannot be established, the situation is different when regarding the property of fractal dimension. Consider the edges of the subdivided pentagon in Figure 8. As discussed in Section 3, in each refinement step, every edge is replaced by three edges in the shape of a stretched Z, see Figure 6a. As known from the Koch curve, which is defined by replacing an edge --- by $\text{---}\wedge\text{---}$, this procedure of replacing edges can be formulated in terms of a Lindenmayer system, see [23, Ch. 1] for theory of these systems and Figure 10 for the concrete system relevant for the presented subdivision scheme.

The three edges constructed in operation 1 of Section 3 are constructed with equal length and smaller enclosed angle $\frac{2\pi}{3}$. Thus, in each refinement step, the new boundary edges are $\frac{1}{\sqrt{7}}$ times as long as in the previous step. Furthermore, there are three times

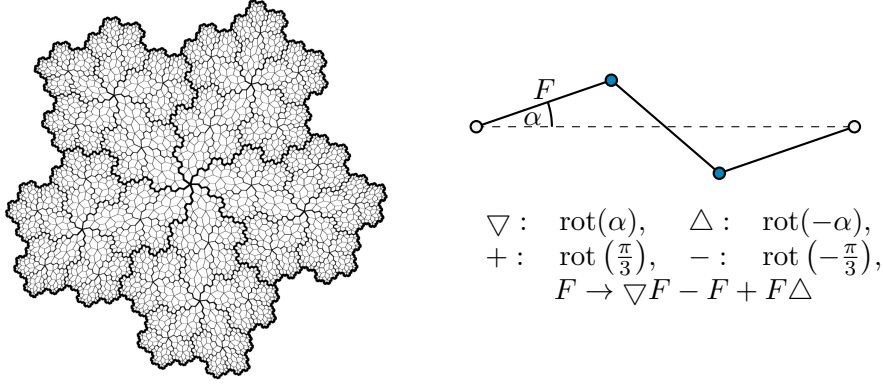


Figure 10.: Fractal curves contained in the edge set after five refinement steps; corresponding Lindenmayer system, for which α can be calculated from the known quantities.

as many boundary edges, thus the total length of the boundary grows by the factor $\frac{3}{\sqrt{7}}$ in each step. Hence, the limit boundary curve has a fractal dimension of

$$D = -\frac{\log(3)}{\log(1/\sqrt{7})} \approx 1.12915.$$

Note that this statement holds for the boundary curve shown in Figure 10 and more generally for the resulting boundary curve of any reasonably regular input mesh. Inner edges are, once they are created in a refinement step t , iteratively replaced by smaller edges. This amounts to the generation of $3^{t'-t}$ edges in a later refinement step t' . Visually, these curves are very similar to the curves at the boundary (see Figure 10), but because of the smoothing operation applied to inner vertices, computing the total length of these curves in a given refinement step t is not as easy as at the boundary.

In regard to the curves involving inner vertices, we can offer some observations. First of all, there are two types of these curves: Those that have at least one fixed point as end-point and those that have two inner vertices at end-points. Here, fixed points are vertices that are not moved during a refinement step. Such points lie, e.g., on the boundary curve as the vertices of the boundary remain in place after their creation, because the smoothing step does not apply to them. Also, the center point of the pentagon is a fixed point because of symmetry. Therefore, the lengths of the five curves running from the center point to the boundary are bounded from below by the distance of the center point to the end-point on the boundary. This is not generally true for curves running between two inner vertices, as subsequent smoothing steps can move the end-points closer to each other. Numerical experiments suggest that curves between inner vertices do grow with repeated executions of refinement steps. However, the growth did not happen with a unique factor, nor did the growth factor change monotonically. This motivates the following (set of) open question(s):

Open Question 2. *Are the inner curves following a fractal pattern? If so, what is their fractal dimension? How do they, if at all, relate to the boundary fractal?*

To explore the self-similarity of the presented images further, we explore two rendering options for an iteratively refined pentagon. The first is based on edges, while the

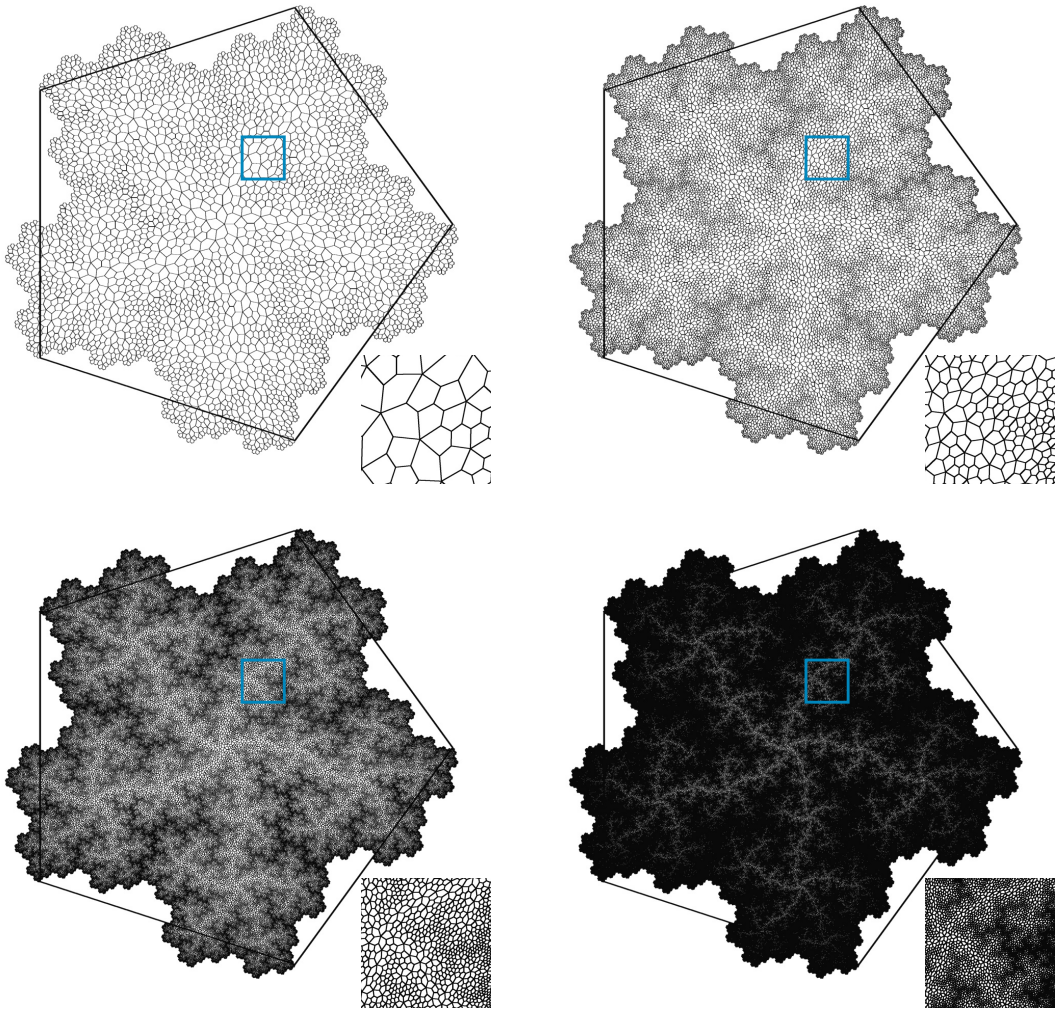


Figure 11.: Repeated pentagon snub subdivision refinement applied to a regular pentagon. From upper left to lower right, figures show 5th, 6th, 7th, and 8th refinement step. Shown are the edges resulting from the corresponding refinement step.

second is based on vertices. The images in Figure 11 continue the series of images from Figure 8, i.e., they show the fifth to eighth refinement step of the underlying pentagon. As discussed in the previous two paragraphs, the edge lengths of the newly inserted edges shrinks with each refinement step. Thereby, structures arise that vaguely remind of lightning during a thunderstorm. In the current resolution, the next—ninth—refinement step would show a completely black picture. Note the gradually arising self-similarity as provided by the zoom-in boxes.

In contrast to the illustrations of Figure 11, in the image in Figure 12, each pixel is colored according to the first refinement step, in which it contains a vertex of the iteratively subdivided regular pentagon. The later this pixel is hit by a vertex, the brighter it is colored. This is most notable on the boundary curve of the subdivided pentagon, whose pixels are created fairly early on in the process. In comparison, the way along one of the wave-like curves from one of the corners of the original pentagon to the center point takes longer to be refined. Thus, in the center, around one dark

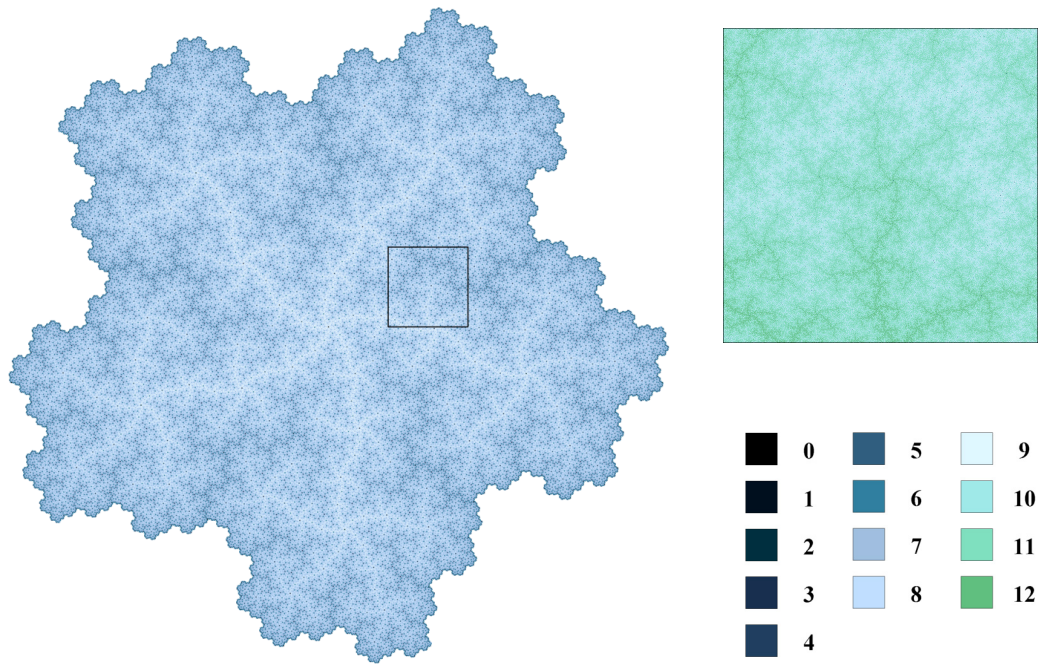


Figure 12.: Pixels colored according to the refinement step in which the first vertex is created within this pixel. The five vertices of the input pentagon are colored in step 0. Throughout the refinement steps 1–9, colors get gradually brighter, while for the refinement steps 10–12, they gradually become green. The left image shows the subdivided pentagon after refinement step 9, while the right image shows the high-lighted version after refinement step 12.

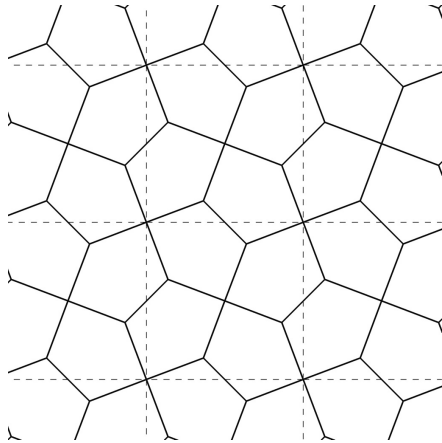
pixel created in the very beginning, there is a collection of comparably bright pixels that are filled late in the process.

Even though the refinement process can be repeated infinitely often, this image does not change after a finite number of refinement steps, because each inner pixel is painted after nine refinements. Similar to the well-known images of the Mandelbrot set [22, Part IV], one can zoom into the image and compute a new image for a smaller area of the pentagon to a finer level, producing self-similar images, see the corresponding zoom-in in Figure 12.

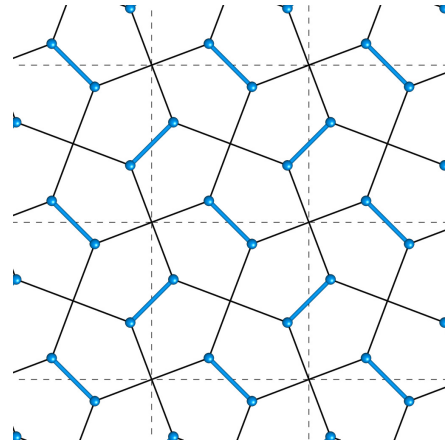
5. Weaving patterns resulting from the pentagon snub subdivision scheme

Aside from fractal properties, the subdivided object exhibits another interesting behavior, when coloring pairs of its pentagons appropriately. To illustrate the coloring, consider a regular square grid to which one refinement step of the pentagon snub subdivision scheme is applied, see Figure 13a.

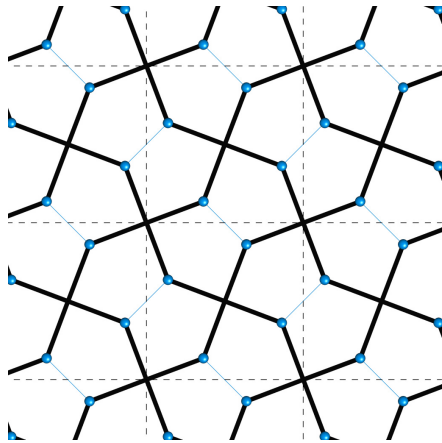
Note that by construction, each of the pentagons shown has exactly one edge that arises as the middle edge of a Z-triplet. Furthermore, when fixing a pentagon, the vertex opposite to said edge is incident to the middle edge of another Z-triplet, confer



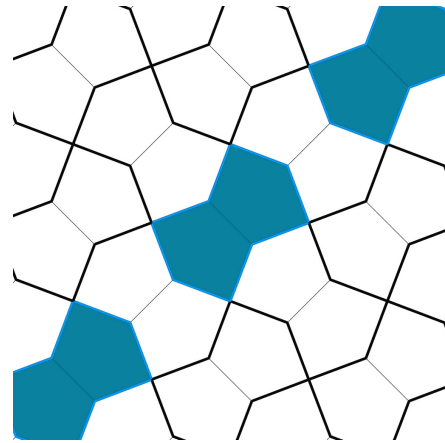
(a) 1st refinement step on a regular square grid.



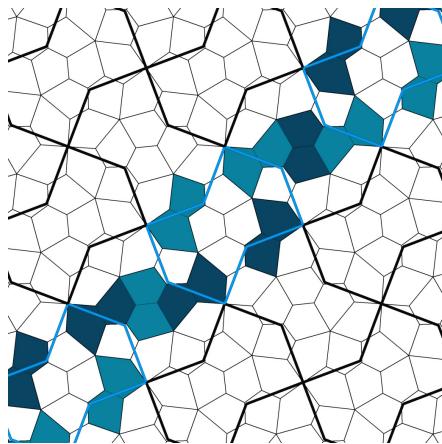
(b) Highlighting the middle edges of all Z-triplets.



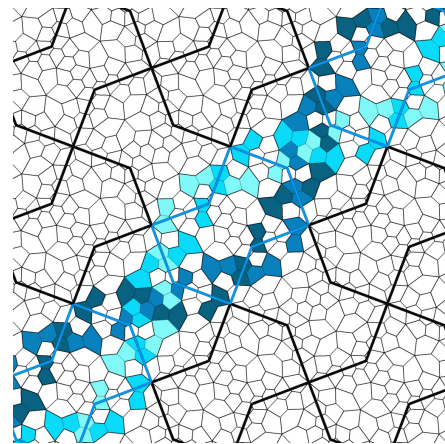
(c) Glue pentagons along the middle Z-triplet edges.



(d) Color a strand, following central Z-triplet edges.



(e) 2nd refinement: a strand splits into two twisted strands.



(f) 3rd refinement: a twisted pair of twisted strands.

Figure 13.: Creating a weaving pattern on a refined square grid by gluing tiles and coloring them systematically.

the highlighted middle edges in Figure 13b. By pairwise gluing all pentagons that share a middle edge of a Z-triplet, the mesh is reduced to octagons, see Figure 13c.

The coloring is applied as follows: An initial octagon \mathcal{O} and a color for \mathcal{O} are chosen. By construction, if \mathcal{O} does not contain a pentagon on the boundary, exactly two vertices of \mathcal{O} are incident to middle Z-triplet edges outside of \mathcal{O} . These two edges connect \mathcal{O} to two other octagons \mathcal{O}' and \mathcal{O}'' . Both are colored in the same color as \mathcal{O} and the coloring is propagated in this way across the pentagon mesh. For a single color, this results in a strand, see Figure 13d.

When refining the mesh further, these colored strands are split into further pentagons and corresponding glued octagons. These can be colored as strands again and it can be observed that each original strand splits into two smaller, interwoven ones. This behavior is illustrated in Figure 13e, where the glued octagons from the first refinement step are still visible and the octagons from the second refinement step clearly weave along the former strand. In fact, this observation generalizes: With each additional refinement step, each strand is split into two interwoven strands that follow their parental one. Thus, after the third refinement step, each original strand from the first step is split into four twisted strands, see Figure 13f.

In fact, the observations made for the coloring of the refined square grid solely depend on properties of the subdivision scheme. Therefore, the coloring procedure can be applied to any bounded or unbounded 2-manifold mesh subdivided by the pentagon snub subdivision scheme.

In the following, we will continue the investigation of the pattern induced by the snub pentagon subdivision scheme. First, we discuss our naming choice (calling it a “weaving”) and discuss related textile analogies. Following this, we present a physical realization of the weaving pattern via paper models. The final part of this section is devoted to generalizing this illustrative approach of weaving patterns to other subdivision schemes aside from snub pentagon subdivision.

5.1. Naming choice: Why call the pattern a “weaving”?

The creation process of the pattern, as outlined above for the regular square grid, has two central components that provide an allusion to traditional textile processes. This is, on the one hand side, the creation of intertwined strands. These arise, as a refinement step splits each strand present before, compare Figures 13d and 13e. On the other hand side, in each refinement step, the pattern is created via strands that overlap each other. This can be seen in Figure 13d, where the colored strand passes below two strands running orthogonally.

Focusing on the first element, as shown in Figure 13d, the initial application of the subdivision scheme creates single strands. However, subsequent subdivision refinement steps split these strands, see Figures 13e and 13f. Note that these subdivided strands still follow the direction of the original one. After the second refinement step, the single initial strand is split into two strands that are crossing above and below each other while following said direction. These two strands are then split by the third refinement step into two even smaller strands respectively. Each pair is intertwined and crosses above and below, while the pairs themselves are also intertwined, crossing above and below each other, just as the parental strands did. This behavior of the strands, following a common direction while being intertwined, i.e., crossing above and below each other, is reminiscent of *braiding* or *plaiting* as the result is “considerably greater in length than in width or diameter” [26, p. 38].

However, we do not consider the braiding of the individual strands to be the visually dominant piece of the final pattern. This is rather given by the crossing of the strands with other strands that run into a different direction. That, in turn, is reminiscent of the process in weaving, where “it is thus possible to obtain at least two distinct interlacings of warp which are separated by the weft thread” [26, 71]. This perspective does, at least at a global level, describe the behavior of the different strands in the pattern described here. This motivates our naming choice of calling the resulting pattern a *weaving*.

As a final remark, note that this naming choice has to be taken with a pinch of salt. In classical weaving processes, the thread that provides the weft is completely moved below or above the threads that form the warp. First, this means that the warp is fixed, while only the weft is moved actively. However, in the following (see Figure 15) we will see that this perspective is not always correct for more complex scenarios than the regular square grid. In those, “[b]oth thread groups are equally active” [26, p. 42], which classifies active-active intertwining. Second, in weaving, all strands that form the thread, are moved above or below another thread together. Figure 13f reveals that the situation is more delicate for the pattern considered here. At points where two directions meet orthogonally, their individual strands do interweave in a complex pattern. Some smaller strands go below those from the orthogonal direction, while others move above. This technique is called *ply-splitting*, see [26, p. 43]. More specifically, in the case of the regular square grid pattern, we encounter a *right-angled or diagonal ply-splitting*, see [26, pp. 44/45]. In these two aspects, the pattern investigated here is not a weaving. However, as the general visual impression is still reminiscent of a weaving pattern, we stick with this terminology. This is to be kept in mind throughout the following discussion. Similarly, our reference to a *weaving pattern* does not imply the explanatory or procedural component that is carried by, e.g., a knitting pattern. Our patterns are simply the induced patterns on the geometry.

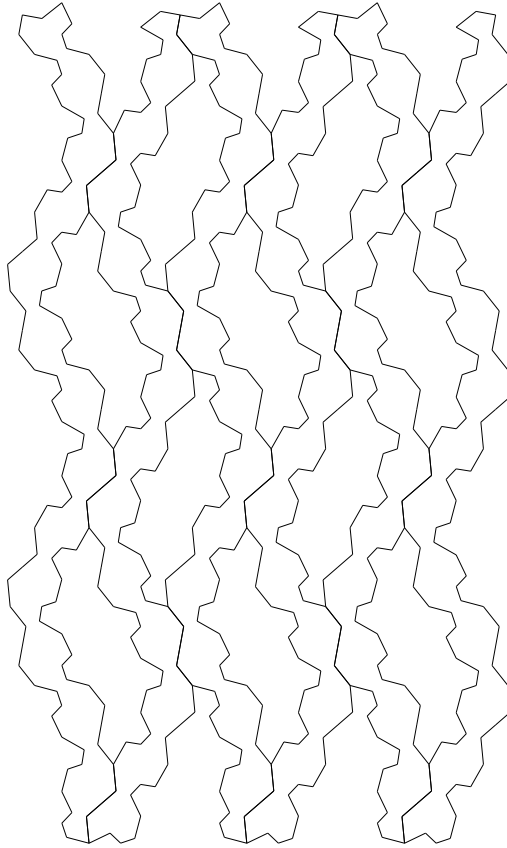
5.2. Physical realization of the pattern

Before we turn to considering the impact of this weaving pattern for other subdivision schemes than the pentagonal snub subdivision, we will briefly investigate the possibility of creating physical models from the above observations. This is to assert that the presented pattern does not only provide a coloring, but a pattern that can be physically realized via a weaving process.

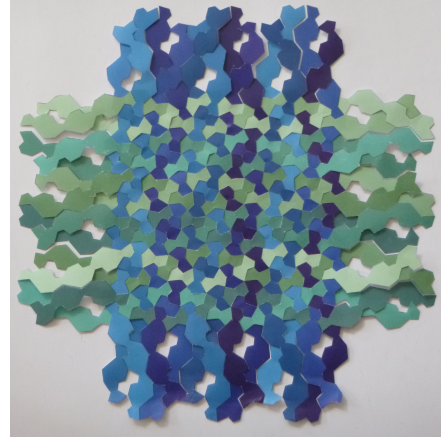
Starting with the simple case of a subdivided square grid as presented in Figure 13, it helps to rotate the pictures by $\frac{\pi}{4}$, as the strands thereby obtain a horizontal or vertical alignment. We can now fabricate a pattern to be printed on paper and cut out to provide a set of strands, which can subsequently be intertwined to provide the final pattern.

A possible module² to be used is shown in Figure 14a. This specific design is meant to be printed four times, providing a north-south, west-east, south-north, and east-west direction when assembling a weaving on the square grid as shown in Figure 13. That is, when assembling the pattern, the module as shown in Figure 14a has to be printed four times, where three of the four copies have to be rotated by $\frac{\pi}{2}$, π , and $\frac{3\pi}{2}$, respectively. Periodic repetition of this module or an appropriate reduction allows for the realization of larger or smaller number of squares in the grid, respectively. Figures 14b and 14c show corresponding results using different color schemes. The

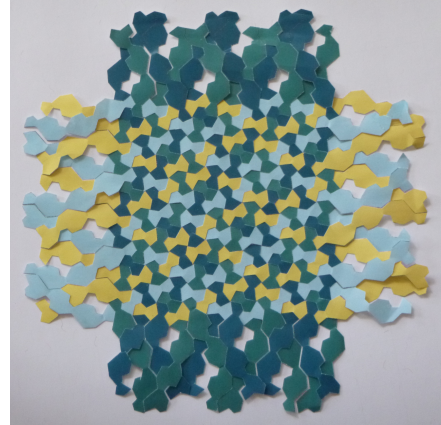
²If you are interested in giving the process a try, find this module in the supplemental material to the article.



(a) One set of paper modules.



(b) Physical realization with paper, highlighting intertwined strands (size: 27 cm×27 cm).



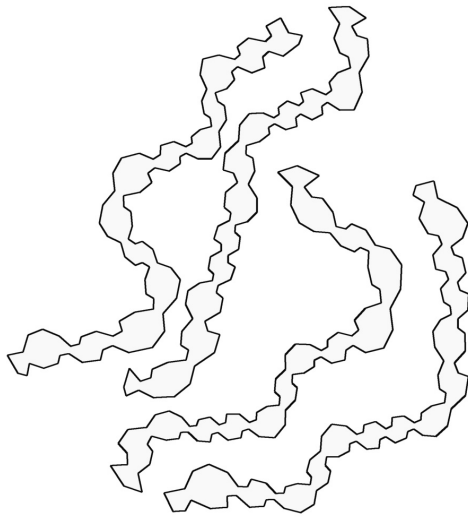
(c) Paper model colored with minimal number of colors to separate the modules (size: 24.5 cm×24.5 cm).

Figure 14.: A single paper cut-out module, a paper model colored according to the description above, and a paper model colored such that every cut-out module is colored differently. Both physical models show weaving of a twice-subdivided square grid.

underlying regular square grid has been subdivided via the pentagon snub subdivision scheme two times. The physical model covers several squares of the grid, which lies at a $\frac{\pi}{4}$ angle to it.

The color schemes are chosen from a maritime palette to underline the wave-like, flowing behavior of the different strands. Furthermore, they illustrate the two different aspects of the pattern as discussed in Section 5.1. The first scheme, utilizing twelve colors, highlights the intertwined pairs of strands that arise from subdividing single strands. It therefore alludes to the braiding component involved. The second scheme, using a reduced palette of four colors, provides a better overview of the general pattern created. Hence, the resulting model rather highlights the weaving part of the pattern.

The modules used for the presented paper realizations from Figure 14 were printed on A4-sized printer paper and cut using scissors and a pen knife. In case of thicker material or a higher number of repetitions of the depicted block, the use of a laser



(a) One set of paper modules.



(b) Physical realization of Figure 8 as a paper model (width/height: 23 cm \times 22.5 cm).

Figure 15.: A set of paper models and a paper model colored according to the description above. The underlying input is a pentagon with five adjacent pentagons that have been subdivided three times via the pentagon snub subdivision scheme.

cutter might be handy as the cutting part was by far the most time-consuming. We do not recommend to print the module in a smaller size since this makes the strands thinner and causes them to tear more easily. While thicker material prevents this, it will likely be harder to assemble. While assembling the model, it is helpful to fix the top of those strands parallel to the warp direction. Then, with every weaving step, align two strands in weft direction. This procedure worked better in our experiments than completely twisting two parallel strands and trying to include the orthogonal ones afterwards, which was unsuccessful. For better orientation, we refer to a coloring such as shown in Figure 14c, which helps to align the modules.

The above discussed a paper model realization of the pentagon snub subdivision scheme when applied to a regular square grid, from which we derived the depicted weaving pattern. As a closing piece for this section, we offer a paper model realization of the subdivided pentagon as shown in Figure 8. Again, the pattern was printed on printer paper and cut using a pen knife. In contrast to the pattern derived from the square grid, here, five copies of the set of modules have to be used, representing the directions as induced by each side of the underlying pentagon. See Figure 15a for a single module set and Figure 15b for the assembled paper model. As a color scheme, we chose to extend the maritime palette used before. Note that this model is substantially harder to assemble due to the lack of orthogonality in the meeting of strands. When assembling it, we found that it is best to work as a pair as some strands need to be connected to several other ones in order to prevent the entire structure from falling apart. The underlying input is a pentagon with five adjacent pentagons that have been subdivided three times via the pentagon snub subdivision scheme.

5.3. Transferring the patterns to other subdivisions

The previous subsection was concerned with physical paper model realizations of the weaving pattern. Now, we are interested in generalizing the pattern to a broader range of subdivision schemes, aside from pentagon snub subdivision. We tackle this question by considering certain classes of meshes.

5.3.1. Meshes from quadrilaterals

Focusing on quadrilateral-based meshes first, we can make the following observation: Taken any quad-mesh, if the vertices are *two-colorable* (i.e., if we can color all vertices of the mesh by either color c_1 or color c_2 such that no two vertices connected via an edge have the same color), the quad mesh can be interpreted as a weaving pattern. This is done by considering each quadrilateral to be the crossing of two strands, one above, one below. In order to determine, which strand goes where, we consider the edges of the quadrilateral. If a strand “enters” a quadrilateral via an edge between a c_1 -vertex on the left and a c_2 -vertex on the right, it will be on top, if the order is switched, it will be below. Consider Figure 16a for an illustration of this, where a strand changes from above to below when passing an edge with a c_1 vertex on the left and a c_2 vertex on the right.

The condition of admitting to a two-coloring can be easily satisfied for any quadrilateral mesh that is a topological disc. It is done by coloring the vertices via a disc growing approach: Color a first vertex without loss of generality by color c_1 , color all vertices that are connected to it with color c_2 , and continue to their neighbors, etc., while always alternating colors. Given this observation, let us revisit the quadrilateral-based subdivision schemes discussed in Section 2.

By the above, a mesh subdivided by the scheme of Catmull-Clark [6], can be interpreted as a weaving. This holds in particular also for meshes of arbitrary topology by performing the following two-coloring after performing a refinement step: Assign color c_1 to all new vertices inserted on the edges during the subdivision, while assigning color c_2 to all old vertices as well as to those new vertices at the face centers (compare Figures 5b and 16b, where edge-vertices are colored white and all other vertices have been colored blue). This will lead to a two-coloring, independent of the topology of the mesh.

If the input mesh already did admit to a weaving pattern via a corresponding two-coloring, a Catmull-Clark refinement step splits one strand into two parallel strands (as opposed to the pentagonal subdivision scheme, where the strands are intertwined), see Figure 16c for an illustration of this. Information of the weaving, as to what strand is above or below, is lost during a refinement step, see Figure 16.

By the above, for the subdivision scheme of Catmull-Clark [6], we can introduce a weaving pattern based on the property that after each refinement step, all resulting faces are quadrilaterals. In the case of the *mid-edge* subdivision scheme [21], this is only the case if the input mesh consists solely of quadrilateral faces and vertices of degree four, as every other face will simply be replicated and every vertex of degree different to four will give rise to a face of corresponding degree. This only leaves the regular square grid as valid input for which we can apply the technique given above. Turning to the subdivision scheme of Doo-Sabin [9], as illustrated in Figures 4 and 5c, one refinement step via Doo-Sabin is equivalent to taking two *mid-edge* refinement steps. Therefore, for both subdivision schemes, the only input which can be subdivided and associated to a weaving pattern via the strategy used above is the regular square grid,

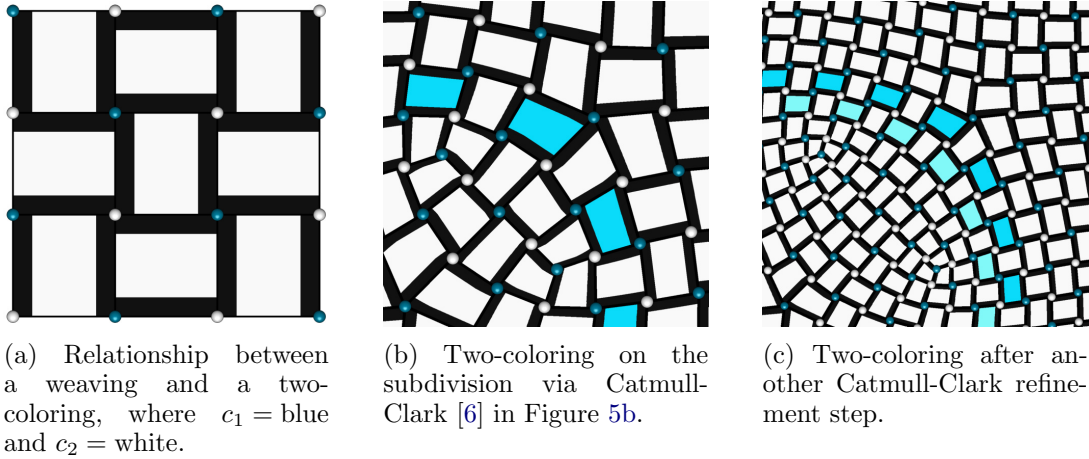


Figure 16.: Weaving from two-coloring: White-blue edges have the strand go on top, while blue-white edges have it go below. Also: Application of the weaving pattern to the subdivision scheme of Catmull-Clark [6].

with a resulting weaving pattern as shown in Figure 16a.

5.3.2. Meshes from triangles

Turning to triangle meshes, we can make another general observation. Let us assume that the considered mesh is purely composed of triangles and that the vertices of the mesh can be colored using two colors c_1, c_2 such that every triangle has exactly one vertex of color c_1 and two vertices of color c_2 . Given this, we can remove all edges from the mesh that span between two vertices colored by c_2 . Thereby, we obtain a mesh made entirely of quadrilaterals, which satisfies the two-coloring criterion described above. Hence, we can impose a weaving on corresponding triangle meshes by gluing pairs of triangles to form quadrilaterals, analogous to the pentagonal case, where we glued pentagons to octagons.

In case of both the *butterfly* subdivision scheme and the subdivision scheme by Loop, if we assume that the input mesh satisfies the coloring criterion sketched above, after applying a refinement step, we can obtain a valid new coloring as follows: keep colors for all old vertices, new vertices on an edge between a c_1 and a c_2 vertex are colored by c_2 , and new vertices on edges between two c_2 -vertices are colored by c_1 . This yields a new triangle mesh that once more satisfies the coloring property. See Figure 17 for an illustration of the triangle gluing as well as the behavior of the weaving under subsequent refinement steps.

Turning to the $\sqrt{3}$ subdivision scheme, we find that it always creates a weaving pattern, independent of the mesh it is applied to. This can be seen as follows: If we remove those edges that have just been flipped in the refinement step, we obtain a set of quadrilaterals, where each quadrilateral contains two vertices of the original mesh and two vertices created as face midpoints, pairwise diagonally in the quadrilateral. Thus, once more, we obtain a two-coloring on a set of quadrilaterals and hence a weaving by gluing sets of triangles. However, the behavior of the weaving pattern under subdivision is interesting. New strands are not parallel, but orthogonal to the strands of the previous step. Only after performing two refinement steps, each strand is separated into three, parallel strands, see Figure 18.

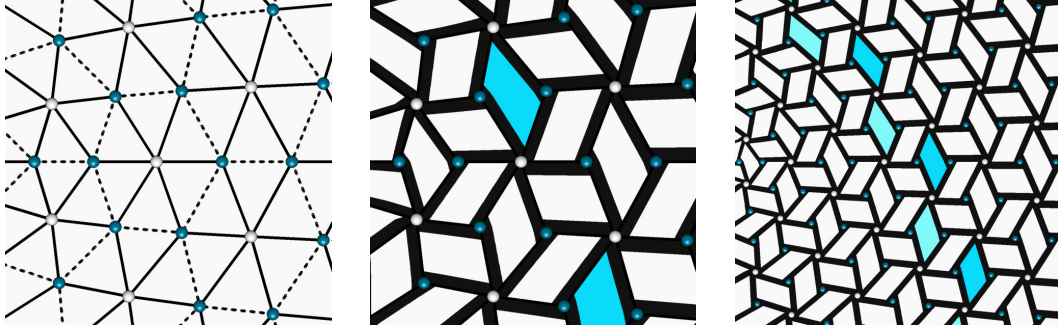


Figure 17.: Weaving patterns from the subdivision scheme by Loop. Remove dashed edges between vertices both colored by the same color. Applying the subdivision splits a strand into two parallel ones.

5.4. A general weaving pattern for meshes

The previous sections discussed the weaving pattern for the pentagon snub subdivision scheme, suitable quadrilateral or triangle meshes, and their corresponding subdivision schemes. For some combinations, no topological or combinatorial restriction are necessary, while other need to satisfy certain properties. While the weaving patterns for meshes composed of quadrilateral faces directly work on these, the patterns for faces of uneven edge numbers, i.e., triangles and pentagons, work by gluing pairs of faces together and thereby introducing a weaving on a set of quadrilaterals or octagons. Here, we present a final weaving scheme that is independent of the underlying mesh and the subdivision scheme. Instead of gluing faces together, it splits the faces of the mesh.

Given a mesh, possibly obtained from some subdivision scheme, we perform the following steps. First, for each face of the mesh, the face mid-point is added as new vertex to the mesh and connected via an edge to all vertices of the parental face. Then, the original edges of the mesh are deleted. Thereby, for each pair of faces that share an edge, their two midpoints, together with the two vertices of the edge, form a quadrilateral. This gives rise to a pure quadrilateral mesh on which a weaving pattern can be induced as described above, see Figure 19 for an illustration of these steps. Note that the process of Christian Mercat employs the same rationale to create Celtic links [20]. Furthermore, by following the outlined procedure, we obtain the same

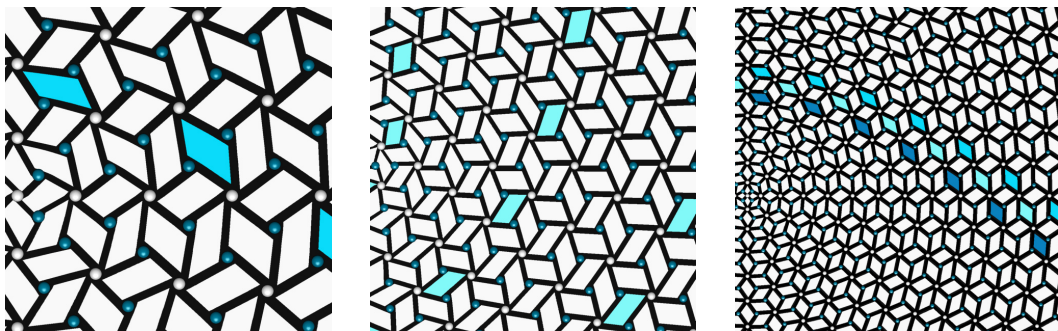


Figure 18.: Weaving patterns from the $\sqrt{3}$ subdivision scheme. Applying the subdivision twice splits each original stand into three.

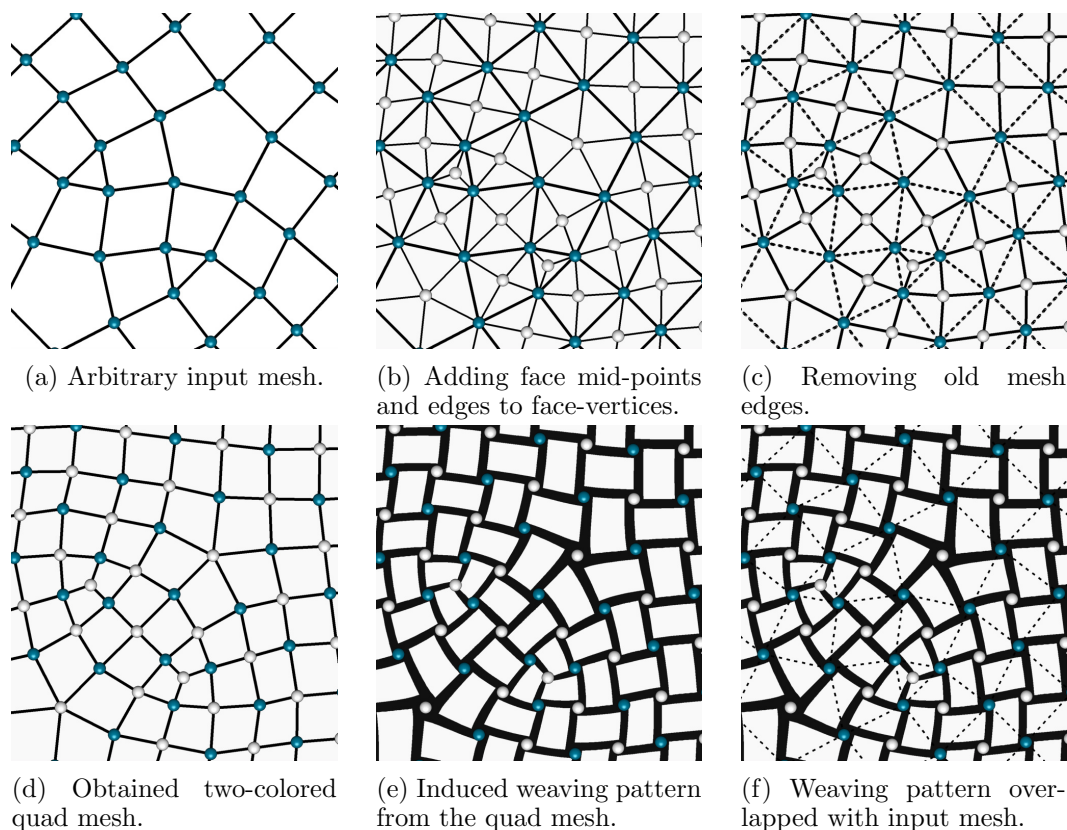


Figure 19.: Weaving pattern for an arbitrary input mesh.

combinatorics that are present in the weaving sculptures of Alison Martin [19].

Just as for the weaving patterns induced by the triangle-based subdivision schemes and the pentagon snub subdivision scheme, in this case, the crossing of the weaving happens on an edge of the original mesh, see Figure 19f. This is in contrast to the weaving patterns as described in Section 5.3.1 build on quadrilateral meshes. There, the crossings of the weaving pattern take place within the faces of the original input mesh. For meshes with faces composed of an uneven number of sides, crossings on top of the original faces might only be possible by allowing, e.g., a different number of strands to enter a face than to leave it, or by including splits in strands on such faces. Aside from the weaving patterns described above, this motivates the following open question.

Open Question 3. *What additional, possible weaving patterns exist for specific of general meshes, where the crossing happens inside the input faces? In particular, what are corresponding patterns for the mid-edge subdivision scheme [21] or to the subdivision scheme of Doo-Sabin [9]?*

6. Jigsaw puzzle artwork

The weaving pattern described in the previous section completes the set of theoretical observations on the pentagon snub subdivision scheme. This concluding section



(a) The jigsaw puzzle illustrating various theoretical aspects discussed in this paper (diagonal 50 cm).

(b) A border piece, a two-colored piece, and a uni-colored piece.

Figure 20.: Final illustrating piece using the various possibilities of the discussed pentagon snub subdivision scheme.

describes an artistic piece that illustrates the theory and makes it tangible beyond the presented paper models, which are somewhat cumbersome to produce and require a high level of dexterity to assemble. The concept of subdivision schemes—with their property of splitting large domains into smaller, as-regular-as-possible pieces—immediately suggests the creation of a jigsaw puzzle for their illustration. This is a specifically good choice in this case, because it combines visual information via coloring the pieces as well as geometric information via the shape of the pieces. In particular the latter adds a tactile experience to the illustration that is not present in images, animations, or computer simulations of subdivision. Prominent examples of previous successful combinations of jigsaw puzzles and mathematics can be found at *Kadon Enterprises, Inc.* [14] or at *nervous system* [24].

The starting point for the specific jigsaw puzzle presented here in Figure 20 is a regular pentagon that is subdivided four times via the pentagon snub subdivision scheme as discussed in Section 3. The resulting pentagons are then glued as described in Section 5 to create octagonal puzzle pieces. However, not all pentagons have a “partner” to be glued to. To overcome this, note that the smoothing operation is not applied to vertices along the boundary curve of the subdivided pentagon. Thus, all outer angles along the boundary are $\frac{2\pi}{3}$ or $\frac{4\pi}{3}$. This can be continued to an almost complete ring of perfect hexagons, which serve as gluing partners to those pieces that are left over. Thereby, we create some enneagonal pieces along the boundary. In particular, the hexagonal side of the boundary pieces helps with their placement as their hexagonal part fixes their inward/outward orientation.

The jigsaw puzzle is made from laser cut poplar plywood and is painted with acrylic paint. The diagonal of the surrounding pentagon measures 50 cm. The edges made by the laser cut represent puzzle pieces obtained from the fourth refinement step, as described above. However, the painted overlying tiling shows the weaving pattern associated to the third refinement step with a total of 20 twenty different colors. Hence, most tiles are painted in two different colors. Those tiles that only exhibit a single color are always at the center of the larger octagons from the coarser weaving pattern, see

Figure 20b. The color scheme is reduced to three primary colors—green, blue, and pink. The different chosen shades made it possible to pair the interwoven strands within the same color class but still to distinguish them from the others. For example, the light and dark green strands originate from one strand and are thus interwoven.

As touched upon in a blog post by the jigsaw puzzle designers from *nervous system* [17], clean laser cutting is a non-trivial task. The pieces and the frame of the jigsaw puzzle presented here have all been cut from a single piece of plywood. Despite the laser cutter’s high precision, the high number of cuts resulted in a certain leeway for the pieces. However, after painting the pieces, these gaps closed by the wood swelling slightly. When repeatedly creating the jigsaw puzzle, it could be beneficial to cut all pieces and the puzzle frame separately, taking into account the width of the introduced cuts, in order to create an even better fitting result.

The jigsaw puzzle is an illustrative collection of the aspects touched upon in this paper. It does not only show the result of a refinement, but also the subdivision process itself as the coloring presents a coarser step than the individual puzzle pieces. This is supported by the uni-colored pieces that always have to be placed at the center of one of the elements of the coarser step. Hence, the jigsaw puzzle adds a different, illustrative level as opposed to the paper models. Handling the puzzle pieces provides a sense of how the different pentagons strive towards being regular. However, the applied replacement by Z-triplets, operation 1, is also visible in the coloring: Those pieces that do exhibit two colors are not split into two convex pentagons. Rather, they are split diagonally, which shows the replaced edge, while the two edges of the puzzle piece that slightly diverge from this edge indicate the two outer edges of the Z-triplet, see Figure 20b. Furthermore, assembling the jigsaw immediately introduces the uniform degree of all inner vertices present, as the convex corners of the pieces come together in multiples of three or five. The fractal boundary curve is alluded to by the hexagonal pattern at the boundary of the jigsaw puzzle and its regular pattern does invoke images of the related Koch curve. When completing the puzzle, the weaving patterns—that are present in any mesh resulting from the pentagon snub subdivision scheme—serve as a guiding motif. The split of a strand into two interwoven strands that occurs with any further refinement step is encapsulated by the choice of colors and their shades.

Our jigsaw puzzle also offers a different coloring and thus a different assembly experience on its back-side, see Figure 21. While the tiles on the front-side are colored according to the weaving pattern of the previous refinement step, the back-side shows the weaving corresponding to the current refinement step. Thus, while the front side has two different kinds of pieces, those with one or those with two colors, the pieces on the back side only carry one color each. Therefore, this back coloring follows exactly the procedure as given in Section 5. The colors of the strand pairs on this side of the jigsaw puzzle are chosen in a way such that they correspond to the front-side color of the enclosed puzzle-piece, see Figure 21b. In particular, this side of the puzzle emphasizes the complex textile-like pattern behavior as discussed in Section 5.1. This becomes apparent when comparing the drawn patterns on the jigsaw puzzle with the illustration of active-active intertwining, taken from [26], see Figure 21a. It is particularly interesting to note the differences between the patterns. For instance, the strands shown on our jigsaw puzzle always alternate between going above and below other strands, which is not true for the active-active intertwining pattern.

The paper model, with its strands of paper, illustrates the weaving component of the introduced coloring pattern. The jigsaw puzzle adds to this by making the different refinement steps tangible, which offers manifold opportunities for exploration of the illustrated mathematics. Hence, both models are good companions whenever

the weather report does announce the next rainy afternoon. They might well teach the explorers about some underlying principles of said report.

Acknowledgments

This research was partially funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – 455095046.

References

- [1] Ergun Akleman and Vinod Srinivasan. Honeycomb subdivision. In *Proc. 17th Int. Symp. Computer & Information Sc.*, pages 137–141, 2002.
- [2] Ergun Akleman, Vinod Srinivasan, Zeki Melek, and Paul Edmundson. Semiregular pentagonal subdivisions. In *Proceedings Shape Modeling Applications, 2004*, pages 110–118. IEEE, 2004.
- [3] Lars-Erik Andersson and Neil F. Stewart. *Introduction to the mathematics of subdivision surfaces*. SIAM, 2010.
- [4] Philip Bowers and Kenneth Stephenson. A “regular” pentagonal tiling of the plane. *Conformal Geometry and Dynamics of the American Mathematical Society*, 1(5):58–86, 1997.
- [5] Gunnar Brinkmann, Pieter Goetschalckx, and Stan Schein. Comparing the constructions of Goldberg, Fuller, Caspar, Klug and Coxeter, and a general approach to local symmetry-preserving operations. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 473(2206):20170267, 2017.
- [6] Edwin Catmull and James Clark. Recursively generated b-spline surfaces on arbitrary topological meshes. *Computer-aided design*, 10(6):350–355, 1978.
- [7] John H. Conway, Heidi Burgiel, and Chaim Goodman-Strauss. *The symmetries of things*. CRC Press, 2016.
- [8] Daniel Doo. A subdivision algorithm for smoothing down irregularly shaped polyhedrons. *Computer Aided Design*, pages 157–165, 1978.
- [9] Daniel Doo and Malcolm Sabin. Behaviour of recursive division surfaces near extraordinary points. *Computer-Aided Design*, 10(6):356–360, 1978.
- [10] Nira Dyn, David Levin, and John A. Gregory. A butterfly subdivision scheme for surface interpolation with tension control. *ACM transactions on Graphics (TOG)*, 9(2):160–169, 1990.
- [11] Bertil Gustafsson. *Scientific Computing: A Historical Perspective*, volume 17. Springer, 2018.
- [12] Ayman Habib and Joe Warren. Edge and vertex insertion for a class of C1 subdivision surfaces. *Computer Aided Geometric Design*, 16(4):223–247, 1999.
- [13] George W. Hart. Conway notation for polyhedra. Online: https://www.georgehart.com/virtual-polyhedra/conway_notation.html, 1998. Accessed May 28, 2021.
- [14] Inc. Kadon Enterprises. Gamepuzzles: for the joy of thinking. Online: <http://www.gamepuzzles.com/index.htm>, 1980. Accessed June 1, 2021.
- [15] Leif Kobbelt. $\sqrt{3}$ -subdivision. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 103–112, 2000.
- [16] Charles T. Loop. Smooth subdivision surfaces based on triangles. Master’s thesis, University of Utah, Department of Mathematics, 1987.
- [17] Jesse Louis-Rosenberg. The secret to clean laser cutting: designing a better laser nozzle using 3d-printing. Online: <https://n-e-r-v-o-u-s.com/blog/?p=8722>, 2020. Accessed June 1, 2021.
- [18] Benoît Mandelbrot. *The fractal geometry of nature*. W. H. Freeman, New York, 1983.

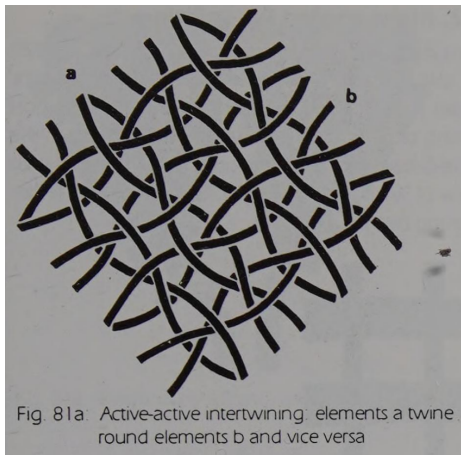
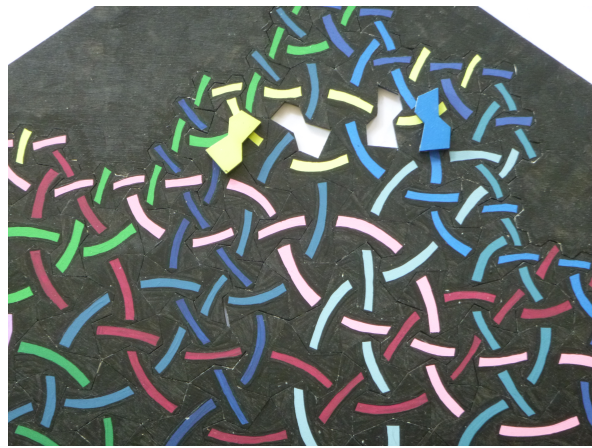


Fig. 81a Active-active intertwining: elements a twine round elements b and vice versa



(a) Figure 81a from [26], illustrating active-active intertwining.

(b) Two pieces flipped, their color matches that of the enclosed strands.



Figure 21.: Alternatively colored back-side of the jigsaw puzzle (diagonal: 50 cm).

- [19] Alison Grace Martin. Superficial study 1–3. Online: <http://gallery.bridgesmathart.org/exhibitions/2013-bridges-conference/alison-martin>, 2013. Accessed March 31, 2022.
- [20] Christian Mercat. Théorie des nœuds et enluminure celte. *L'Ouvert*, 84:1–22, 1996.
- [21] Jörg Peters and Ulrich Reif. The simplest subdivision scheme for smoothing polyhedra. *ACM Transactions on Graphics (TOG)*, 16(4):420–431, 1997.
- [22] Clifford A. Pickover. *Chaos and Fractals: A Computer Graphical Journey*. Elsevier Science Publishers B. V., NLD, 1998.
- [23] P. Prusinkiewicz and J. Hahn. *Lindenmayer Systems, Fractals, and Plants*, volume 79 of *Lecture Notes in Biomathematics*. Springer-Verlag, New York, 1989.
- [24] Jessica Rosenkrantz and Jesse Louis-Rosenberg. nervous system: jigsaw puzzles. <https://n-e-r-v-o-u-s.com/shop/line.php?code=12>, 2007. Accessed June 1, 2021.
- [25] Malcolm A. Sabin. Recent progress in subdivision: a survey. In N. A. Dodgson, M. S. Floater, and M. A. Sabin, editors, *Advances in Multiresolution for Geometric Modelling*, chapter 11, pages 203–230. Springer, Berlin, Heidelberg, 2005.
- [26] Annemarie Seiler-Baldinger. *Textiles: a classification of techniques*. Smithsonian Institution Press Washington DC, 1994.
- [27] Min Yan. Pentagonal subdivision. *The Electronic Journal of Combinatorics*, 26(4):P4.19, 2019.
- [28] Denis Zorin and Peter Schröder. A unified framework for primal/dual quadrilateral subdivision schemes. *Computer Aided Geometric Design*, 18(5):429–454, 2001.