# GENERATION OF TOPOLOGICAL INTERLOCKING CONFIGURATIONS FROM A GEOMETRIC APPROACH

A Dissertation

Submitted to the Faculty

of

Purdue University

by

Andres Mauricio Bejarano Posada

In Partial Fulfillment of the

Requirements for the Degree

of

Doctor of Philosophy

May 2020

Purdue University

West Lafayette, Indiana

# THE PURDUE UNIVERSITY GRADUATE SCHOOL
# STATEMENT OF DISSERTATION APPROVAL

Dr. Christoph M. Hoffmann, Chair

Department of Computer Science

Dr. Daniel G. Aliaga

Department of Computer Science

Dr. Voicu S. Popescu

Department of Computer Science

Dr. Thomas Siegmund

Department of Mechanical Engineering

**Approved by:**

Dr. Christopher W. Clifton

Head of the School Graduate Program

To my lovely wife Pauleth,
with whom I started this academic journey,
and who believed on me every single day.

## ACKNOWLEDGMENTS

I want to express my gratitude to my advisor, Prof. Christoph Hoffmann, for his guidance and words of wisdom during my Ph.D. studies at Purdue University. He always found the right questions that forced me to rethink my research approaches and expand the results to more general scenarios. I will never forget two sentences he mentioned during our many weekly meetings: “If Linear Algebra was taught from a geometric point of view, our audience in Computer Graphics would be larger,” and “respect symmetry and geometry will reward you.” I couldn’t agree more.

My gratitude and appreciation also go to the faculty and fellow graduate students of the Computer Graphics and Visualization Lab (CGVLAB) in the Computer Science Department. Our frequent discussions and weekly Graphics Lunches allowed me to learn many more Computer Graphics and Visualization concepts than I thought I would. Also, they challenged me to generalize my research to unconventional geometries. A Topological Interlocking Stanford Bunny was not enough for them. They also want the Möbius Strip, and I am working on it.

And last but not least, I want to thank Prof. Thomas Siegmund and his research students in the Mechanical Engineering Department. They helped me to think about Topological Interlocking Configurations as structures the way they do. And yet, they encouraged me to kept my work on geometric terms. I’m glad they did.

## TABLE OF CONTENTS

Page

Page

# LIST OF TABLES

# LIST OF FIGURES

Figure Page

Figure Page

Figure Page

Figure Page

# ABBREVIATIONS

| | |
|---|---|
| HBM | Height–Bisection Method |
| TAM | Tilting Angle Method |
| TI | Topological Interlocking |
| TIC | Topological Interlocking Configuration |
| TIM | Topological Interlocking Material |
| TOC | Tile Offset Clipping |

# ABSTRACT

Bejarano Posada, Andres Mauricio Ph.D., Purdue University, May 2020. Generation of Topological Interlocking Configurations from a Geometric Approach. Major Professor: Christoph M. Hoffmann.

A Topological Interlocking Configuration (TIC) is an assembly where the shape and alignment of the blocks define the kinematic constraints. Conventional TICs are single-layered structures made of convex blocks. The interface between the blocks in an assembly is face-to-face contact. The traditional convention disregards the use of joinery, adhesive, or other mechanisms that keep two pieces next to each other. However, TICs require a support structure that prevents the lateral strain of the blocks.

The generation process of a TIC starts with a surface tessellation that describes a geometric domain. Each tile in the tessellation represents a traversal section of a block. For regular tessellations and uniform generation parameters, such sections lie in the middle of their respective blocks. Additionally, such conditions guarantee the blocks align adequately with each other. If one of such conditions does not hold, then the resultant blocks may not be aligned. Furthermore, there could be overlapping between the blocks, which makes a TIC invalid.

Traditionally, the generation parameters are angle values set at the edges of the tiles. The angles must match between tiles such that each block in the assembly has a geometry that imposes kinematic constraints to its neighboring blocks. Using the same angle values on regular and semi-regular tessellation produces feasible blocks. That is not the case for non-regular tessellations, curvilinear surfaces, and free-form 3D meshes. In such cases, the generation method must find specific angle values to design the blocks and reduce overlapping.

In this thesis, we propose a TIC generation framework focused on the generation of valid interlocking assemblies based on multiple types of surface tessellations. We start with the Height-Bisection method, a TIC generation approach that uses the distances from a tile to its respective evolution sections as the generation parameters. The method considers the bisector vectors between two tiles to define the parameters that generate aligned blocks to each other. We expand the generation model to a complete pipeline process that finds feasible generation parameters. The pipeline includes clipping parameters and methods in case that overlapping between blocks cannot be avoided.

Additionally, we describe a generalization of the mid-section evolution concept to include multiple evolution steps during the generation process. Our approach considers the angles and distances required to generate infinitely many TICs, including shapes that are not possible using the traditional generation method and the Height–Bisection method. Finally, we consider the interlocking assemblies that cannot maintain static equilibrium due to the shape of the surface tessellation. We consider the Structure Feasibility Analysis method to find the location and magnitude of the minimum tension forces that guarantee a TIC will reach a static equilibrium state. We describe how to update the generation parameters according to the results of the feasibility analysis. Our results show that the proposed pipeline generates valid TICs based on different surface tessellations, including closed and free–form shapes.

# 1 INTRODUCTION

## 1.1 Motivation

Interlocking assemblies made of convex building blocks are of interest in architecture and engineering. Such assemblies represent structures whose building blocks interact with each other through face–to–face interfaces, disregarding the use of joinery, adhesive materials, or excessive support structures. Sketches of barrel vault designs for cathedrals during the Renaissance are the earliest documented examples of convex interlocking assemblies (Figure 1.1).

From a design point of view, convex interlocking assemblies require block shapes that enforce kinematic constraints to the block's neighborhood. That is, each block contributes to locking the blocks adjacent to it; likewise, its neighbors lock it as well. Such a design principle finds its place in modern architecture. Some architects consider convex interlocking assemblies to design floors, walls, vaults, and reciprocal frames based on free–form geometric domains. The current availability of CAD tools–both open–source and commercial licensed software–and 3D printing have made convex assemblies appealing to architects, engineers, researchers, and hobbyists.

From an engineering perspective, convex interlocking assemblies have remarkable properties when considered as a material. As an example, these assemblies proved to be resistant under block failure (e.g., a crack does not propagate between blocks since there is no continuum between them) due to the face–to–face interface mechanism. Furthermore, they dissipate forces better when compared against monolithic structures. Such properties have expanded the interest on convex interlocking assemblies opening the door to new research venues under the names of Topological Interlocking Configurations (TICs) and Topological Interlocking Materials (TIMs).

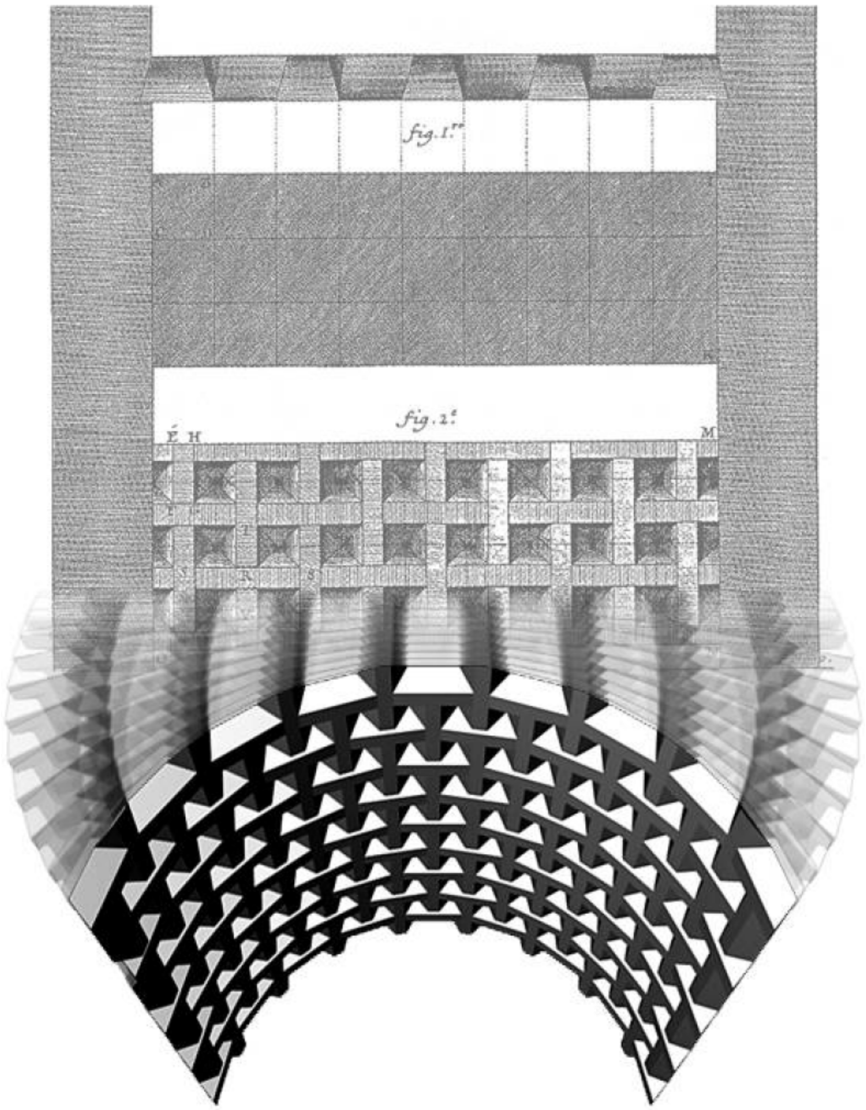


Figure 1.1. Joseph Abeille's ashlar barrel vault. Source: Fallacara [1]

Despite such discoveries, the geometric understanding of these assemblies has remained stagnant. There is a formal explanation of the topological interlocking principle on assemblies made of convex building blocks. Additionally, there is a TIC generation algorithm that uses angles and a surface tessellation to generate an interlocking assembly as input. The algorithm designs a convex block per tile in the tessellation. The output is an assembly made of blocks that maintains the topological interlocking principle. The assembly works fine when the surface tessellation is planar, its tiles are regular polygons, and the angle values are all the same. The familiar shapes of the convex blocks are tetrahedra (for square tiles) and antiprisms (for other tile shapes with an even number of sides).

The generation algorithm returns inconvenient assemblies when at least one of the mentioned conditions (planar tessellation, regular tile shapes, and equal angle values) does not hold. Such inconvenience manifests as either non–alignment or overlapping between the blocks. Furthermore, other polyhedra different than tetrahedra and antiprisms also have topological interlocking properties. Solids as cubes, dodecahedra, icosahedra (and their truncated versions) maintain the topological interlocking principle. However, the generation algorithm cannot design them regardless of the input parameters. Finally, there is the question of whether a TIC can maintain static equilibrium under the assumptions of gravity, static friction between blocks, and block density.

This thesis focuses on such issues: generating TICs based on general surface tessellations, finding a variety of convex interlocking shapes, and guaranteeing a TIC will maintain static equilibrium. Our approach is strictly geometric. We consider the surface tessellation as the scoped shape for the assembly. Each tile in the tessellation is the starting point to generate the respective block. The set of input parameters (e.g., angles or lengths) that apply to each tile guide the block generation process. Only for the static equilibrium analysis, we consider gravity, static friction, and block density since they are required to reason on compression–only structures.

There is an ongoing motivation for formal methods about convex interlocking assemblies based on general surface tessellations. TIMs are in the scope of the automobile and military industries. Additionally, current approaches in materials research consider the shape of the building blocks as an input parameter in their analysis. Up to this day, the shape parameters result from iterative trial–and–error methods. Finally, there are architects interested in generating free–form TICs by differentiating building blocks–consecutive steps that change adjacent block shapes–and maintaining structural stability as much as possible.

## 1.2 Problem Statement

We bring geometric approaches to both the generation process and equilibrium analysis of TICs. Starting with a general surface tessellation, we propose generation methods for TI assemblies based on a variety of convex shapes. Then, we discuss approaches to solve the issue of overlapping blocks in an assembly by finding appropriate clipping parameters. Finally, we use the static equilibrium analysis on an assembly to determine the stability of the structure and adjust the generation parameters if necessary.

This thesis makes the following major contributions towards the design and analysis of TICs based on non–planar surface tessellations:

1. **Height Parameters Set** We present the distance from a tile to its respective evolution sections as an input parameter to generate aligned convex interlocking blocks. We introduce the mathematical framework to use such distances as input parameters in the generation process. Furthermore, we propose two TIC generation methods based on such lengths, each one of them focusing on specific requirements for the resultant blocks of the assembly.

2. **Fixing Overlapping Blocks** We study the scenarios that make a generation method to return an assembly with overlapping between the blocks. Our approach adjusts the generation parameters until the overlapping stops occurring, or it is minimal. The cases with minimal overlapping require clipping the blocks. We consider different alternatives to clip the pieces without compromising the interlocking between the blocks.

3. **Mid–Section Evolution Generalization** We propose an extension of the Mid–Section Evolution concept by introducing parameterized evolution steps. Such generalization explains how to generate infinitely many convex interlocking polyhedra from different surface tessellations. This method may return either convex or concave polyhedra based on the evolution step parameters. The blocks of the assembly maintain the topological interlocking principle.

4. **Configuration Equilibrium Analysis** We use the Measure of Feasibility technique to run the equilibrium analysis of a TIC under the assumption of a compression–only structure (gravity, static friction, and block density). We adjust the generation parameters according to the returned distribution of tension forces in the assembly until the required tension forces are minimum.

## 1.3 Overview

This work proposes frameworks to generate stable TICs based on general surface tessellations. We organize the contributions and related discussions as follows.

Chapter 2 reviews the previous work on TICs from architecture and engineering points of view. We present a brief discussion about the relevance of tetrahedra–based assemblies during the Renaissance; additionally, we mention the groundbreaking research results on TIMs that attracted the attention of researchers in materials engineering.

Chapter 3 introduces the discussion of a TIC generation framework to maintain alignment between blocks. Kanel–Belov *et al.* proposed the first formal generation method in [2], it requires an angle value set on each edge of the tiles from the surface tessellation. The method uses the angles to calculate incident tilted planes on the edges of the tiles. The intersection of these planes defines the geometry of the interlocking blocks in the assembly. We name the approach as Tilting Angle Method (TAM). We consider the challenges of the angle based approach and propose a new generation method using height values set on each tile of the surface. The height values are the distances from a center point in a tile towards the respective top and bottom evolution sections of the block. The method defines the tilted planes as a function of such heights. For non–matching planes on the same edge, we use the bisector plane between them. We name the approach as Height–Bisection Method (HBM). We provide examples of convex interlocking assemblies based on 2D and 3D surface tessellations.

Chapter 4 focuses on the correctness of TICs and piece overlapping removal. A correct TIC must have no overlapping between its pieces. However, overlapping may occur due to infeasible generation parameter values. We show how to search feasible parameter values that remove overlapping or reduce it. There are situations where the remaining option is clipping the blocks of the assembly. For such cases, we describe clipping approaches to make pieces valid without compromising the Topological Interlocking principle. Finally, we propose a new TIC generation method that guarantees correctness by reshaping a subset of blocks in the assembly, avoiding post–processing on the blocks. We name the approach as Shape Fitting Method. We provide analysis and examples of the Shape Fitting Method on developable surfaces, making emphasis on the cylindrical case.

Chapter 5 expands the concept of Mid–Section Evolution into a general scenario. Dyskin *et al.* [3] showed that all Platonic Solids have convex interlocking properties. However, TAM and HBM cannot generate all of them. We propose a generalization of the Mid–Section Evolution concept described by Dyskin *et al.* [4] by introducing parameterized evolution steps. Such steps evolve a tile, the mid–section of a block, in the surface tessellation by reshaping it until it reaches an evolution section. We show how such parameterized evolution steps generate all Platonic Solids and other convex interlocking solids. Additionally, we describe the requirements for evolution step parameters to maintain the convex interlocking principle.

Chapter 6 considers the problem of structural stability on TICs. We use the Structure Feasibility Analysis proposed by Whiting *et al.* [5] to determine whether a TIC is stable under the assumptions of compression–only structures (gravity, static friction between blocks, and block density). Our generation pipeline considers the distribution of required tension forces to readjust the generation parameters of the assembly.

Chapter 7 discusses the conclusions of the thesis and provides an outline for future work on TICs from a geometric approach.

# 2 LITERATURE REVIEW

## 2.1 Introduction

The contributions proposed in this thesis are focused on the geometric insights to generate Topological Interlocking Configurations. In this section, we mainly discuss selected literature that concerns assemblies of convex interlocking shapes from a geometric point of view. Additionally, we review specific results from engineering research that consider interlocking configurations composed of non–regular polyhedra.

## 2.2 Topological Interlocking Configurations

A Topological Interlocking Configuration (TIC) is an assembly of building blocks with interfaces relying on shape and alignment. Such blocks are connected solely by face contact, without the use of adhesives, connectors, or any other binding mechanism to keep pieces together. The structure holds itself by the kinematic constraints imposed by the neighborhood surrounding every block in the configuration and a peripheral support structure. Dyskin *et al.* in [6] coined the term TIC as a “material architecture of regular assemblies of identical interlocked elements” with “[the] possibility of establishing self–locking in assemblies of simple convex–shaped elements free of stress concentrators”. The authors used a planar assembly composed of regular tetrahedra to illustrate the concept.

The kinematic constraints on TICs imply two different considerations while designing an interlocking assembly: each block individually and the entire assembly as a single entity. At the individual level, each block prevents motions on the blocks in its vicinity. That is, each block gets both push and pull motions locked by the wedging actions of the neighboring blocks. With the assembly as a single entity, we require to use a peripheral support frame that guarantees the configuration will not collapse due to gravity. Both planar and curvilinear assemblies require the frame to prevent motions of the blocks at the boundary of the assembly.

Generating a TIC starts with a surface tessellation. A generation method designs an interlocking block for each tile in the tessellation. The method requires generation parameters per tile that guide the design of the respective block. The parameters can be either given or calculated (if the blocks must follow a specification). The design of each block helps to lock the push and pull motions of its neighboring pieces.

The generation method proposed by Kanel–Belov *et al.* requires direction values and tilting angles for all edges of the tiles in a tessellation. The direction values toggle between inside and outside of the tile. Two consecutive edges in a tile must have different direction values. The angle values can be the same if the tessellation is planar, and its tiles are regular polygons. In Chapter 3, we discuss additional details about this generation method and design complications that arise when using non–regular tiles or non–planar surface tessellations.

### 2.2.1 History

The earliest records of interlocking face–to–face contact elements are fragmented lintels used during the middle ages. An example of such systems is a sketch of a cross–joined lintel drawn by Leonardo Da Vinci, shown in Figure 2.1. Two triangular slabs stick together to make a piece, where one triangle is horizontally flipped concerning the other. The sketch appears in Leonardo's Codex Atlanticus f.0091 v [7].

In 1699, Joseph Abeille and Sebastien Truchet proposed their designs for flat vaults made of single shaped blocks. Abeille's proposal uses an ashlar–type block with the shape of an isosceles trapezium. Such block, when repeated in series, generates a flat vault that supports itself when the assembly process finishes. Due to their trapezoidal shape, the blocks leave dimples in the surface of the vault. Later in the same year, Truchet's proposal improved over Abeille's with the design of interlocking shapes

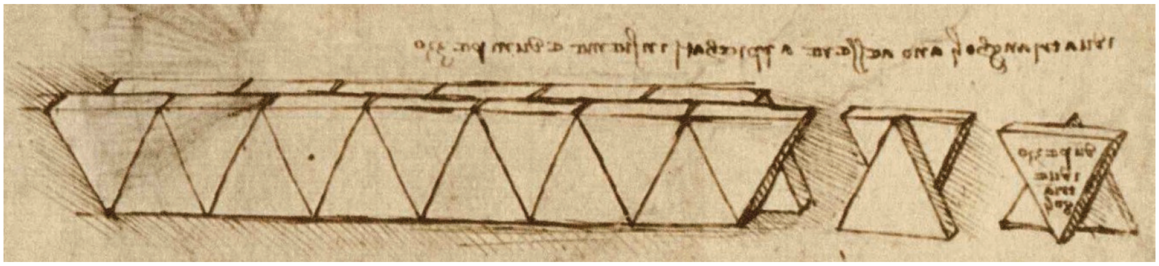


Figure 2.1. Da Vinci's cross–joined lintel. Source: Maurizio Brocato [7]

that leave no void on the surface. Improvements and later developments based on both designs included the removal of surface depressions and the use of different construction materials. Figure 2.2 shows both Abeille's and Truchet's interlocking patterns. The works of Giuseppe Fallacara [1] [8], Brocato and Mondardini [9] [10] and Brocato *et al.* [11] extended the discussion about both Abeille's and Truchet's vaults along with their historical and architectural relevance.

Nearly 300 years later, a design proposal for vertically interlocked paving used the same interlocking principle. In 1984, Michael Glickman argued that traditional interlocking paving blocks were large, heavy, prone to damage due to connections, and challenging to install. He proposed a block that is easy to manufacture, convenient to transport, and unlikely of accidental damage while handling [12]. The block is a truncated tetrahedron posed over one of the original tetrahedron edges. Its central section must be at equal distance from both top and bottom faces obtained after truncation. Additional spacer ribs embedded in the block improve the load equalization.

Early in the 21st century, a third independent approach considered the very same principle applied to the design of materials and structures. In 2001, Dyskin *et al.* [6] considered the concept of Key Blocks while discussing the kinematic constraints among pieces and how to keep blocks in place. The authors concluded that an assembly of identical tetrahedra, as shown in Figure 2.3, contains self–interlocking. That is, the blocks in a column tetrahedra act as key blocks to the respective adjacent

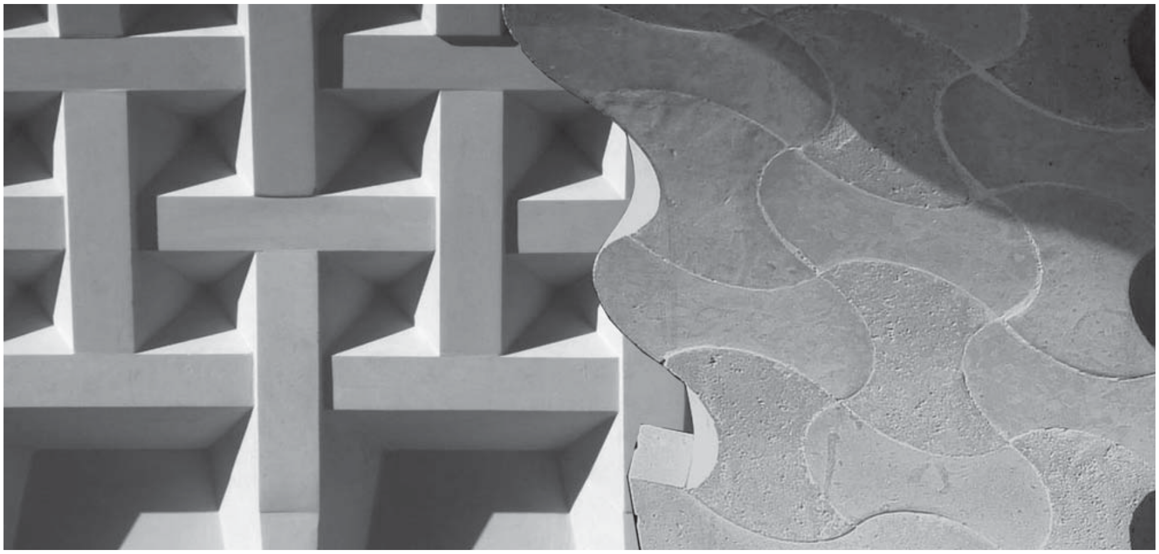


Figure 2.2. Abeille and Truchet vaulted patterns. Source: Giuseppe Fallacara [8]

block columns. Such assemblies require peripheral constraint structures to prevent lateral strains on the blocks at the boundary.

The authors found relevant mechanical attributes of such assemblies. Dyskin *et al.* [4] [13] mentioned that blocks in a planar tetrahedra–based TIC are hardly deformed by concentrated force loading. Furthermore, the bending compliance of such assembly is lower than that of a plate of the same thickness. Regarding block failures, Dyskin *et al.* in [14] discussed that cracks on a failed block retard to propagate to the neighboring blocks. Since blocks are weakly connected, the cracks lack a medium to propagate through the assembly. The authors found that planar tetrahedra–based TICs can tolerate up to 59% random block failure.

The initial analysis on TICs considered assemblies made of tetrahedra–like blocks. Dyskin *et al.* [4] expanded the principle of self–interlocking tetrahedra using both octagon–based and circle–based shapes for the construction of planar assemblies. Their analysis acknowledged the existence of a variety of unknown Topological In-

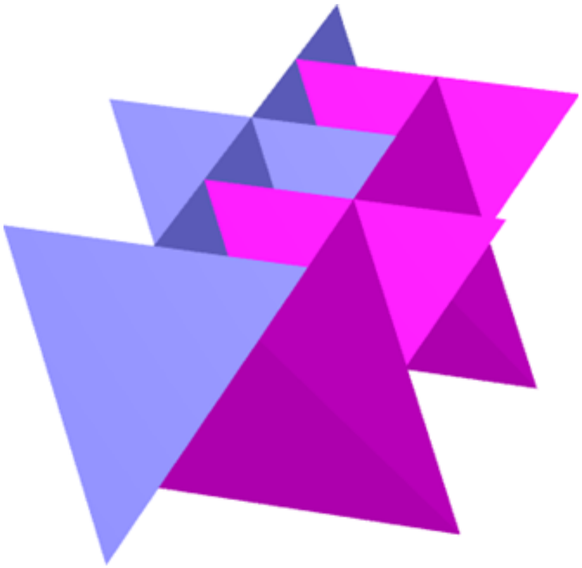


Figure 2.3. A general view of self–interlocking tetrahedra. Two column tetrahedra sections are displayed with different colors.

terlocking shapes. It was a matter of time until researchers found more interlocking shapes and formalized the Topological Interlocking principle.

### 2.2.2 Topological Interlocking Principle

The basic premise for convex interlocking is that an arrangement of certain aligned convex polyhedra results in a self–interlocking assembly. Kanel–Belov *et al.* [2] specified the foundations that describe both generation and interlocking criterion for a layer of convex polyhedra.

Generating a TIC starts with a surface tessellation that describes the overall shape for the assembly. The edges of each tile in the tessellation are assigned with direction values pointing inwards and outwards the tile. The directions from two consecutive edges must point differently. Then, consider an incident plane for each edge of a tile. The plane inclines an angle $\theta$ in the direction assigned to the respective edge. The intersection points between the planes from three consecutive edges in a tile define the vertices of a polyhedron. Such polyhedron is the interlocking block associated with the respective tile from the tessellation. The geometry of a tile represents the intersection polygon between the corresponding block and a secant plane. Translating the secant plane along the normal vector of the tile results in a reshaped polygon. These reshaped polygons describe the evolution of the tile. Section 3.3 describes the algorithmic representation of the generation process.

The shape of each tile determines the overall shape of the corresponding blocks. The generation method designs a tetrahedron from a square tile, and a cube octahedron or cube using a hexagon tile. For the cube, we need to include the intersection points between the planes tilted in the same direction within the tile. The blocks will have regular shape provided a proper tilting angle $\theta$ for each case (see appendix). Furthermore, the tiles represent the mid–sections of their corresponding blocks. The evolution along both directions of the normal vector of a tile describes both halves of the respective polyhedron.

The Interlocking Criterion states that "A layer of solids is a set with interlocking if and only if the following condition holds: For every polyhedron, the corresponding [tile] vanishes during the evolutionary transformation". The proof of this principle is the evolution of the tile bounded by the planes containing the faces of the corresponding polyhedron. The tile collapses into a single point or a line segment if and only if the planes contain a finite region.

### 2.2.3 Interlocking Geometries

Many polyhedra comply with the Topological Interlocking principle. Platonic Solids, certain Archimedean Solids, and antiprisms are part of the list. Additionally, some non–convex shapes (mostly based on convex counterparts) also comply with the interlocking principle by having face–to–face contact only (i.e., no joinery or connectors) in an assembly.

### Platonic Solids

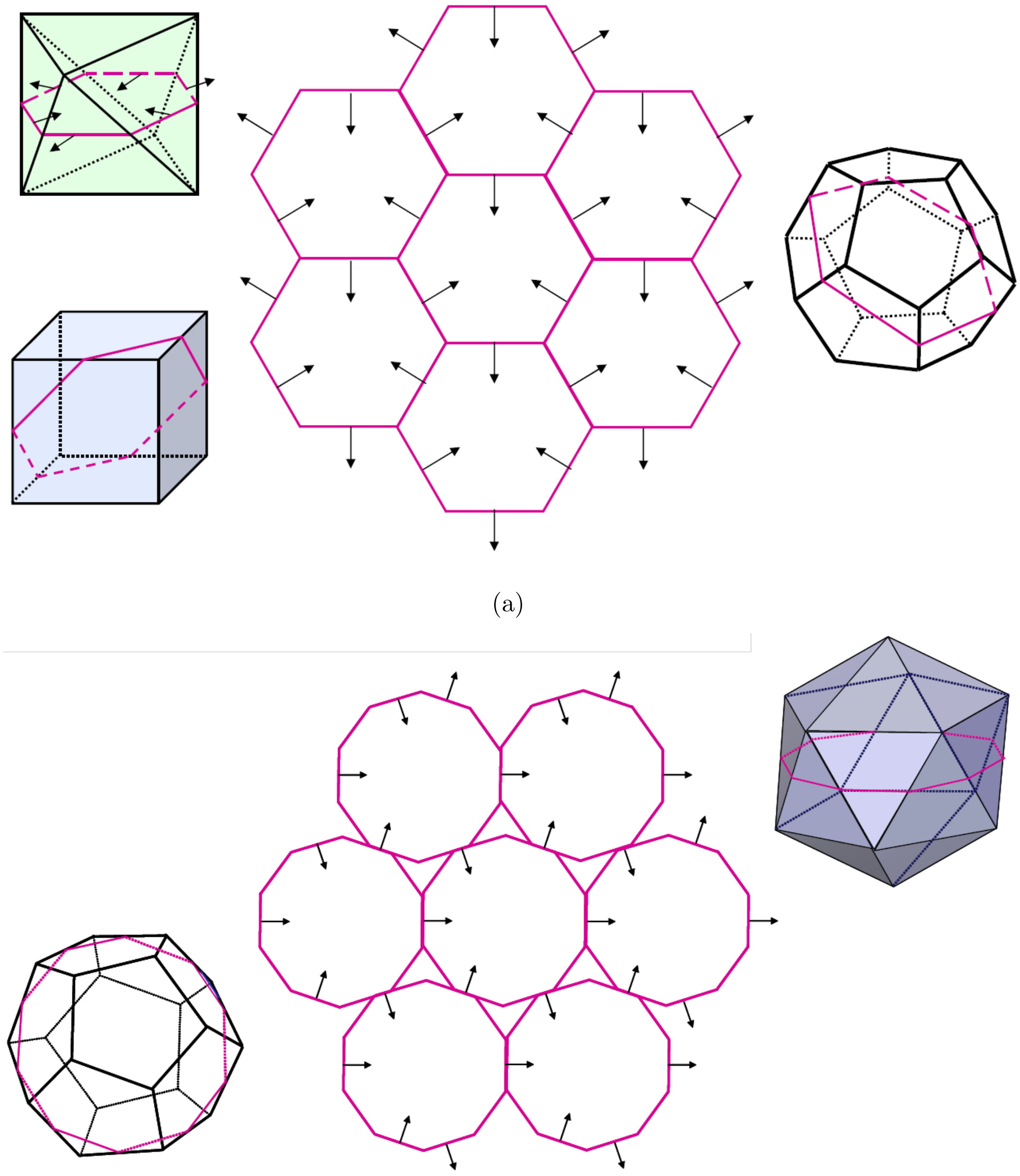
Dyskin *et al.* [3] reported that the Platonic solids can maintain Topological Interlocking when assembled in planar sections. Each solid results from the evolution of specific tile shapes. A square evolves into a tetrahedron. A hexagon evolves into an octahedron, cube, or dodecahedron. A decagon evolves into a dodecahedron or an icosahedron. A surface tessellation made of decagons requires additional kite–like tiles to fill the gaps. In such a case, the generation method must ignore the kite–like tiles to avoid unexpected results. Figure 2.4 shows the poses for the solids to fit on the hexagon and decagon tiles.

The truncated versions of the Platonic Solids also have interlocking properties. Dyskin *et al.* in [3] argued that "certain truncations of [the Platonic] solids leave the interlocking property unaffected". Examples of interlocking assemblies made of truncated tetrahedra are given by Glickman [12] and Dyskin et al. [6]. Furthermore, Dyskin *et al.* in [15] described three different arrangements for an assembly of buckyballs (i.e., truncated icosahedra) to be in interlocking (associated with the symmetries of fifth and third order). Figure 2.5 shows such arrangements.

### Nexorades

Brocato and Mondardini [9] [10] considered the analysis and generation of spherical vaults following the Abeille's design. Their approach focused on the optimal parameters for the stereotomy of the blocks. Additionally, the authors considered a finite model analysis over the final assembly for mechanical purposes.

The name of the resulting assemblies is nexorades. A nexorade is a "structure made of nexors, a beam often having four simple connections, two at its ends to be supported and two at intermediate points to bear other nexors" [10]. A classical nexor resembles a truncated irregular tetrahedron. Nexors generated from regular tetrahedra are notoriously similar to the block shapes reported by Glickman and Dyskin. The principle behind nexorades is similar to Topological Interlocking. However, the

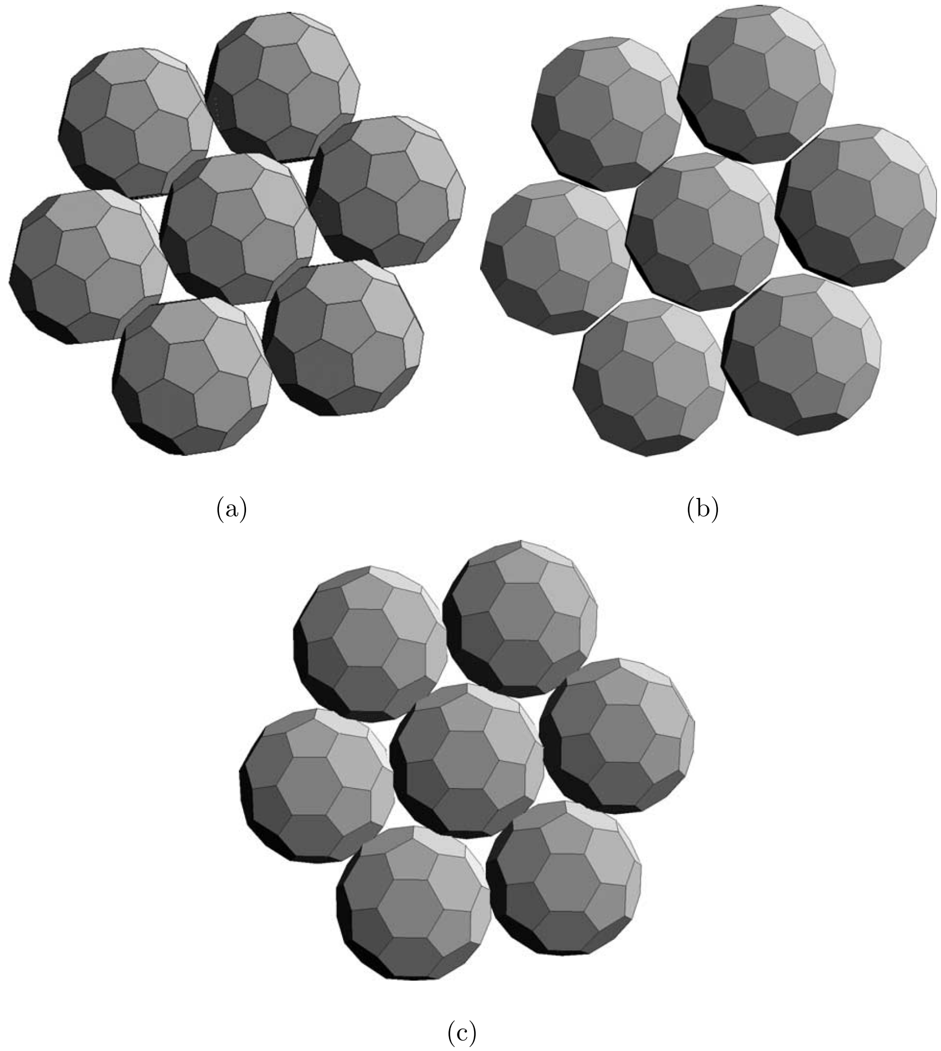


(a)

(b)

Figure 2.4. a) Hexagonal tiling with associated Platonic solids, b) Decagonal tiling with associated Platonic solids. Source: Kanel–Belov *et al.* [2]

nexors require joinery to keep sections together due to the spherical geometry of the vault. Mauricio Brocato in [7] proposes a mathematical model that describes TICs



(a)

(b)

(c)

Figure 2.5. Assemblies of interlocking buckyballs. a) Associated with fifth order symmetry axes, b) Associated with third order symmetry axes and pentagonal face contacts, c) Associated with third order symmetry axes and hexagonal face contacts. Source: Dyskin *et al.* [15].

as a continuum of structural systems made of interlocking ashlars (masonry blocks cut from larger individual stones).

Semi–Regular and Non–Regular TICs

Recent advances concerning geometrical approaches come from computational architecture. Weizmann *et al.* [16] used TICs to build facades, prioritizing their work on extending the catalog of resultant convex interlocking assemblies based on different types of tessellations. They concluded there is a connection between the geometry of the blocks and the structural performance of the assembly.

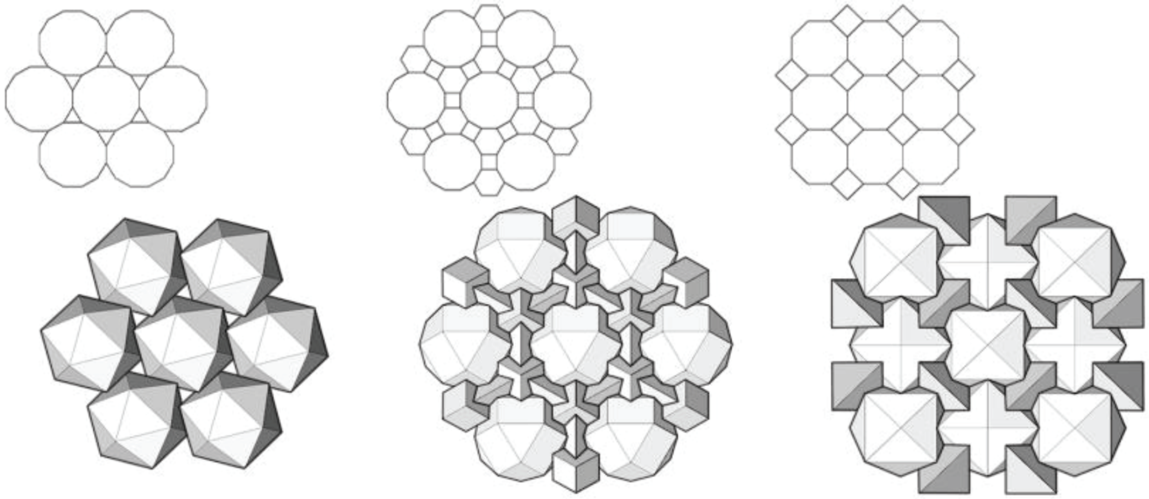
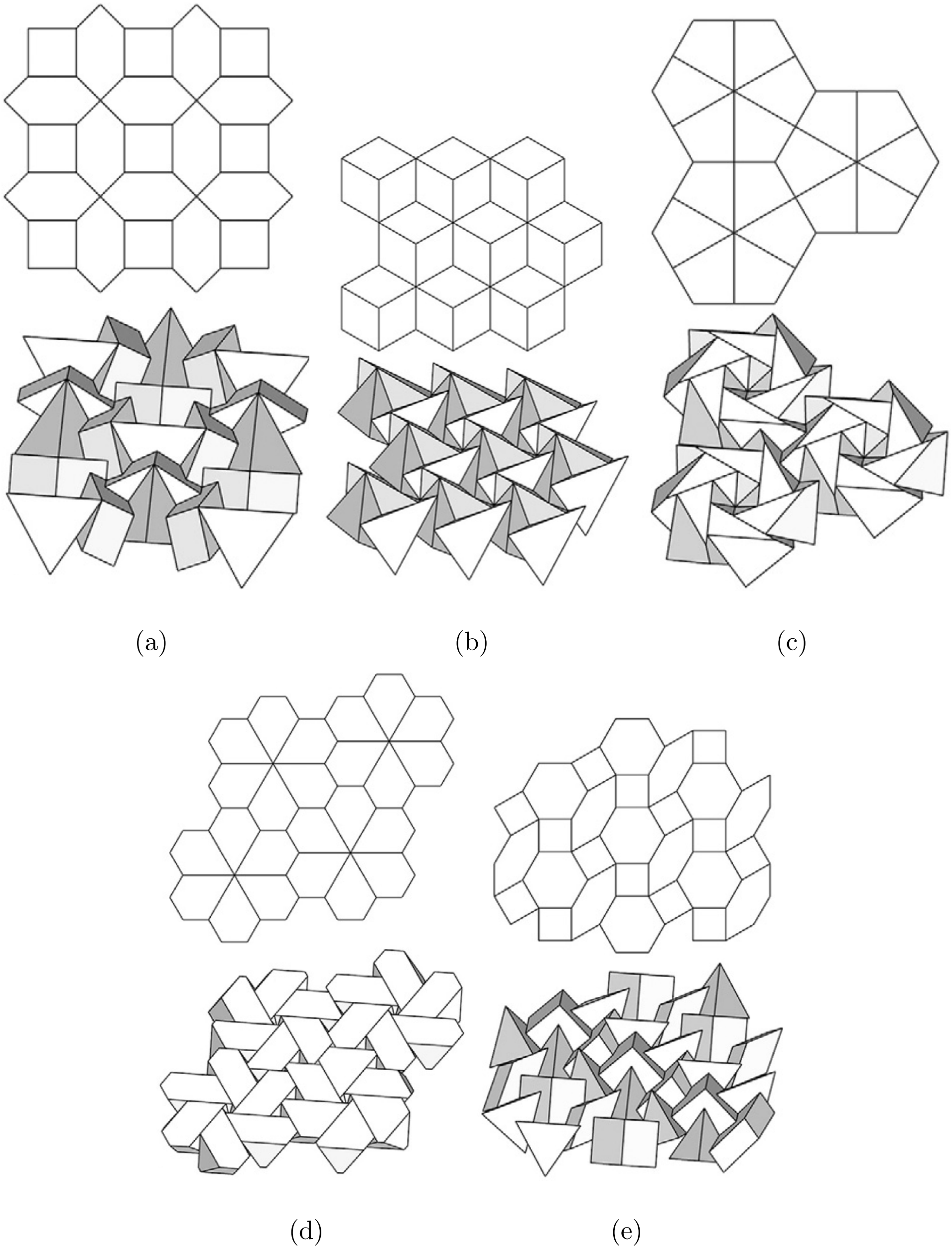


Figure 2.6. Semi–regular tessellations and respective TICs. Source: Weizmann *et al.* [16]

Their work in [17] continued the search of an expanded catalog of TICs. They conclude that, from all semi–regular tessellations, three are appropriate for TI purposes (shown in Figure 2.6). Furthermore, they state there is no limit for TICs based on non–regular tessellations since their number is infinite. Figure 2.7 shows examples of non–regular tessellations and their respective TICs. The authors also considered the support structure required to assemble floors made of convex interlocking blocks. The results were subject to structural simulation for load carrying capacity and deflection analysis. Finally, in [18], they introduced a computational method to generate planar surface tessellations aimed for TICs generation. Their solution is a parametric


(a) (b) (c)

(d) (e)

Figure 2.7. Examples of non–regular tessellations and their respective TICs. Source: Weizmann *et al.* [16]

approach that generates surface tessellations by indicating the number of edges for the tiles and the angle values between the edges. Their system generates a TIC using a resultant surface tessellation.

TICs based on Non–Planar Surface Tessellations

The design of TICs based on non–planar surfaces has been of interest to researchers in the topic. Vella and Kotnik [19] considered how curvature in the geometric domain affects the resultant pieces of a non–planar Abeille–based TIC. They observed the interdependency between curvature in the geometric domain and the resultant assembly. They found that both curvature and tilting angles are inversely proportional to each other. Additionally, curvature and piece vertices distribution are directly proportional (resulting in smaller pieces that approximate the steep curvature).

Weizmann *et al.* [17] adapted the generation method formulated by Kanel–Belov *et al.* in [2] to work on curvilinear tessellations. Their implementation includes an additional step that adapts the pieces to the curvature of the surface. In such a case, the assembly of the blocks must follow a specific order.

Tessmann and Rossi [20] [21] described approaches to design interlocking pieces based on parametric design logic and discrete combinatorial processes. The former returns the geometry of each block according to their location in the tessellation. The latter focuses on exploiting the combinatorial capabilities of repeating block shapes in the assembly. They applied such approaches to generate TICs bounded by two NURBS surfaces. The boundary information helps to generate distorted, trimmed tetrahedra that align with the curvature of the surfaces. An example of such configurations is shown in Figure 2.8.

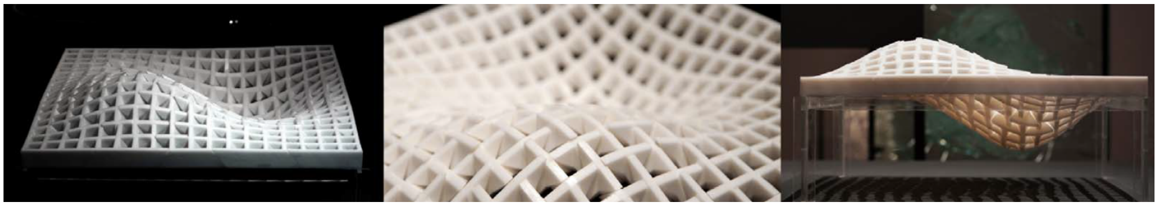


Figure 2.8. TIC with 3D printed mass customized elements. Source: Tessmann and Rossi [21]

### Non–Convex Geometries

The Topological Interlocking principle also holds for concave blocks provided that two blocks interact with each other through a face–to–face interface only. Truchet's design is an example of a TIC based on concave blocks.

An example of a concave Topological Interlocking shape is the Osteomorphic Block. Estrin *et al.* in [22] [23] introduced the block for Topological Interlocking purposes. An Osteomorphic Block has a shape that prevents both push and pull motions simultaneously through a single face. This approach differs from convex polyhedra, which faces can only lock one motion at a time. At least one face of the Osteomorphic Block is concave. Its shape comes after a sufficiently smooth surface that minimizes stress concentrations. The Osteomorphic Block can take different shapes according to its location in the assembly. Figure 2.9 shows examples of such blocks and a resultant interlocking assembly.

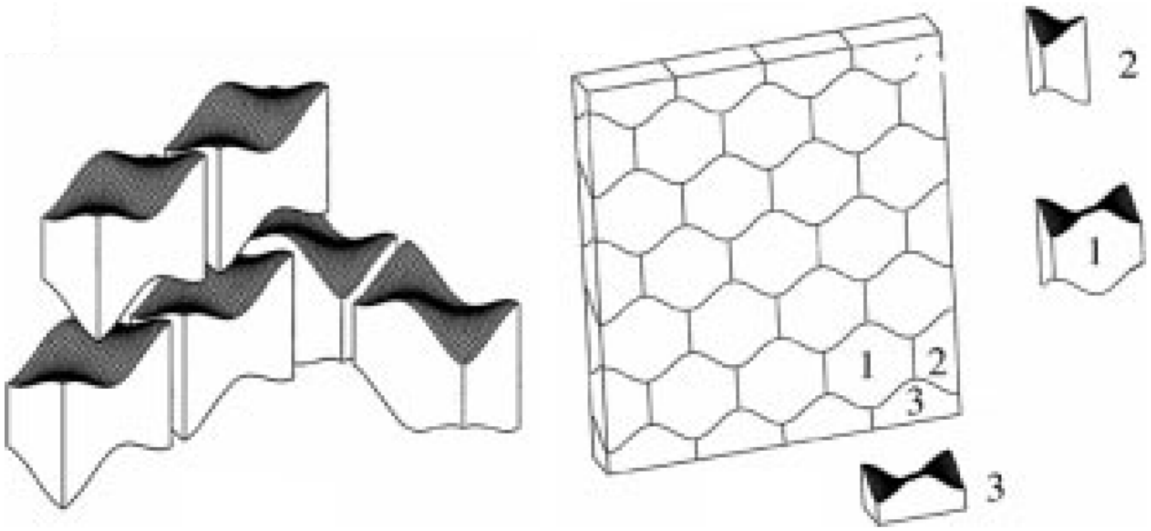


Figure 2.9. Osteomorphic Blocks. Shape 1 is a complete block, shapes 2 and 3 are half–blocks. Source: Dyskin *et al.* in [24]

Dyskin *et al.* in [24] discussed the properties of TICs based on Osteomorphic Blocks. In terms of geometry, their shape allows the design of assemblies with corner structures. Additionally, the different shapes of the Osteomorphic Brick allow the definition of edges in the assembly. Such benefits come from the periodicity and symmetry attributes of the surface that describe the interacting faces of the block. The dynamics of the assembly made of Osteomorphic Blocks are similar to their convex–based counterparts.

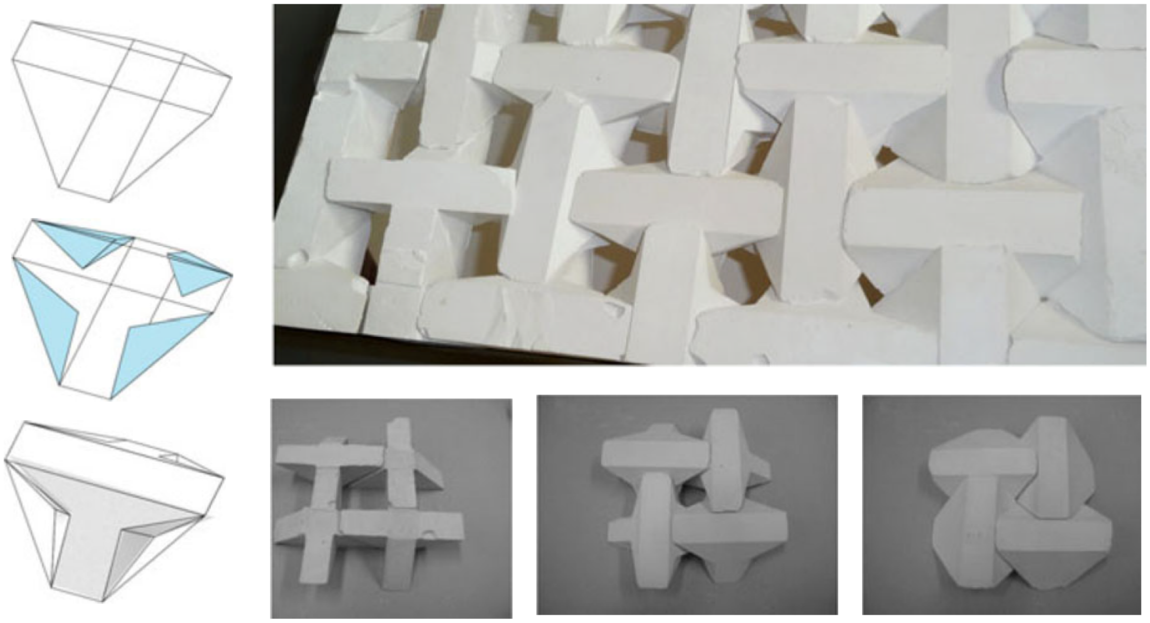


Figure 2.10. Non–Convex tetrahedron–like piece and resultant interlocking assembly. Source: Tessmann [25]

Other concave interlocking shapes follow a more polygonal style. Tessmann [26] [25] and Tessmann and Becker [27] reported the result of student projects who built TICs as geometrical differentiated, reversible, force–locked systems. The solutions presented by the students considered a variety of shapes derived from tetrahedra. Although not convex, the planar faces preserve the fundamental notion of the Topological Interlocking principle. Figure 2.10 shows an example of such solutions. Another proposal considers the design of a valid border for a finite assembly. The pieces at the boundary have a special shape that enables the assembly of a frame. Such a solution involved windmill–shapes to constrain the boundary, allowing the configuration to end on edge without additional peripheral structure. Figure 2.11 shows the boundary and the assembly it holds.

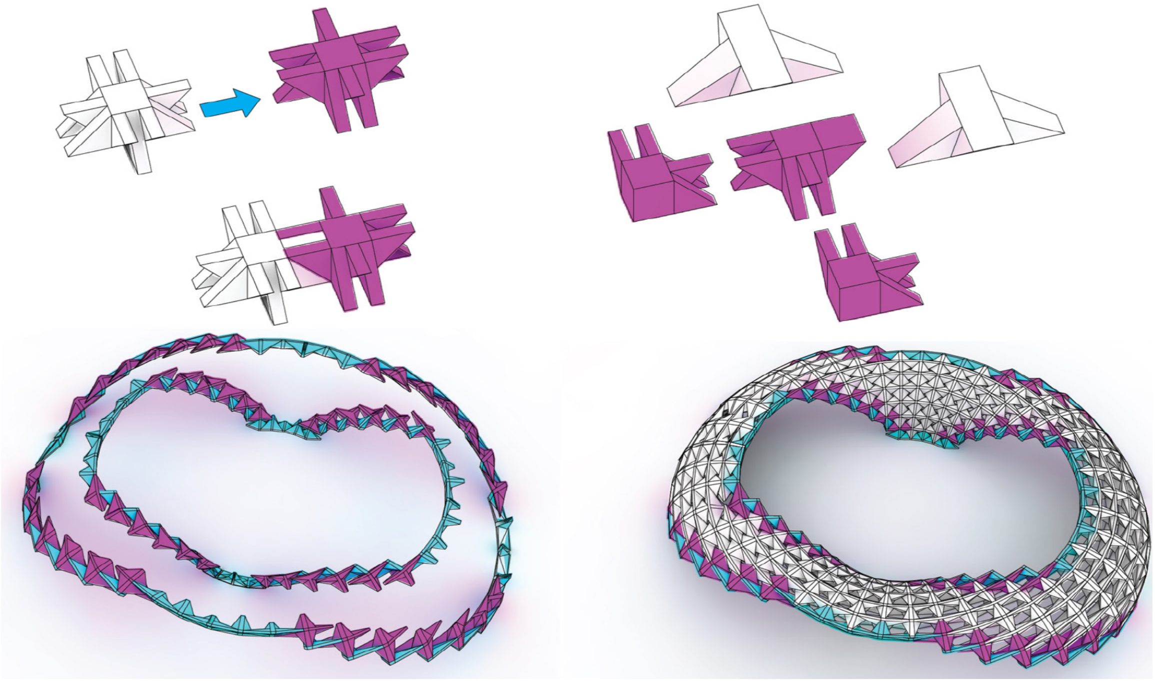


Figure 2.11. Boundary made of concave Topological Interlocking blocks. Source: Tessmann [26]

## 2.3 Topological Interlocking Materials

The term Topological Interlocking Material (TIM) refers to the study of TICs when considered as a non–monolithic material. That is the mechanical attributes of assemblies composed of multiple pieces arranged as a layer. Materials researchers have studied the mechanics of TIMs by considering the relations between physical attributes (e.g., density, friction, material composition) and observed reactions on the pieces (e.g., displacement, stress, fatigue, percolation) under the presence of loads. The number of such studies is larger than the studies based solely on geometric insight. Nevertheless, geometry plays a vital role in such studies. The shape of the blocks in the assembly is an input variable to the material analysis.

In the following subsections we describe recent and relevant work on TIMs, in some cases making emphasis on the relevance of the geometry of the blocks for solving the problem. For further information on the matter, we recommend the extensive reviews by Siegmund *et al.* [28] and Dyskin *et al.* [29] which describe the research progress of this field.

### 2.3.1 Remanufacturing

A relevant attribute of TIMs is remanufacturing. That is, the possibility to replace failed blocks without compromising the overall structure. Mather *et al.* [30] considered the performance effects of remanufacturing a TIMs after failure. They described three different approaches: 1) Replacing the failed blocks using new ones, then shuffling the blocks (both remaining and new), and reassembly the TIM by picking blocks randomly. 2) Replacing the failed blocks using new ones, then leaving the remaining blocks at their original position. And 3) Switch a failed block with a good one, the location for the failed blocks is at positions perceived as non–critical.

The results indicated that remanufacturing does not affect the mechanical performance of the assembly. However, a repeated remanufacturing reduces the magnitude of the stiffness, ultimate load, and energy absorption of the TIM. Even the performance of a remanufactured TIM using the third approach does not get compromised significantly. The possibility to reuse failed blocks makes TIMs attractive for environmental and sustainable purposes.

### 2.3.2 Strength, Toughness, and Stiffness

In conventional materials, strength and toughness correlate negatively. However, experimental data indicate that such correlation for TIMs is linear and positive. Khandelwal *et al.* in [31] discuss that the combination of segmentation (i.e., assemblies made of unit segments) and cellularity (i.e., number and type of cells) of TIMs would allow the design of materials with attractive properties. The authors analyzed the mechanical properties of TIMs based on tetrahedra (using different block densities), including force–displacement and material deformation. The experimental results show that the force–displacement response exhibits an ideal softening response (i.e., reduced stress with increased deformation). The thrust line and finite element analysis of the TIMs predicted the obtained experimental data. The results make TIMs attractive since they manifest such properties even though discrete elements form the material.

The deformation of a TIM in response to an applied load is another research topic on TIMs. Khandelwal *et al.* [32] considered the active and adaptive external constraints to guarantee variable stiffness and energy absorption characteristics in TIMs. They found that TIMs are good for energy absorption while keeping low–stress levels. The experimental data indicate that the assemblies can dissipate energy at varying force and stiffness levels. Furthermore, the authors found that failed blocks prevent the increase of transmitted forces. This attribute becomes relevant for applications where the material must not allow forces beyond a critical value.

### 2.3.3 Energy Absorption

TIMs can absorb more impact energy than solid counterparts. Feng *et al.* in [33] studied the energy absorption capabilities of assemblies based on tetrahedra blocks. The authors simulated TIMs impacted by projectiles with different velocities. The energy absorption reaction depended on the velocity of the projectiles. The response to lower velocities was slow deformation of the TIM. Larger velocities resulted in fragmented blocks due to the impact. However, the segmented nature of the TIMs prevented the dissipation of the energy through the assembly. The fragmented blocks prevented the failure to propagate beyond the region of impact. Such results indicate that TIMs are more damage tolerant than monolithic systems.

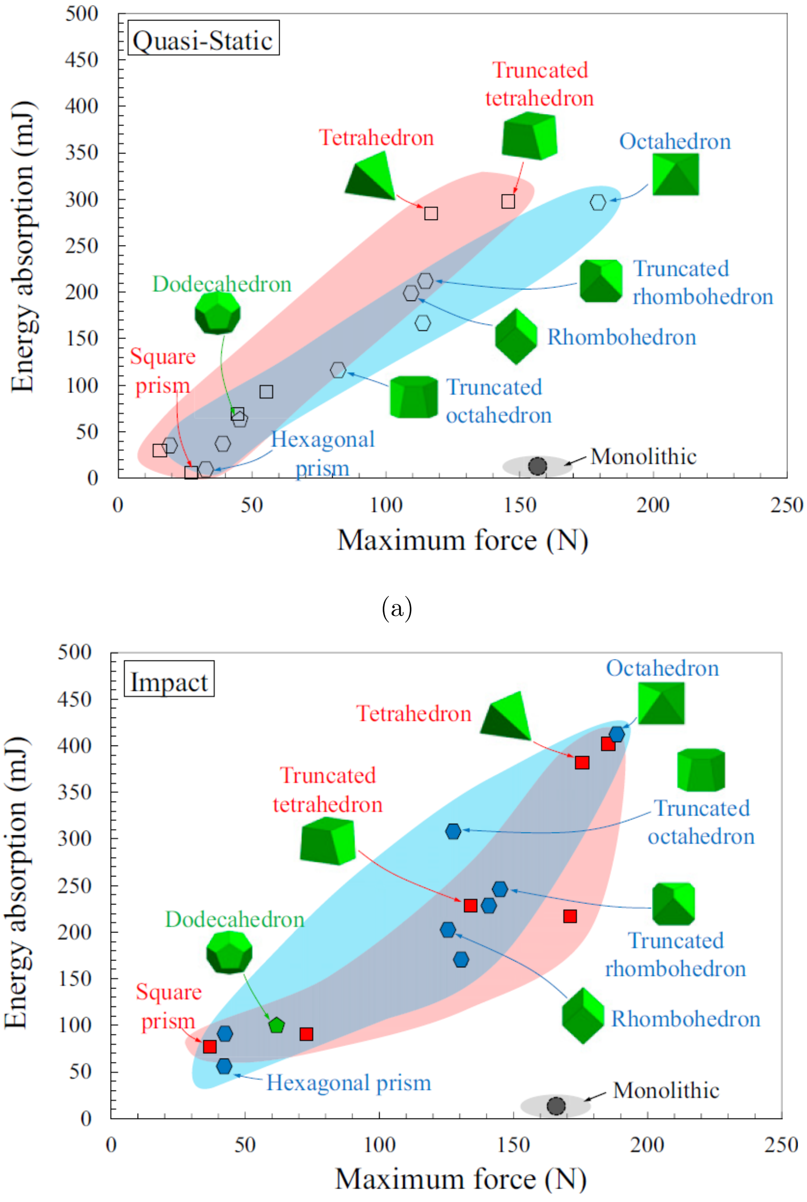
### 2.3.4 Geometry and Assembly Performance

The shape of the blocks has implication on the performance of the interlocking assembly. Mirkhalaf *et al.* [34] [35] considered the shapes of the interlocking pieces in their study of strength and toughness of TIMs. Multiple configurations designs (based on square and hexagon tiles) were 3D printed and tested. Authors found there is a positive effect of increasing the values of the tilting angles on the energy absorption of the TIM. Figure 2.12 shows the energy absorption vs strength results according to the shape of the pieces of the configurations.

Weizmann *et al.* [36] considered the relation between the geometry of the blocks and the structural performance of the assembly. They considered a variety of planar TICs based on different interlocking geometries. All configurations were subject to indentation tests to analyze the displacement of the pieces after applying loads. Their results show that a higher number of tiles and block faces with lower tilting angles reduce the structures strength. Figure 2.13 shows four different TICs and their respective displacements after applying loads.

## 2.4 Summary

The study on TICs have been done from different types of fields and perspectives. The reviewed literature regards TICs as an architecture design principle, a formal interlocking concept, and as a non–monolithic material. The evolution of the concept through time shows its usefulness at multiple physical scales. The discoveries on TICs from the last two decades have expanded our understanding of the concept. There is an ongoing motivation to generate non–planar assemblies and find the benefits they

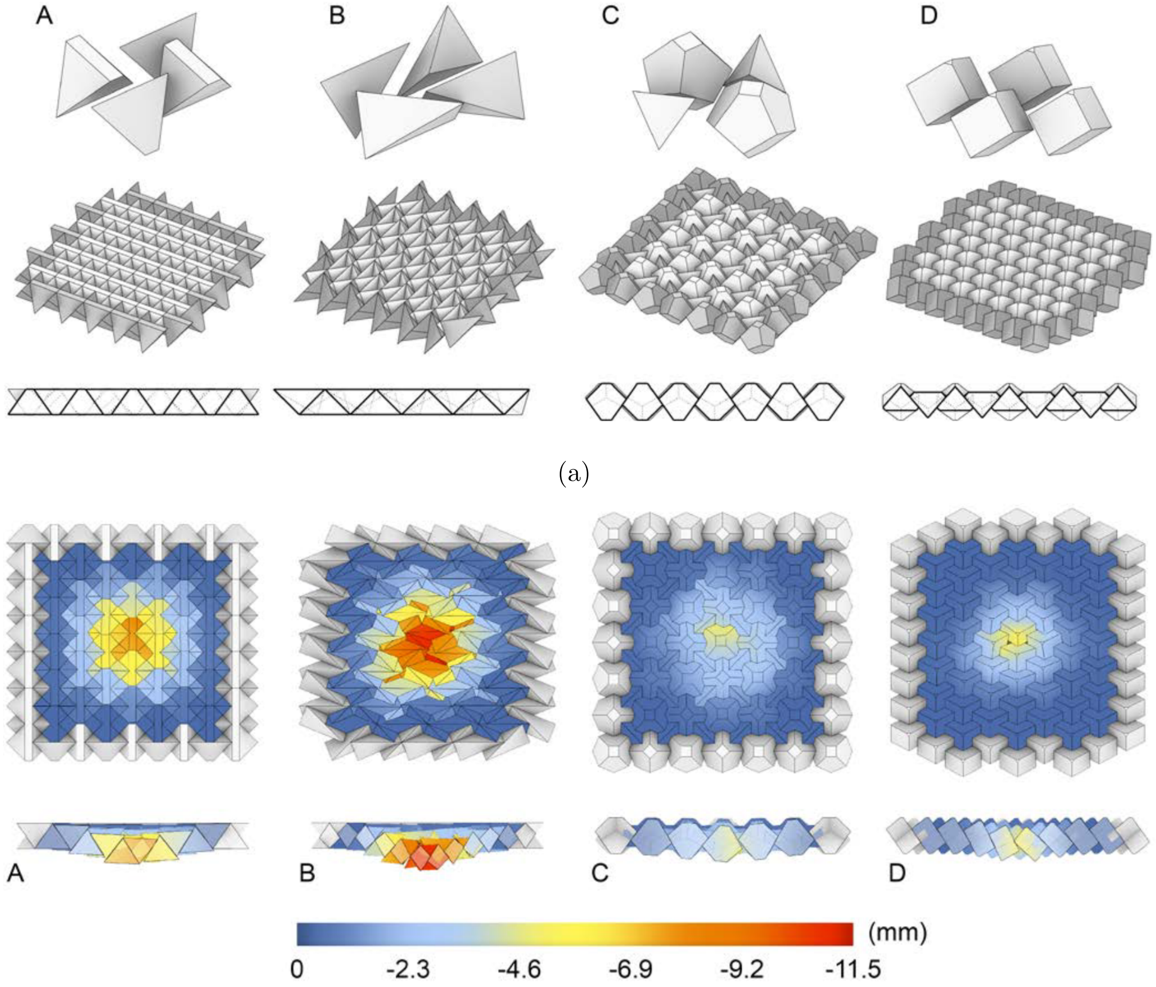


(a)

(b)

Figure 2.12. Energy absorption of TIMs based on piece shapes. a) Quasi–static conditions, b) Impact conditions. Source: Mirkhalaf *et al.* [34]

may offer. There are multiple challenges to be addressed in terms of generation and purpose. The former relates to finding interlocking geometries that adapt to non–



(a)

(b)

Figure 2.13. TICs and vertical displacement. A) trimmed tetrahedra, B) skewed tetrahedra, C) polyhedra based on octagons and squares, D) Cubes. Source: Weizmann *et al.* [36]

planar surfaces and still can maintain a stable state. The latter focuses on mechanical attributes while studying TICs in physical settings.

The reviewed geometric approaches find the blocks by defining tilting angles and adapting the geometries such that they align with the curvature of the surface tessellation. The latter process requires manual intervention or the use of CAD tools. The reshaping criteria (e.g., shape distortion and trimming) is left as a post–processing step to guarantee feasible configurations in real settings. There is still a need for a unified generation framework to design TICs based on any surface tessellation (either planar, curvilinear, or free–form). The following chapters describe our contribution and rationale to reach such a unified framework based on a geometric approach.

# 3 PIECE GENERATION USING HEIGHT PARAMETERS

A version of this chapter has been previously published in the International Journal of Architectural Computing: Bejarano, A., & Hoffmann, C. (2019). A generalized framework for designing topological interlocking configurations. International Journal of Architectural Computing, 17(1), 5373. https://doi.org/10.1177/1478077119827187

## 3.1 Introduction

The fundamental elements in a TIC are the pieces. Each piece must guarantee its interlocking and contribute to its neighbors as well. Kanel–Belov *et al.* in [2] introduced the formal definition and development of convex interlocking. Maurizio Brocato in [7] proposed an analysis model of TICs when considered as a continuum interlocking structural system. Generating a TIC requires a geometric domain and a set of parameters that define the shape of the pieces. For regular domains (i.e., regular planar tessellations), a single parameter value is sufficient to determine the shape of the pieces. More general domains require additional parameter estimation to guarantee valid piece shapes. Determining the right input parameters can be a tedious process, in particular when the geometric domain misses symmetry and regularity attributes.

We introduce a TIC design method based on height parameters that guarantee valid TICs on a single iteration over the geometric domain. The contribution starts with an algorithmic formulation of the problem to vision the shape generation process from a computational approach. Then, we state the traditional generation method proposed by Kanel–Belov *et al.* in [2] using such algorithmic formulation. Finally, we describe our generation method based on height parameters. We consider the resultant TICs from both generation methods when using regular and non–regular geometric domains.

## 3.2 Overview

The TIC generation methods solve a design problem: generate a convex interlocking assembly given a geometric domain (represented as a surface tessellation or a 3D mesh) and a set of input parameters. The pieces of the assembly are shaped by the evolution of the tiles (i.e., faces) until they reach both extreme locations of their respective evolution sections. We formulate a generation method that uses a height parameter and a center point per face. Both parameters define the extremes of the respective evolution sections. For adjacent faces, the parameter of the incident edge is the bisector between the planes of interest that pass through it. This chapter first reviews the TIC generation method based on angle values on each edge. Next, we introduce the height–based parameter set and the algorithm to generate the pieces. Such contributions are the foundation of other generation methods discussed in Chapters 4 and 5.

## 3.3 Background: Tilting Angle Method

We name Tilting Angle Method (TAM) to the TIC generation approach described by Kanel–Belov *et al.* in [2]. The method works on even–sided polygonal tiles with alternating direction values on the edges of the faces. Through every edge, there is a plane $P$, initially orthogonal to the tessellation plane. $P$ is tilted out of the perpendicular by a tilting angle $\theta$, in the indicated direction value. The intersection of the tilted planes through the edges of each tile defines the geometry of a polyhedron.

### 3.3.1 Edge Directions and Tilting Angles

We explain the steps of the TAM using a checkerboard tessellation as an example. The method starts by assigning an arrow (indicating the tilting direction and value) to each edge. The color of each tile determines the direction of the arrows. Dark tiles have both north and south arrows pointing outward the tile while both east and west arrows point inward the tile. Similarly, light tiles have both north and south arrows pointing inward the tile while both east and west arrows point outward the tile. All arrows must comply with the following rules: they have the same length, they start on the midpoint of the respective edge, and they are perpendicular to the edge in the tessellation plane. The arrows indicate the chosen tilting angle by their length and the tilting direction by their direction (Figure 3.1).

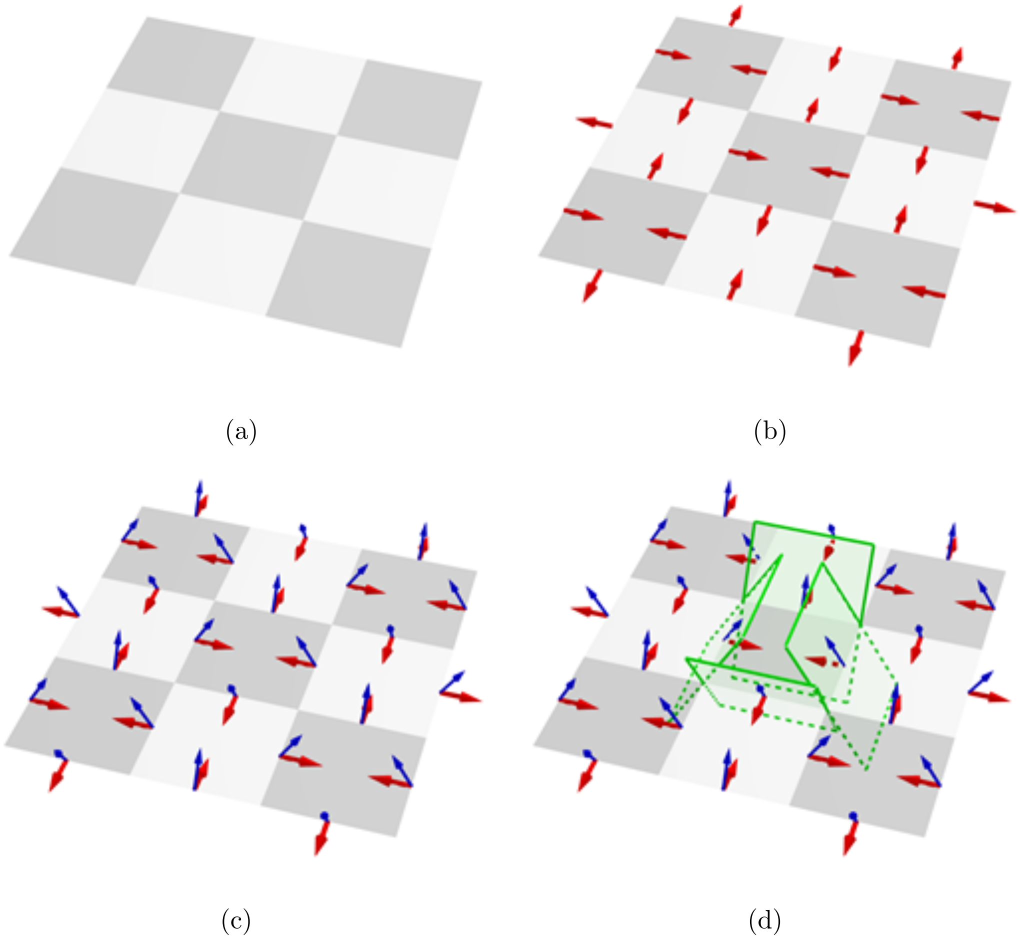
Kanel–Belov *et al.* [2] generalized the tilting method for plane tessellations composed of even–sided tiles. The method sets the direction arrows such that inward and outward directions alternate along the edges of every tile. Each edge gets a tiling angle value and a tilted plane, as described for the checkerboard case. The intersection of the tilted planes incident to a tile generates a polyhedron as before. Figure 2.4(a) shows examples of the arrow setup applied to a hexagonal tessellation.

### 3.3.2 Piece Design

We describe the TAM using an algorithmic formulation. Let

$$\mathrm{DCEL}(M) = \{\mathrm{vertices}, \mathrm{faces}, \mathrm{halfedges}\}$$

be the Doubly Connected Edge List representation of the mesh $M = \{V, F\}$ containing the information of the geometric domain. Let $d \in \{-1, 1\}$ be the initial direction value to be used. Traverse through each face $f \in$ faces and set the alternating direction values on the half edges of the respective face, such directions are set as: halfedges[$h$].direction $= d$ and halfedges[$h$].twin.direction $= d$. Alternate $d$ and $d$ each time a direction value is set on a half edge of the face. For



(a)

(b)

(c)

(d)

Figure 3.1. Steps of the TAM on a checkerboard: (a) Initial checkerboard, (b) arrow setup (red), (c) Tilted direction arrows (blue), (d) Incident tilted planes for the middle tile (green).

every half edge, define halfedges$[h]$.midpoint $= h_m$ and halfedges$[h]$.twin.midpoint $=$ halfedges$[h]$.midpointwhere $h_m$ is the midpoint of the edge. Let $C_f = \text{center}(\text{faces}[f])$ be a center point of the current face. The direction arrow on each edge is described by the unit vector halfedges$[h]$.vector $= \frac{PC_f}{\|PC_f\|}$ where $P$ is a point along halfedges$[h]$ such that $PC_f$ and the direction vector of the half edge are orthogonal. Since the direction vector is associated with an edge, it is valid to set halfedges$[h]$.twin.vector $=$ halfedges$[h]$.vector.

The vectors associated with each half edge are rotated with respect to the direction vector of the respective half edge. Then:

$$\text{halfedges}[h].\text{tilted} = \text{rotate}(\text{halfedges}[h].\text{vector}, \text{direction}(\text{halfedges}[h]), \theta')$$

where $\theta' = \theta * \text{halfgedges}[h].\text{direction}$ and

$$\text{rotate}(V, K, \theta) = V\cos(\theta) + (K \times V)\sin(\theta) + K(K \cdot V)(1 - \cos(\theta))$$

is the vector $V'$ representing vector $V$ rotated with respect to the unit axis vector $K$ by the angle $\theta$, This expression is known as the Axis–Angle rotation or Rodrigues' rotation formula. Let $\text{plane}(\text{halfedges}[h]) = \{a, b, c, d\}$ be the components of the plane incident to the half edge $\text{halfedges}[h]$. The vertices of the resultant interlocking block are defined by the intersection of planes associated with three consecutive half edges of a face. Then, a vertex $v = \text{intersect}(\text{plane}(\text{prev}), \text{plane}(\text{curr}), \text{plane}(\text{next}))$ where $\text{prev} = \text{halfedges}[h].\text{previous}$, $\text{curr} = \text{halfedges}[h]$, $\text{next} = \text{halfedges}[h].\text{next}$ and $\text{intersect}(A, B, C)$ is the intersection point of the planes $A$, $B$ and $C$ assuming the ranks of the coefficient matrix and the augmented matrix from the linear system defined by the components of the three planes is equal to 3, which means the three planes intersect at a point. Figure 3.2 shows the elements, from two neighboring squares, used by TAM to generate the respective interlocking pieces.

### 3.3.3 Observations

A TIC is valid when the constructed polyhedra keep the interlocking behavior and do not overlap with each other. The resultant TICs generated by the TAM are valid under certain conditions. First, the faces of the tessellation mesh must be regular, and second, the faces must exist in the same plane. The interlocking behavior guaranteed by the TAM is formally established in [2]. When the faces of the mesh are not regular, or the faces are not coplanar to each other, then overlapping artifacts appear in the resultant configuration. Building a TIC using the TAM on meshes with non–regular faces will generate an invalid configuration due to overlapping polyhedra. In the

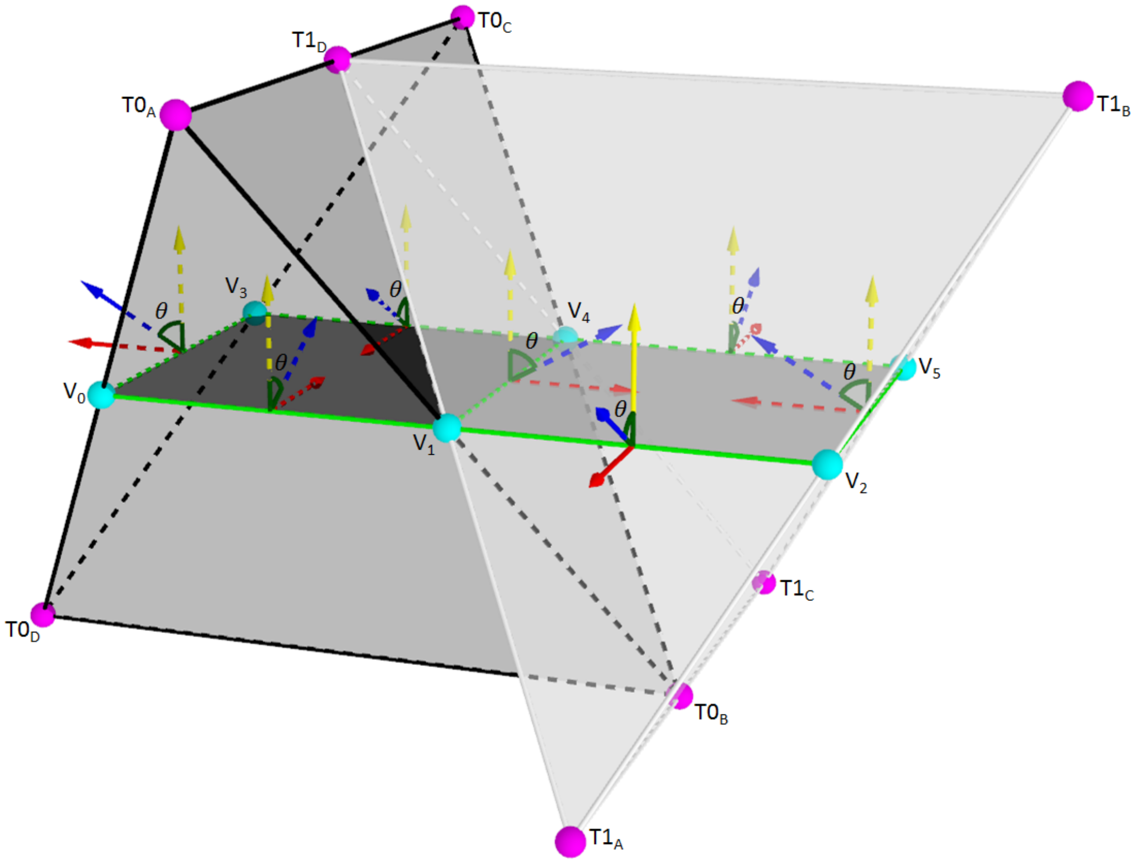
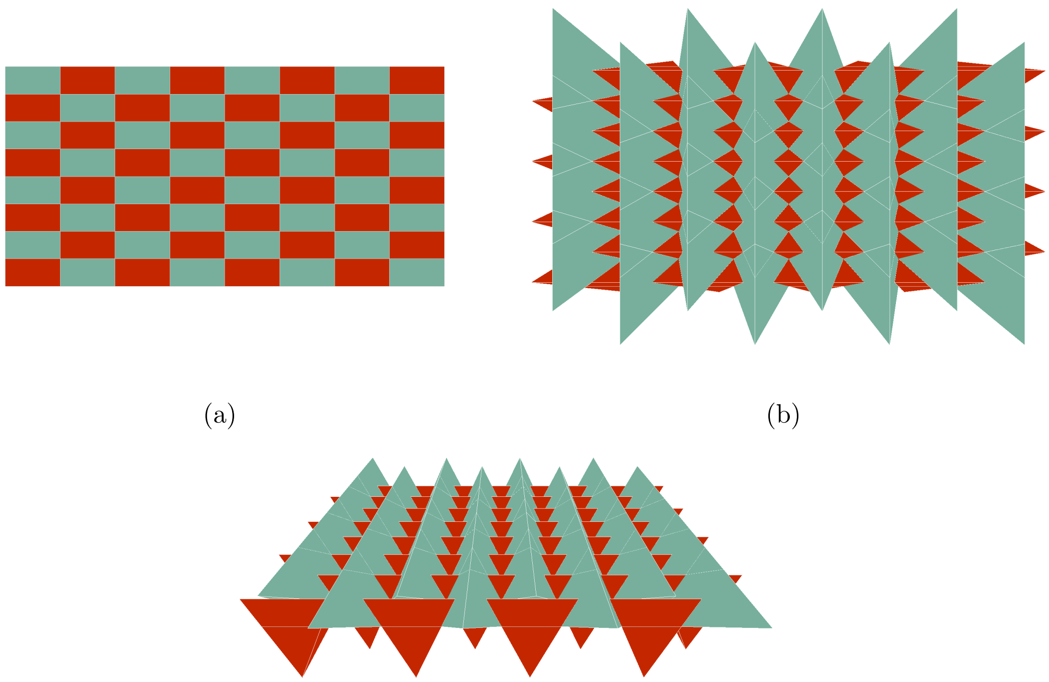


Figure 3.2. Elements used by the TAM to generate the pieces. Cyan points are the vertices of two squares in a tessellation. Red arrows describe the directions associated with the edges of the squares. Yellow arrows are the normal vectors of the edges. Blue arrows are the rotated vectors that describe the tilted planes by an angle $\theta$. Magenta points are the vertices of the resultant interlocking pieces T0 and T1 from the left and right squares, respectively.

original description of the method, all edges get the same tilting angle to determine their respective incident tilted planes. Nevertheless, using the same angle value for all edges does not guarantee that the resulting configuration will be valid in a more general or less–regular tessellation. An example of such situation is a TIC based on a rectangular mesh and generated using the TAM, Figure 3.3 shows the resultant TIC using $\theta = \frac{\pi}{3}$ on the edges. The resultant TIC shows two issues:



(a)

(b)

(c)

Figure 3.3. TIC based on a rectangular mesh using the TAM: (a) Initial mesh, (b) Obtained tetrahedra with $\theta = \frac{\pi}{3}$, (c) Oblique visualization.

- Overlapping Polyhedra: The resultant tetrahedra overlap along a row or a column of the rectangular mesh.
- Polyhedral Misalignment: The Resultant tetrahedra are not appropriately aligned such that their equatorial sections are not incident to their respective tiles.

Forcing the central section of a piece to be incident to the respective face in the mesh solves both issues. Referring to Figure 3.4, let $f \in F$ be a face of the mesh, let $C_f$ be the chosen center point within the face, let $N_f$ be the normal of the face. Assume that the vertices of the face are coplanar (i.e., the face exists in a plane). Let $T_f$ and $B_f$ be the respective points on the top and bottom sections of the resultant piece found along the line defined by $C_f$ and $N_f$, that is, $T_f = C_f + \lambda N_f$ and $B_f = C_f - \lambda N_f$ for some scalar $\lambda \in \mathbb{R}^+$. The resultant piece from $f$ is valid for a TIC if its equatorial section exists on the same plane as $f$, which is equivalent to $\text{dist}(C_f, T_f) = \text{dist}(C_f, B_f)$ where $\text{dist}(A, B)$ is the Euclidean distance between points $A$ and $B$. Then, the resultant piece is valid if both top and bottom sections are at the same distance with respect of $C_f$, and both $T_f$ and $B_f$ are in their respective sections. For a tetrahedron, such sections are the top and bottom edges. For an antiprism, such sections are the top and bottom faces, which are rotated a $\frac{\pi}{n}$ angle to each other, where $n$ is the number of sides of the even–sided polygonal tile.

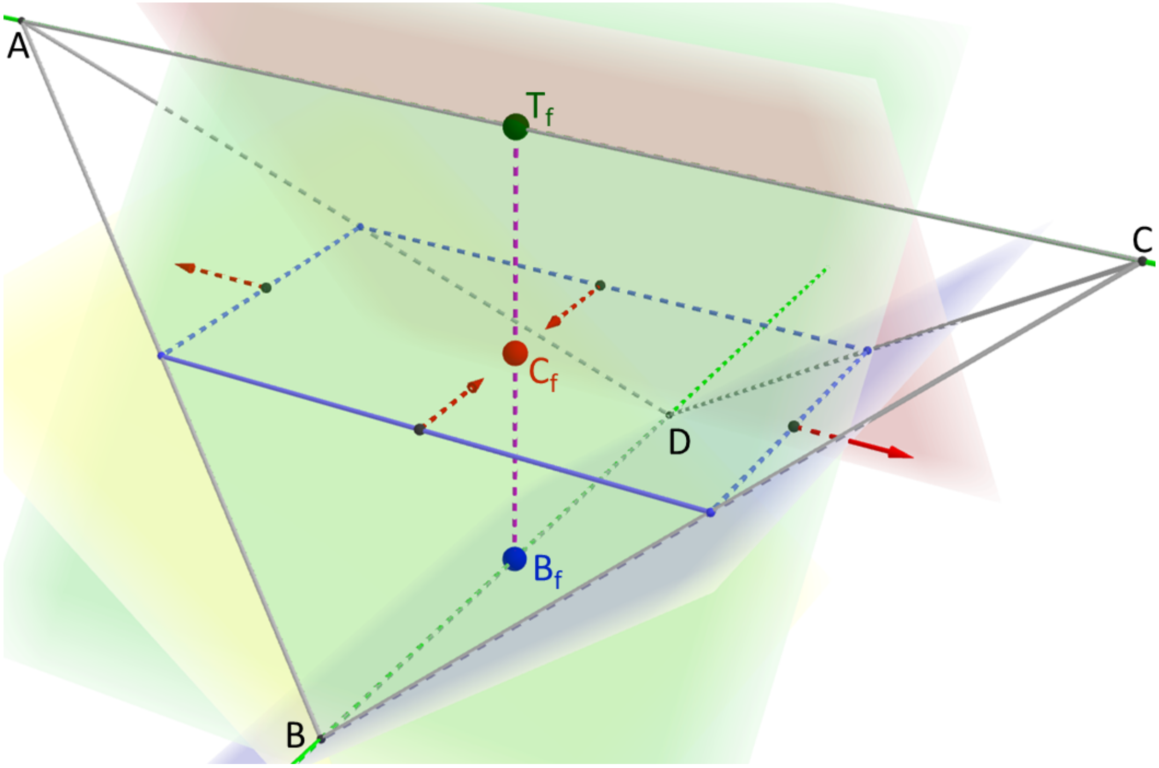


Figure 3.4. Valid tetrahedron from a rectangular face based on the distance from $C_f$ to both $T_f$ and $B_f$.

The previous observation leads to the definition of a required constraint for interlocking tetrahedra and antiprisms: the vertices of a piece must lie on their respective top or bottom sections of the piece. Since each vertex is the intersection point of the tilted incident planes from three consecutive edges in a face, then the constraint must be represented by a parameter during the definition of such planes. TAM defines each plane using the rotated direction vector and the vertices of an edge. The method knows the vertices of the face; then, the rotation angle is the remaining parameter left to implement the required constraint. It means that for every edge $e$ in a face, the rotation angle $\theta_e$ must be calculated such that the vertex determined by the intersection of three consecutive tilted planes comply with the constraint. For certain tessellations, such as the checkerboard, the same angle value applies to all edges to generate valid assemblies. However, such a statement does not hold for other quadrilaterals or even–sided polygons. Then, the angles could be user input or the output of an algorithmic search for correct tilting angles on each edge. Figure 3.4 shows the geometric nature of the problem for the case of a rectangular face.

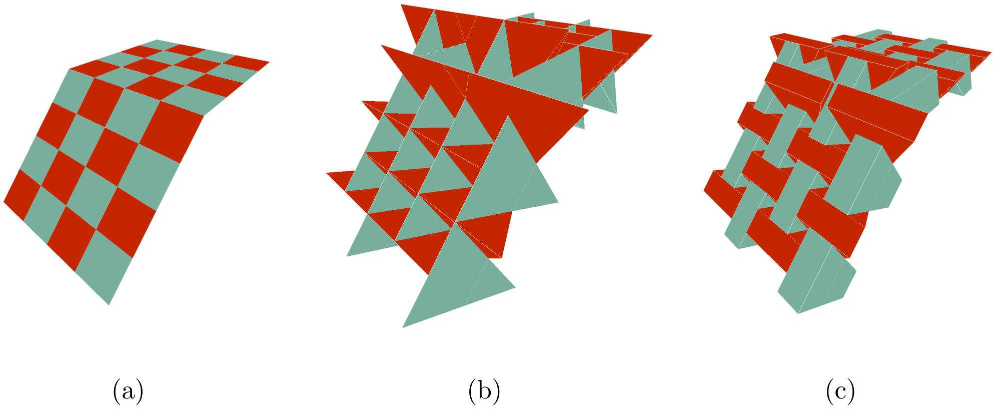


Figure 3.5. TICs based on a hinged mesh, generated using the TAM with $\theta = \frac{\pi}{3}$ and $\theta_e = \frac{3\pi}{4}$: (a) Initial mesh, (b) Obtained tetrahedra, (c) Truncated tetrahedra until overlapping sections are removed.

The TAM also generates no valid TICs when the incident faces of an edge are not coplanar. Let $a, b \in F$ be two faces in a mesh both incident to a common edge $e$, let $\theta_e$ be the dihedral angle between $a$ and $b$. The generated pieces from such faces present overlapping sections concerning two–step neighboring pieces due to their shape elongations caused by the additional tilting angle value provided by $\theta_e$. A practical solution for this problem is the truncation of the generated pieces until the removal of all overlapping sections between them. Still, the truncated pieces must keep the contact interface between them such that the Topological Interlocking principle holds, truncating the pieces more than required might eliminate such principle. Figure 9 shows both a TIC and the truncated pieces based on a hinged mesh with $\theta_e = \frac{3\pi}{4}$. Meshes with non–planar neighboring faces are common for computational purposes such as 3D modeling, data visualization, and finite element analysis, among others. Considering such meshes is important to generate valid TICs.

## 3.4 Height–Bisection Method

As described in the previous subsection, a problem during a TIC generation process is setting the right tilting angle for each edge. A slow solution for the problem would be to define first a seed angle value for every edge in the mesh. Then a TIC is generated and tested for correctness. If an overlap is detected, the conflicting edge angle values would be suitably adjusted. These steps repeat until the resultant configuration is valid. Adjusting the angles requires the traversal of all the edges in the mesh to recalculate their angle values.

### 3.4.1 Top/Bottom Sections and Heights

The polyhedra generated using both a checkerboard and the TAM constitute a structure of significant interest. Dyskin *et al.* in [4] reported that both top and bottom sections of the pieces are a grid of squares. When the tessellation squares of the midplane move towards the top and bottom sections, the squares are said to evolve. The vertices of the grid correspond to the projection of the vertices of the generated tetrahedra over the checkerboard. Similar grids are obtained from valid TICs based on meshes with quadrilateral tiles and generated using the TAM. From a geometric perspective, the grid is a structure that contains the connectivity information between the faces of the mesh. Each face $f \in F$ is represented by a point $p_f$, a line segment between points $p_a$ and $p_b$ is defined if there exists an edge $e$ such that faces $a, b \in F$ are incident to it. This description is the classical representation of the dual graph of a mesh; however, it omits infinite line segments. The projection of the vertices from the squared grid over the checkerboard represents the centers of the tiles. Such centers lie at the same location as the respective barycenter, centroid, and the diagonals of each tile. However, the mentioned centers do not lie at the same location for all polygons (and could even be non–existent as for irregular polygons and their intersecting diagonals). In a technical sense, a center could be any point within the face that does not lie on edge. Then, any center point works to build a grid if its calculation applies to all faces in the mesh.

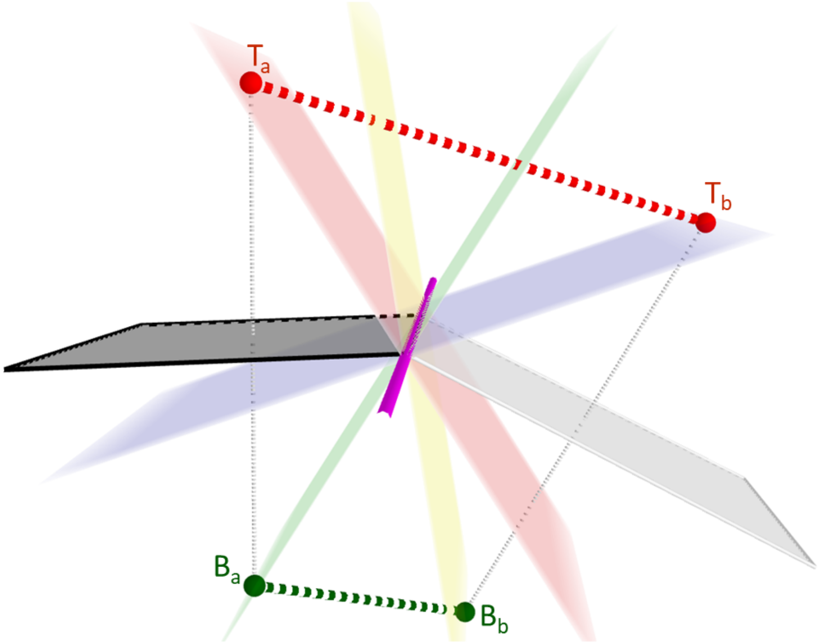


Figure 3.6. Four planes incident to an edge, each one containing a particular section point. Black quadrilateral corresponds to face $a$. White quadrilateral corresponds to face $b$. The red line segment represents the connection between the top section points $T_a$ and $T_b$. The green line segment represents the connection between the bottom section points $B_a$ and $B_b$. Red plane contains $T_a$, green plane contains $B_a$, blue plane contains $T_b$, yellow plane contains $B_b$. Planes are incident to the edge described in magenta (edge is extended for visualization purposes only).

The dual graph of a mesh is then a structure that anticipates the orientation of both top and bottom sections of a valid TIC based on such mesh. For a face $f \in F$ with center $C_f$ and normal vector $N_f$ its respective top and bottom section points $T_f$ and $B_f$ are defined as $T_f = C_f + \lambda N_f$ and $B_f = C_f - \lambda N_f$ for some scalar $\lambda \in \mathbb{R}^+$. Let $a, b \in F$ be two faces of the mesh incident at edge $e$, there exist four planes $P_1$, $P_2$, $P_3$ and $P_4$ that pass through the edge $e$ and contain section points $T_a$, $B_a$, $T_b$ and $B_b$ respectively. Figure 3.6 shows an example of the four planes incident to the edge, each one of them containing its respective section point. For meshes with regular faces the planes $P_1$ and $P_4$ are the same since section points $T_a$, $B_b$ and edge $e$ are coplanar, the same situation occurs with planes $P_2$ and $P_3$ and their respective section points. Let $e_d$ be the direction value of edge $e$ as described in the previous subsection, $e_d$ indicates which are the planes of interest associated with the angle value of the edge. When $e_d$ goes inwards with respect to face $a$ (which is equivalent of going outwards with respect to face $b$) then the planes of interest are $P_1$ and $P_4$. On the other hand, when $e_d$ goes outwards with respect to face $a$ (which is equivalent to going inward with respect of face $b$) then the planes of interest are $P_2$ and $P_3$. The bisector of the dihedral angle between such planes is the tilting angle value associated with edge $e$ required to generate a TIC based on the mesh. Then, the angle value for each edge in the mesh is a function of the selected center $C_f$ and $\lambda$ which is the distance from $C_f$ to the respective top and bottom section points.

### 3.4.2 Piece Design

We propose an algorithm for the angle setup problem for the case of irregular planar meshes based on the previous observations. Let DCEL$(M)$ and $d$ be defined as mentioned for the TAM. Select the center point $C$ to be used for all faces and set the height value $\lambda > 0$ which is the distance $\text{dist}(C_f, T_f)$ and $\text{dist}(C_f, B_f)$ for each face $f \in$ faces. For each face define: face$[f]$.center $=$ center$(f, C)$ where center$(f, C)$ is a function that receives a pointer $f$ to the face and the center type $C$ and re-

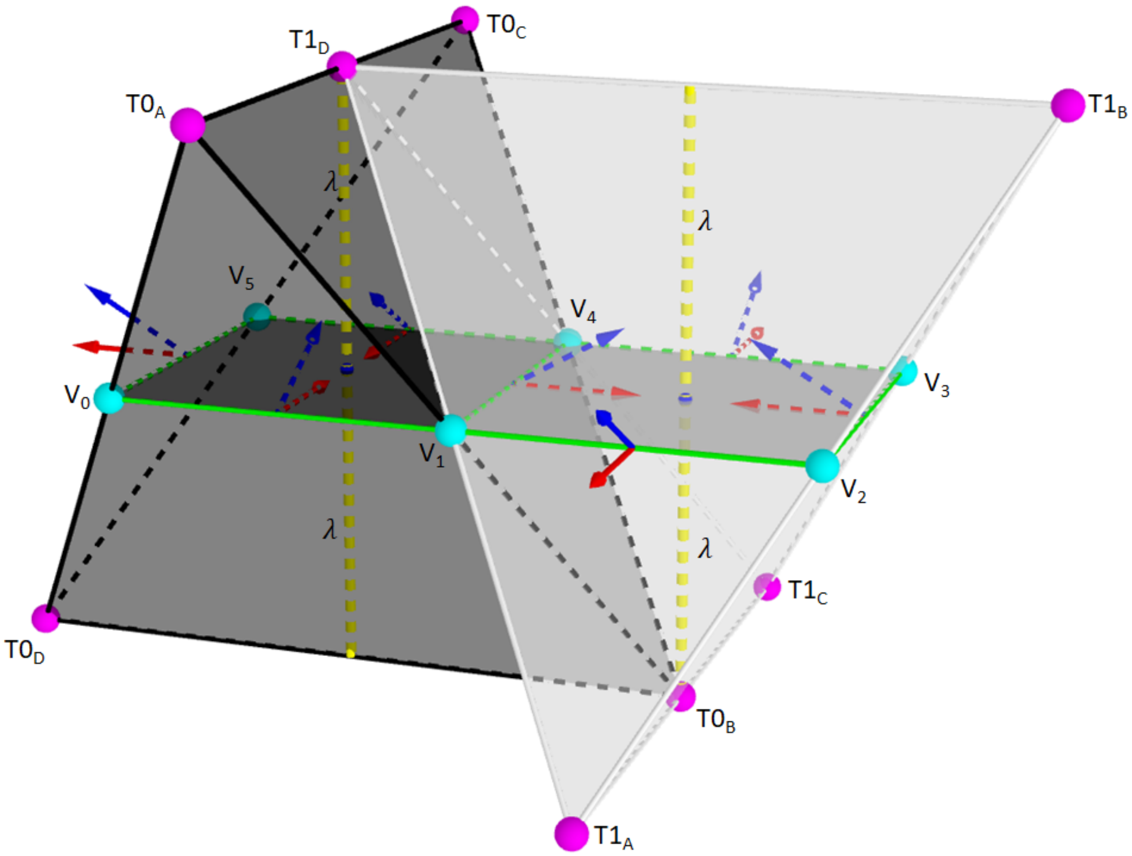


Figure 3.7. Elements used by HBM to generate the pieces. The cyan points are the vertices of two squares in a tessellation. The red arrows describe the directions associated with the edges of the squares. The yellow dashed lines are the line segments of length $\lambda$ representing the distance from a face center point to the respective top and bottom sections. The blue arrows are the rotated vectors that describe the tilted planes. The magenta points are the vertices of the resultant interlocking pieces $T0$ and $T1$ from the left and right squares, respectively.

turns the coordinates of the requested center point. The top and bottom section points associated to each face are defined as: faces$[f]$.top $=$ faces$[f]$.center $+ \lambda N_f$ and faces$[f]$.bottom $=$ faces$[f]$.center $- \lambda N_f$, where $N_f$ is the normalized normal of the face $f$. For each half edge $h$ in the face with an existing twin half edge set:

$$A = \text{halfedges}[h].\text{start}$$

$$B = \text{halfedges}[h].\text{end}$$

$$D = \begin{cases} \text{faces}[f].\text{bottom} & \text{halfedges}[h].\text{direction} == 1 \\ \text{faces}[f].\text{top} & \text{halfedges}[h].\text{direction} == -1 \end{cases}$$

$$E = \begin{cases} \text{halfedges}[h].\text{twin.face.top} & \text{halfedges}[h].\text{direction} == 1 \\ \text{halfedges}[h].\text{twin.face.bottom} & \text{halfedges}[h].\text{direction} == -1 \end{cases}$$

that is, points $A$, $B$, $D$ and $E$ define the planes of interest $P_1 = \text{plane}(A, B, D)$ and $P_2 = \text{plane}(A, B, E)$. Let $N_1 = \|DA \times DB\|$ and $N_2 = \|EB \times EA\|$ be the normal vectors of the respective planes of interest. The required bisector plane is represented by its normal vector: halfedges$[h]$.vector $= \|\frac{1}{2}(N_1 + N_2)\|$

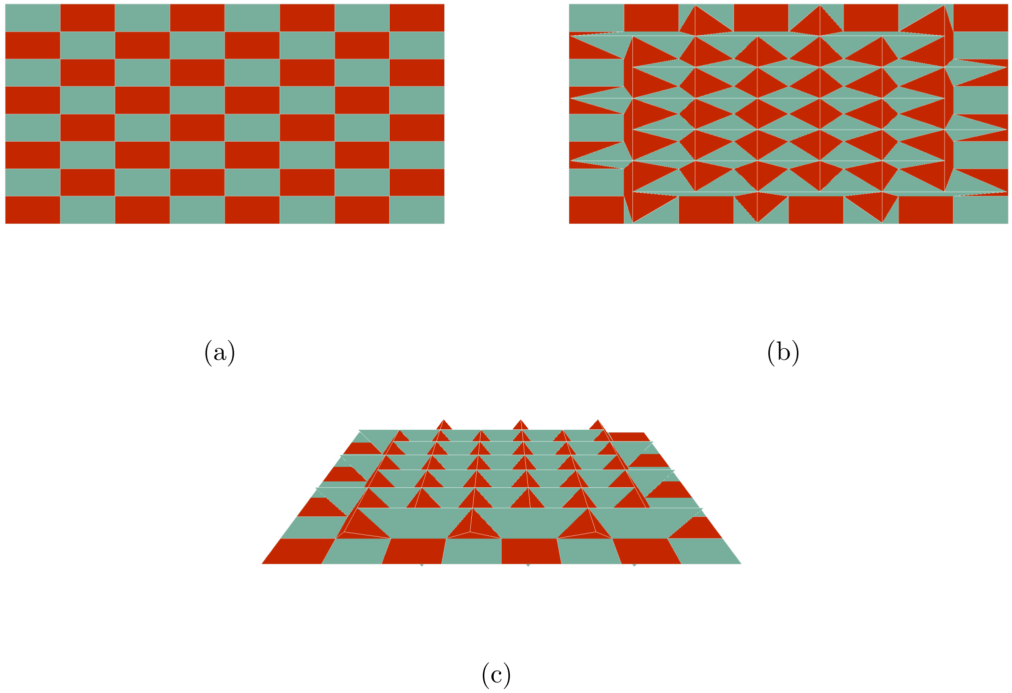
Since the direction vector is associated with an edge, it is valid to set

$$\text{halfedges}[h].\text{twin.vector} = \text{halfedges}[h].\text{vector}$$

The intersection of planes associated with three consecutive half edges in a face (as described for the TAM) define vertices of the resultant interlocking block. Figure 3.7 shows the elements, from two neighboring squares, used by HBM to generate the respective interlocking pieces.

The HBM generates the pieces of the configuration using two parameters: a height value and a center of the faces. The method does not calculate the actual tilting angle for each edge. The selection of a face center point opens the door to slightly different but congruent TICs given the same mesh and height value. As mentioned before, the calculation of such center point must be the same for all faces of the mesh. The actual angle value (as required in the TAM) is the angle between the tilted vector and the normal vector of the edge. Figure 3.8 shows the TIC generated using the HBM on a rectangular mesh (the same used in Figure 3.3).

An immediate consequence of the proposed method is the non–generation of pieces from faces next to the outer face of the mesh. That is, no piece results from a face where at least one of its half edges has no twin half edge, or the incident face of the twin half edge is non–existent. Such a situation does not generate a piece due to the missing geometric information required for the calculation of the tilting vector of the respective edge.


(a)

(b)

(c)

Figure 3.8. TIC on a rectangular mesh using the HBM: (a) Initial mesh, (b) Obtained tetrahedra with $\lambda = 0.2$, (c) Oblique visualization.

## 3.5 Results

We have described two TIC generation methods: TAM based on tilting angle values, and HBM based on height values and center choice. The former is traditional to handle the piece generation process. The latter is an improvement to calculate the expected rotation vectors based on both top and bottom section grids. The results in this section compare the two TIC methods for a variety of tessellations in the plane. We also consider TICs based on cylindrical, spherical, and toroidal tessellations.

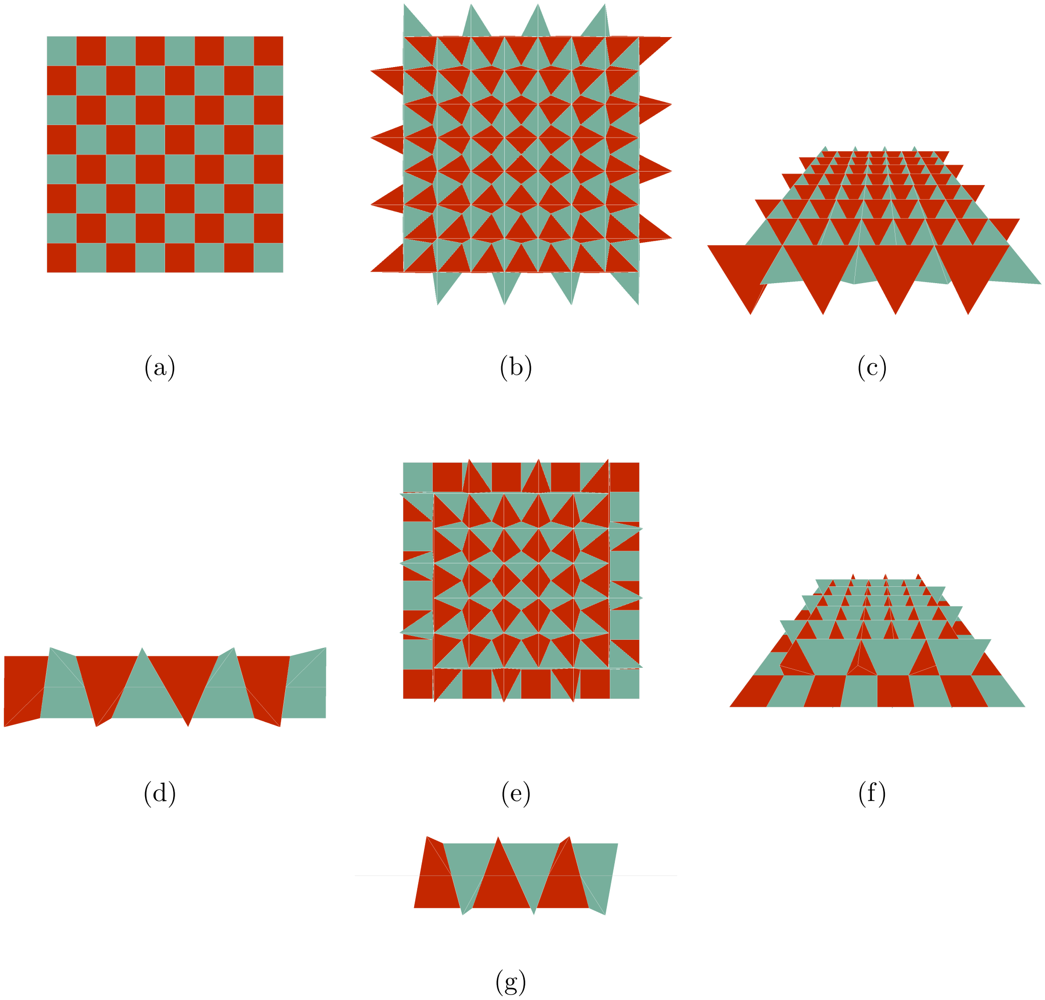
### 3.5.1 2D Meshes

When generating a TIC, there are two characteristics:

- The equatorial mid–section of each piece lies in the same plane as the face of the tessellation mesh.
- On regular planar meshes, the pieces are aligned to each other using different tilting angles for their edges.

We generated multiple TICs using the described generation methods on meshes based on simple polygons further subdivided using mid–point subdivision [37]. In such meshes, the face topology is quadrilateral. Figure 3.9 shows the resultant configurations starting with a subdivided square. This shape is the basic case where valid TICs are possible by using any parameter values for both generation methods. The resultant configurations preserve the alignment of the pieces. That is, the central section of the pieces exists in the same plane as their corresponding faces from the tessellation. As expected, the HBM does not generate pieces from the faces in the boundary of the mesh due to missing neighbor information. A solution is to assume the missing neighbor information matches perfectly with the geometry of the respective face.

Using a different initial shape affects the resultant configurations significantly. Figure 3.3 shows the generated TIC using the TAM based on a subdivided rectangle. The resultant pieces present overlapping sections around the vertices. This situation is caused by using the same angle value for all edges in the mesh. Additionally, the pieces do not align with each other. Both top and bottom sections of the pieces do not represent the grid showing the evolution of their mid–section towards them. On



(a) (b) (c)

(d) (e) (f)

(g)

Figure 3.9. Generated TICs starting from a square with 3 subdivision iterations. (a) Geometric domain, (b) TIC using TAM (top view), (c) TIC using TAM (oblique view), (d) TIC using TAM (alignment), (e) TIC using HBM (top view), (f) TIC using HBM (oblique view), (g) TIC using HBM (alignment).

the other hand, the HBM generates a TIC with no overlapping sections and with full alignment between pieces. The generated configuration is shown in Figure 3.8. Both top and bottom sections of the pieces have the expected evolution grid. Only the pieces from the faces at the boundary are incorrect due to the missing geometric information. We generated TICs based on various types of quadrilaterals using both methods. The observations made from the configuration based on a subdivided rectangle apply as well for such quadrilaterals.

The generation methods can work on other convex polygons after subdividing them. Figure 3.10 shows the resultant TICs based on a subdivided triangle. As expected, the TAM generates pieces that overlap at the vertices; furthermore, the pieces do not align with each other. On the other hand, the HBM generates pieces with reduced overlap. The same observations apply to the generated configurations based on the subdivided pentagon and hexagon.

### 3.5.2 3D Meshes

A major goal of TIC generation is to produce valid configurations for any given 3D mesh. While the 2D case must deal with open meshes at the boundaries, this does not have to be the case in 3D where geometric domains could be partly open (e.g., cylinders, paraboloids) or fully closed (e.g., solids, spheres, tori). The cylinder (Figure 3.11) is the first 3D shape into consideration. Since we can map a checkerboard onto its surface, both generation methods can be used and are intuitive. Our experiments indicate that the TAM generates pieces with large significant overlap and misalignment. On the other hand, the HBM reduces the amount of overlap to small sections between longitudinal pieces only, leaving meridian pieces valid. The resultant configuration shows that the equatorial sections of the pieces lay on the same plane as their respective faces of the mesh.

We approximate spherical surfaces by using inflated subdivided solids. A Platonic solid approximates a sphere by subdividing its faces and normalizing the distances of the vertices. Figure 3.12 shows a sphere approximated by an inflated dodecahedron with subdivided faces. The resultant configuration using the TAM generates bumps along the surface; each bump resembles a face from the original Platonic Solid. Such bumps are caused by piece misalignment due to the same tilting angle used for all pieces. However, after truncating the resultant pieces, the configuration resembles

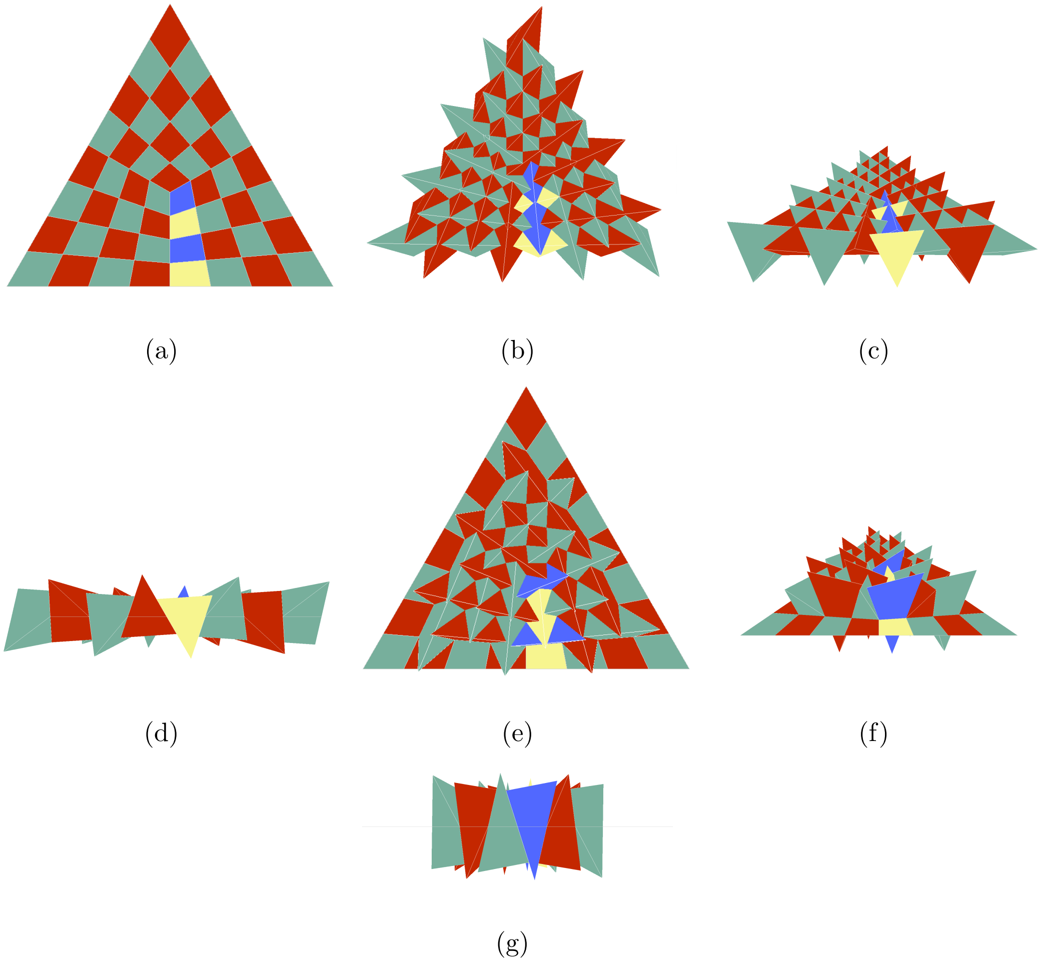
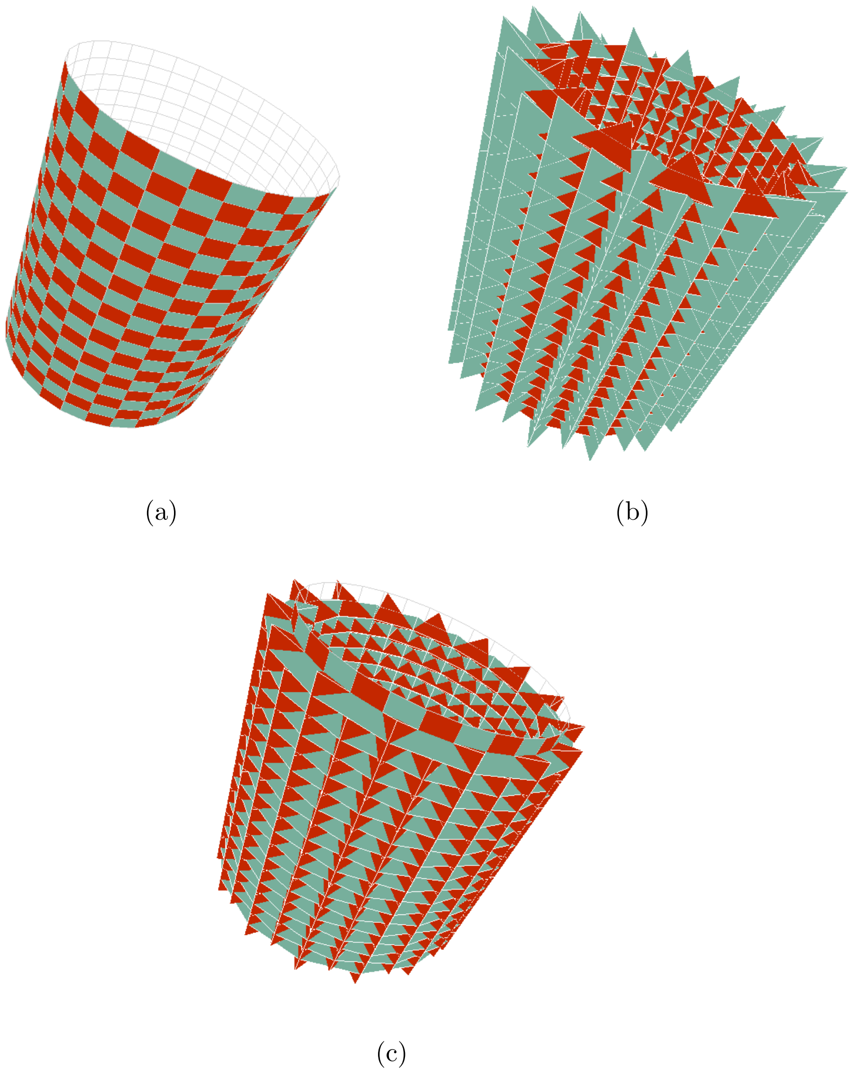


Figure 3.10. Generated TICs starting from a triangle with 3 subdivision iterations. (a) Geometric domain, (b) TIC using TAM (top view), (c) TIC using TAM (oblique view), (d) TIC using TAM (alignment), (e) TIC using HBM (top view), (f) TIC using HBM (oblique view), (g) TIC using HBM (alignment).

a nexorade as described in [10]. Such a result suggests there could exist a framework to convert one problem representation to the other. This connection is not surprising since TICs and nexorades are generalized versions of the Abeille's bond of tetrahedron–based pieces. Finally, when using the HBM, the spherical surface is preserved by the resultant configuration. This result comes from the alignment of the pieces with their respective faces of the mesh.



(a)

(b)

(c)

Figure 3.11. TICs on a cylindrical mesh: (a) cylindrical mesh with radius 2 and height 5, (b) TIC using the TAM with $\theta_e = \frac{\pi}{3}$ , (c) TIC using the HBM with $h = 1$ and $C_t$ = barycenter.

We have considered toroidal meshes. Those meshes have both positive and negative curvature faces. Figure 3.13 shows a torus mesh and the resultant TICs using both generation methods. As with the previous 3D meshes, the TAM does not guarantee the alignment of the pieces and create overlapping pieces. The HBM reduces such overlapping due to the piece alignment. Furthermore, the resultant configura-

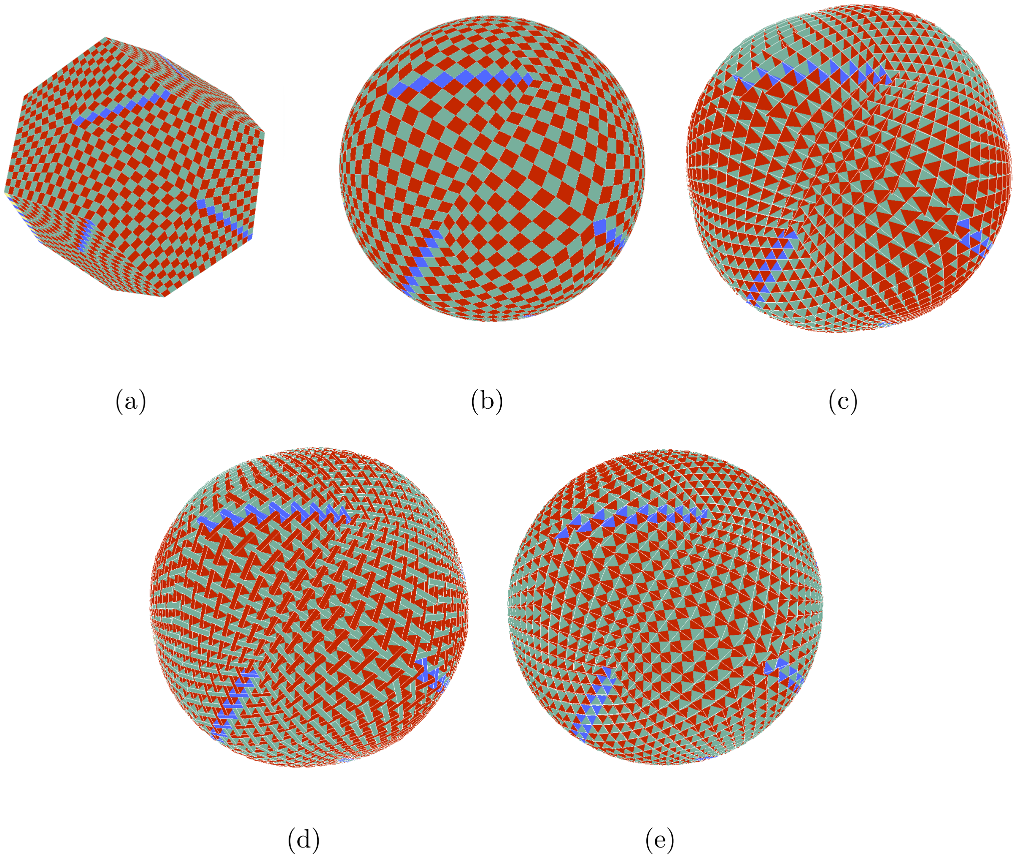
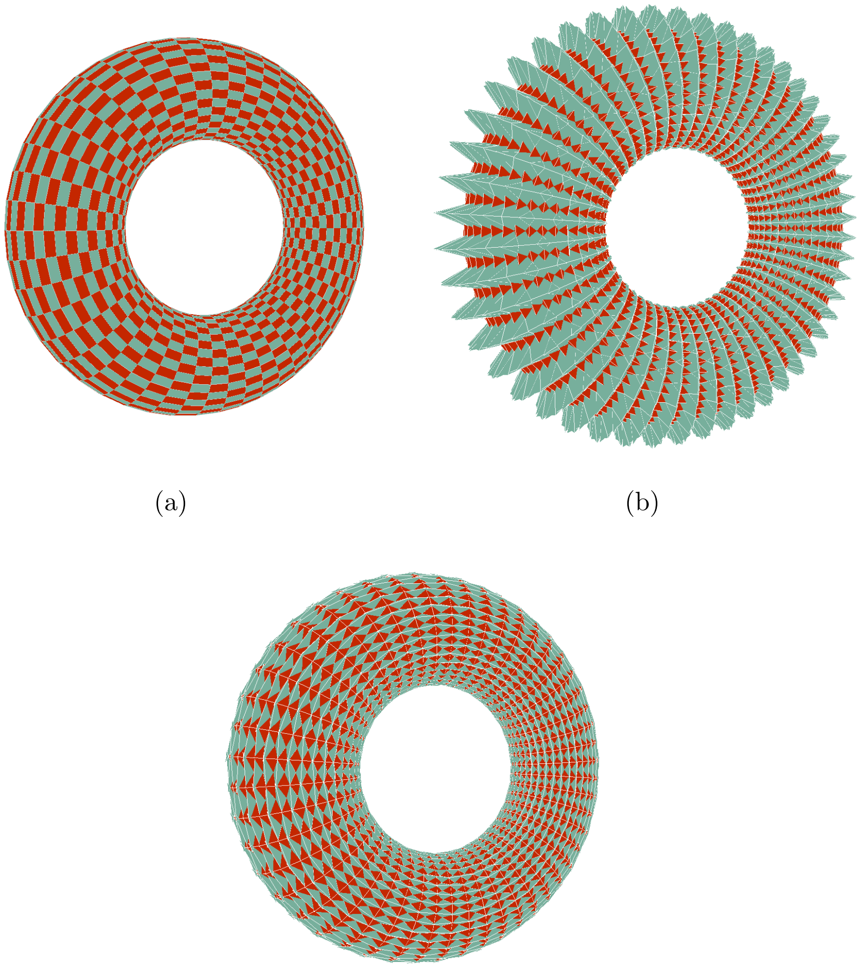


Figure 3.12. TICs on a spherical mesh: (a) Subdivided dodecahedron, (b) Inflated dodecahedron (normalized vertices), (c) TIC using the TAM with $\theta_e = \frac{\pi}{3}$ , (d) TIC using the TAM with $\theta_e = \frac{\pi}{3}$ with truncated pieces, (e) TIC using the HBM with $h = 1$ and $C_t =$ barycenter. Colors have no meaning but differentiate adjacent tiles and pieces.

tion using the Height–Bisection resembles the original mesh more accurately when compared with the configuration generated using the TAM.

## 3.6 Discussion

We introduced the HBM as a TIC generation method that preserves the alignment of the top and bottom sections of the pieces as described by the evolution grids of the mesh. The method requires a height parameter $\lambda$ and a center $C$ from each face. The calculation of the rotation vectors for all edges requires a single iteration over the



(a)

(b)

(c)

Figure 3.13. TICs on a torus mesh: (a) Torus with external radius 3 and internal radius 1, (b) TIC using the TAM with $\theta_e = \frac{\pi}{3}$ , (c) TIC using the HBM with $h = 1$ and $C_t =$ barycenter.

faces from the geometric domain. Calculation such values in a single iteration is an improvement concerning the TAM. The resultant configurations maintain the piece alignment. That is, the equatorial section of each piece exists on the same plane as its corresponding face in the mesh.

We tested both generation methods using 2D and 3D geometric domains. The generated TICs suggest that the HBM produces configuration with less overlapping than the configurations generated using the TAM. Furthermore, applying a slight truncation over the sections of the pieces removes the overlapping sections. In some cases, the overlapping sections are non–existent due to the correct tilting vector calculated by the HBM. Finding a valid TIC for a 3D mesh increases the complexity of the solution due to the additional information given in the geometric domain. The dihedral angle between incident faces introduces an additional rotation angle that alters the location of the top and bottom section points for each face. It could be the case that such section points are not suitable to generate an interlocking piece. Even though piece truncation seems to be the simplest solution, it involves the possibility of invalidating the piece (and even the entire TIC) when the cuts nullify the interface between two neighboring pieces. In terms of shape detail, it was found an increase of midpoint subdivisions among the faces is required for the resultant TIC to resemble the original domain; however, the trade–off to increase the level of detail is an increase in the number of calculations to be performed in a single iteration.

# 4 PIECE VALIDATION AND SHAPE–FIT GENERATION

## 4.1 Introduction

The TIC generation methods discussed so far return an assembly of pieces based on the parameters set at the edges or tiles of a surface tessellation. Such methods generate a piece per tile in the domain; this process considers information from the neighborhood of the tile. By doing this, the generation methods guarantee that two adjacent pieces will fit appropriately to one other. However, some assemblies may have non–adjacent pieces with overlapping between them. Overlapping pieces are not valid for TI purposes, which forces their redesign to make them valid.

We first describe a valid piece in terms of the surface tessellation, generation parameters, and the generation method. Then, we introduce a search mechanism to adjust the generation parameters such that any two pieces in the assembly do not have an overlapping between them, or it is minimal. For the latter case, we consider clipping the pieces to make them valid. We introduce approaches to search the clipping parameters that remove the sections of a piece that make it not valid in its respective assembly.

Finally, we propose a TI generation method for assemblies based on non–planar surface tessellations composed of square tiles (e.g., barrel vault and cylinder). The method labels the tiles of the surface tessellation following a checkerboard pattern. A label describes one of two shapes: a regular tetrahedron or a quasi–tetrahedron. We name quasi–tetrahedron to the shape that results from fitting an overlapping tetrahedron within its neighbors. The returned assembly is a closed–form solution in terms of the surface tessellation and the side length of the square tiles.

## 4.2 Overview

The main goal is to generate TI assemblies, based on any surface tessellation, with no overlapping between its pieces. We consider two pieces to be valid if there is no 3D overlapping between them. Then an assembly is valid if all of its pieces are valid. Figure 4.1 shows our proposed pipeline to generate a valid assembly based on a given surface tessellation and a set of generation parameters. We first generate an initial assembly and check whether it is valid or not. If it is not valid, we readjust the generation parameters such that the resultant assembly is valid or the overlapping is minimal. In the latter case, we clip the sections of the pieces that make them not valid. This pipeline guarantees that the final assembly will be valid, and its overall shape resembles the given surface tessellation.

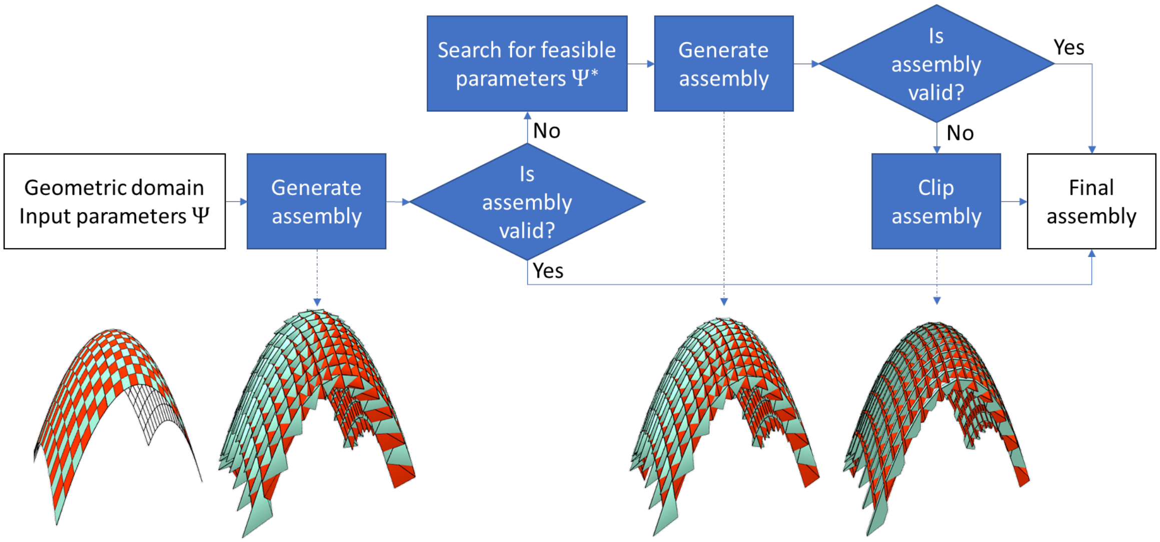


Figure 4.1. Pipeline to generate valid TI assemblies.

In section 4.3, we explain the requirements for a piece in an assembly to be valid. Then, we discuss how to search for generation parameters that either remove or reduce the overlapping between the pieces. It may be possible that such parameters do not remove all overlapping entirely. In section 4.4, we introduce three clipping approaches that remove the sections from a piece that make it not valid. The clipping approaches can be applied to any TI assembly directly. However, the amount of overlapping to remove is less when working with an assembly generated using optimal generation parameters. Finally, in section 4.5, we propose a TI generation method for valid pieces when the tiles of the surface tessellation are squares. The method generates both regular tetrahedra and quasi–tetrahedra in a single iteration over the tiles.

Having valid assemblies is a requirement for the static equilibrium analysis method discussed in Chapter 6.

## 4.3 Valid Assemblies and Feasible Generation Parameters

In this section, we address the issue of overlapping sections between pieces in an assembly. In Chapter 3, we showed valid TICs generated using specific surface tessellations and generation parameters. If the tiles of a planar surface tessellation are regular polygons and they have the same shape, then we can set the same single value to all generation parameters. These conditions guarantee that the resultant pieces of the assembly will have the same geometry. Moreover, the pieces will not overlap with each other. As an example, to generate a valid assembly of regular tetrahedra, we can use the TAM on a checkerboard and all tilting angles set to 54.74°.

Nothing guarantees the pieces of an assembly will not overlap if the surface tessellation is non–planar or its tiles are not regular polygons. In such cases, we need to check whether the pieces overlap with each other or not. If there are at least two pieces in the assembly with overlapping, then a redesign of the assembly must be considered. A redesign approach is to adjust the generation parameters such that the overlapping disappears. Still, there may be situations (due to the surface tessellation, generation parameters, or the generation method) where we cannot remove overlapping entirely. We then search for the set of generation parameters that reduced such overlapping to a minimal. In the next subsections, we describe using formal terms the overlapping problem and how to solve it by searching through the generation parameters space.

### 4.3.1 Valid Piece

We first consider the geometry of a piece as the intersection of a finite number of planes (as described in Section 3.3.2). We use the same terminology to define the overlapping section between two pieces in an assembly.

A TI assembly is valid from a geometric point of view if its pieces do not overlap with each other. The condition requires each piece to have a fixed location in space concerning its associated tile from the surface tessellation. Let $T = \{B_i \mid i \in 0, 1, ..., n-1\}$ be an assembly where $B_i = \{V_i, F_i\}$ is a piece of the assembly, and $n = |T|$ is the number of pieces in $T$. Since each face in a piece is planar, then $\forall f_i^k \in F_i, k = 0, 1, ..., m-1$ where $m = |F_i|$ is the number of faces of $B_i$, then $\exists! P_i^k = \text{plane}(N^k, Q^k)$ with normal vector $N^k$ and reference point $Q^k$ such that $f_i^k \in P_i^k$, which implies $B_i = \bigcap_{k=0}^{m-1} P_i^k$. This description is the definition of a convex polyhedron described by the intersection of finitely many half–spaces. A piece $B_i$ is valid if $B_i \cap^* B_j = \emptyset, \forall B_j \in T, j = 0, 1, 2, ...n-1, i \neq j$ where $\cap^*$ is the regularized intersection operator [38], and $\emptyset$ is a geometry of lower dimension whose representation is an empty set. If $B_i \cap^* B_j \neq \emptyset$ then both $B_i$ and $B_j$ are overlapping with each other, therefore they are considered to be not valid pieces.

Checking if a piece is valid is equivalent to determining whether it is in collision with another piece or not. Collision detection between polyhedra is a well–known problem and has been an active topic of research in Computer Graphics and Robotics fields. Surveys on the matter by Lin and Gottschalk [39], Jimenez *et al.* [40] and Ericson [41] explore a variety of techniques to detect collisions on either static or dynamic scenarios. The geometry representations for both convex and non–convex polyhedra vary between techniques. Modern collision detection approaches from Wilke *et al.* [42], Weller *et al.* [43] and Govender *et al.* [44] take advantage of GPU capabilities for parallel processing and real–time results when working with non-convex polyhedra, deformable shapes and topology–changing objects.

### 4.3.2 Overlapping Sections

We consider two pieces to overlap with each other if their regularized intersection is not a lower dimension geometry. Both TAM and HBM generation methods guarantee that adjacent pieces will fit appropriately according to the requirements of Topological Interlocking. That is, any two adjacent pieces in an assembly lock the push or pull motion from one another. The interaction between non–adjacent pieces is not required since it does not contribute to each other's interlocking behavior. Then, overlapping (if any) can only occur between two pieces at least two hops apart from each other along with the tiles of the surface tessellation.

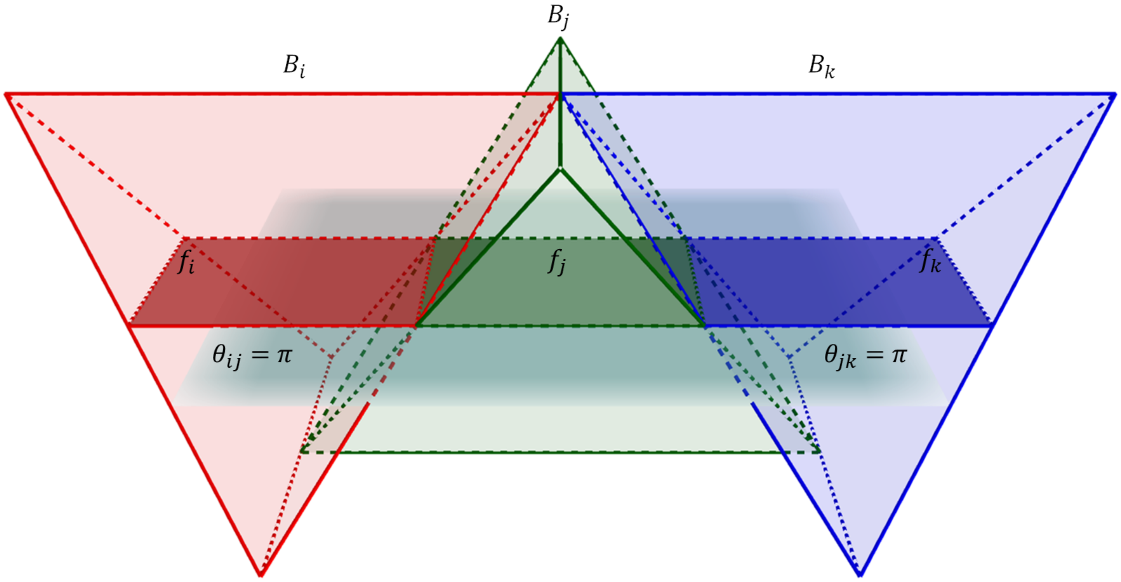


Figure 4.2. Coplanar tiles $f_i, f_j, f_k$ and respective valid pieces $B_i, B_j, B_k$

Determining if a piece is valid must be considered in terms of the surface tessellation, the generation parameters, and the generation method. Let's consider a surface tessellation $M$ composed of three square coplanar tiles $f_i, f_j, f_k$ of side length $l$ (example shown in Figure 4.2). In this case the dihedral angles between the tiles are $\theta_{ij} = \pi$ and $\theta_{jk} = \pi$). A TI generation method $G(M, \Psi)$ using input parameters $\Psi = \{\psi_i, \psi_j, \psi_k \mid \psi_i = \psi_j = \psi_k\}$ returns an assembly $T = \{B_i, B_j, B_k\}$. Assume without loss of generality that $\Psi$ is the set of parameters that makes $G$ to generate regular tetrahedra based on $M$ ($\psi = 54.74°$ for $G =$ TAM or $\psi = \frac{l}{\sqrt{2}}$ radians for $G =$ HBM, $\forall \psi \in \Psi$). In this case, all pieces are valid, which makes $\Psi$ feasible for $M$ using $G$. There are still infinitely many feasible parameters sets that generate valid pieces (although not regular) provided that $\psi_i = \psi_j = \psi_k$.

Reducing the dihedral angles $\theta_{ij}$ clockwise and $\theta_{jk}$ counterclockwise makes the tiles in $M$ to become non–planar, hence changing elements in the surface tessellation (e.g., vertices location and normal vectors for tiles $f_i$ and $f_k$). Such changes can make $G$ to return an assembly with invalid pieces using the same $\Psi$ as before (Figure 4.3 shows this situation). Since the returned assembly is not valid, then $\Psi$ is non–feasible for the modified $M$ using $G$.

We consider the sum of the volumes from all overlapping in an assembly. The resultant value indicates the lack of validity of the assembly. The assembly is valid if such value is zero. Otherwise, it is not valid. Given an assembly $T = \{B_i \mid i = 0, 1, ..., n\}$ generated using $G(M, \Psi)$, its lack of validity comes from the expression

$$y(T) = \frac{1}{2} \sum_{i=0}^{n-1} \sum_{\substack{j=0 \\ i \neq j}}^{n-1} \text{vol}(O_{ij}) \tag{4.1}$$

where $O_{ij} = B_i \cap^* B_j$ is the polyhedron representing the overlapping between pieces $B_i$ and $B_j$, and

$$\text{vol}(P) = \frac{1}{6} \sum_{i=0}^{m-1} a_i \cdot N_i \tag{4.2}$$

is the formula for the volume of a 3D polyhedron $P$ (with $m$ faces, and a point $a_i$ and normal vector $N_i$ for each face in $P$) based on the divergence theorem [45].

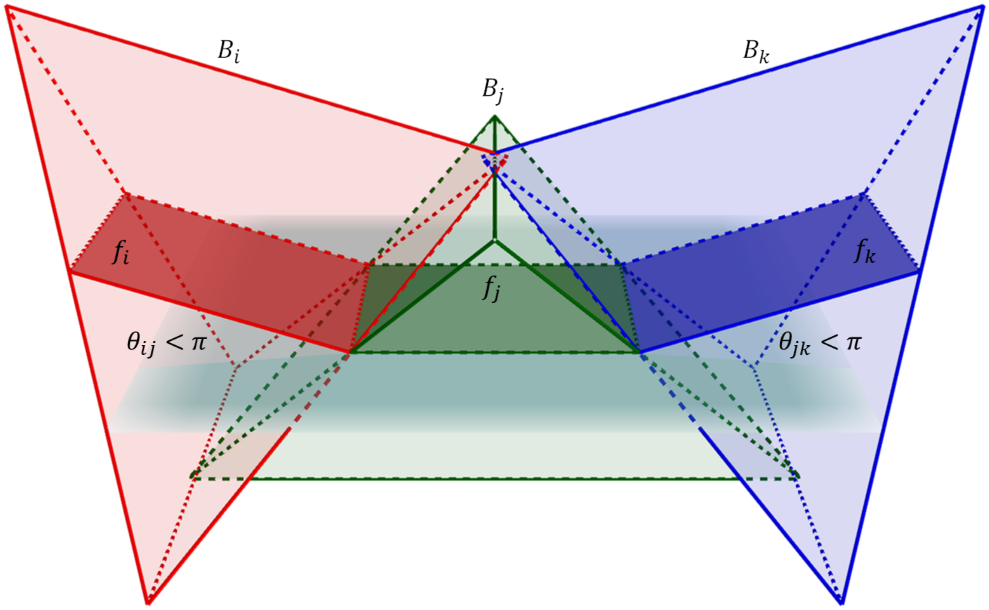
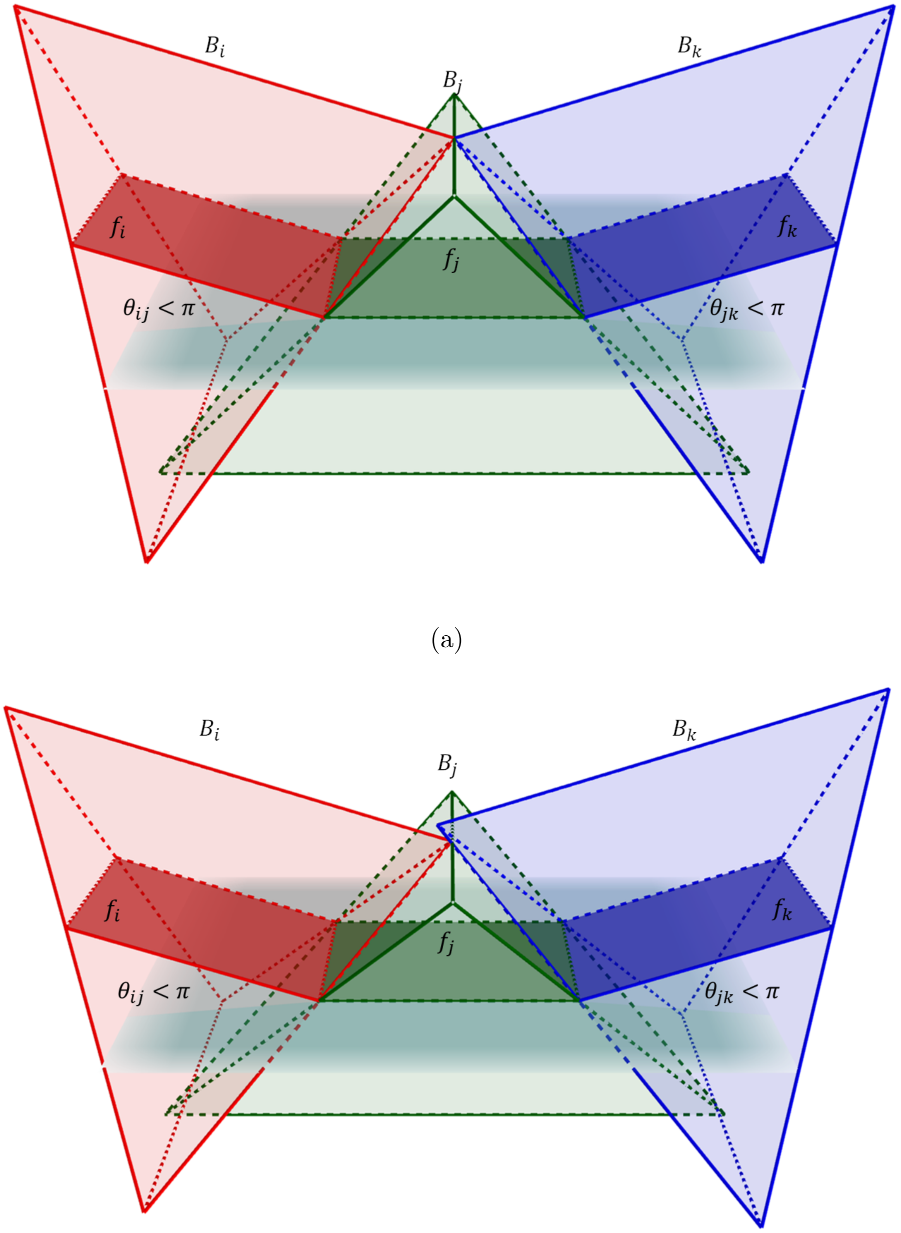


Figure 4.3. Overlapping between $B_i$ and $B_k$ caused by a non–feasible parameters set $\Psi$ for non–coplanar tiles.

### 4.3.3 Feasible Parameters Search

Finding feasible parameters set for a given surface tessellation and a generation method may fix the overlapping between pieces. Two solutions for the case shown in Figure 4.3 require changing specific generation parameters. The first solution changes $\psi_j$ such that $B_j$ separates $B_i$ and $B_k$ (see Figure 4.4(a)). The second solution changes $\psi_i$ so $B_i$ does not overlap with $B_k$ (see Figure 4.4(b)).

These solutions suggest that a feasible parameter set might not be unique for a given context. Still, a change at any $\psi \in \Psi$ requires checking whether the resultant assembly is valid or not. Doing so is a computationally expensive task. Furthermore, the existence of feasible parameters does not guarantee that the pieces will be useful in practice (e.g., pieces align to each other) or even functional (e.g., maintain static equilibrium, discussed in Chapter 6). Finally, it could be the case that the only feasible parameter set is the trivial solution $\Psi = \mathbf{0}$, which generates no pieces.



(a)

(b)

Figure 4.4. Changing $\Psi$ such that pieces $B_i$ and $B_k$ become valid. a) changing $\psi_j$, b) changing $\psi_i$

We can find a feasible parameters set by searching through the parameters space following a conventional optimization approach:

$$\begin{aligned} \underset{\psi \in \Psi}{\operatorname{argmax}} \min_{M,\Psi} \quad & y(T) \\ \text{s.t.} \quad & lb \leq \Psi \leq ub \end{aligned} \tag{4.3}$$

where $y(T)$ is the expression shown in 4.1, and $lb$ and $ub$ are the respective lower bound and upper bound vectors for $\Psi$.

## 4.4 Piece Clipping for Assembly Validation

In this section, we consider the problem of removing sections from the pieces of an assembly until they are valid. There are situations where it is not possible to find a set of generation parameters such that a generation method returns a valid assembly based on a given surface tessellation. The solution is to physically remove the overlapping sections until all of the pieces become valid.

The expression 4.3 may return an optimal set $\Psi^*$ that is not feasible but reduces overlapping to a minimal. We show practical solutions to validate pieces in an assembly. These approaches clip a piece in regards to a clipping plane, such that it removes the sections that make a piece not valid. None of such clipping approaches need a $\Psi^*$ to transform a not valid assembly into a valid one. Still, such an optimal assembly would require less clipping to become valid. Hence the shape of a clipped piece will be closer to the original.

### 4.4.1 Piece Clipping

We describe the clipping procedure on a piece as the regularized intersection between such a piece and a plane. A plane clips a piece if it passes through the piece and splits it up into two halves. Each half must regularized geometry.

Let $B = \{V, F\}$ be a piece in an assembly $T$, and $P = \text{plane}(N, Q)$ be an arbitrary plane with normal vector $N$ and reference point $Q$. We say $P$ clips $B$ into $B^+$ and

$B^-$ if $B = B^+ \cup^* B^-$ having $B^+ \neq \emptyset, B^- \neq \emptyset$ and volumes $\text{vol}(B^+) > 0, \text{vol}(B^-) > 0$. $B^+$ lies in the positive half space $P^+$ defined by $P$ (in the direction of $N$) and $B^-$ lies in the negative half space $P^-$ (in the opposite direction of $N$). Figure 4.5 shows a piece $B$ with respective $B^+$ and $B^-$ when clipped by a plane. A vertex $v_i \in V$ lies in $P^+$ if $(v_i - Q) \cdot N > 0$, $P^-$ if $(v_i - Q) \cdot N < 0$, or $P$ if $(v_i - Q) \cdot N = 0$. If all $v_i$ lie in the same half space then $P$ does not clip $B$ (which means either $B^+$ or $B^-$ is $\emptyset$). Moreover, if all vertices lie in the same half–space or $P$ itself, then $P$ is in contact with at least one vertex, edge, or face from $B$.

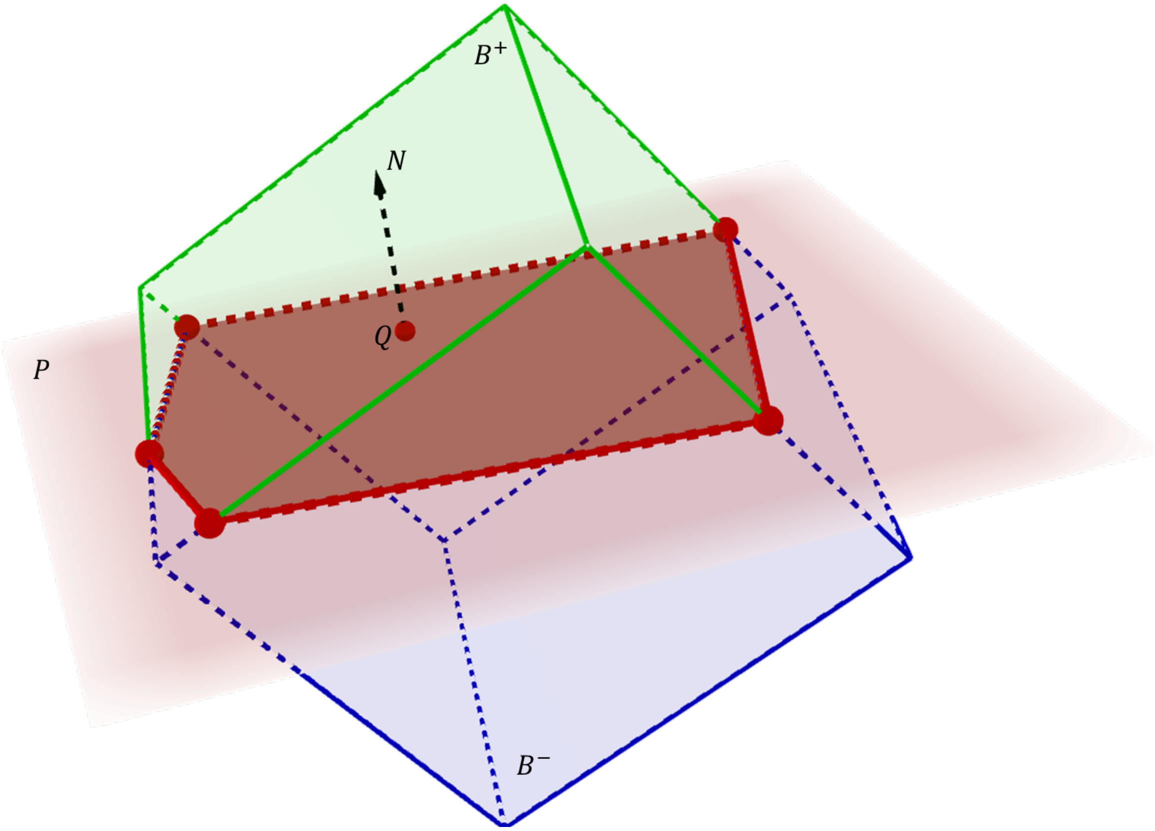


Figure 4.5. Piece $B$ clipped by plane $P = \text{plane}(N, Q)$. $B^+$ is shown in green, $B^-$ is shown in blue.

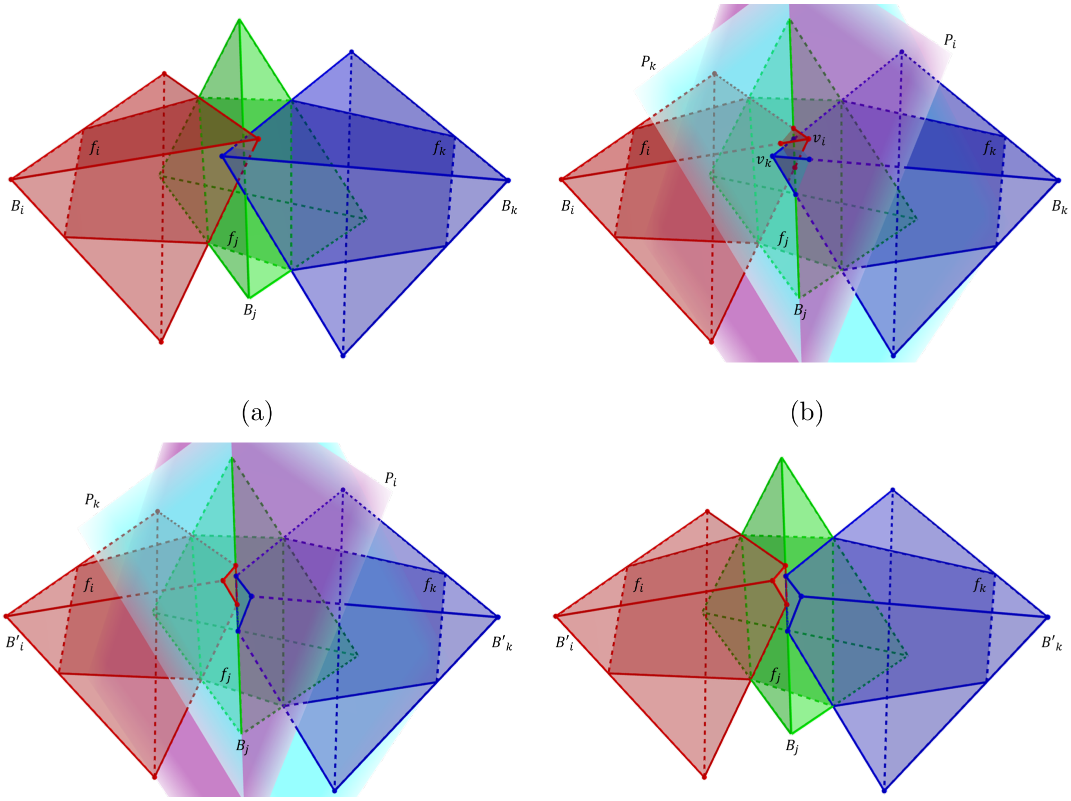
Pieces on a TI assembly are convex polyhedra represented using a DCEL. Such a format is suitable for the definition of polygon processing algorithms that manipulate the geometry. A clipping algorithm must label each vertex $v_i \in V$ according to its location with respect of $P$ ($P^+, P^-$ or $P$ itself). Then, the algorithm determines the intersection vertices along the edges whose endpoints lie on different half–spaces. Finally, the vertices of each face $f \in F$ are re–indexed to generate the faces of the two resulting parts $B^+$ and $B^-$. In the context of TI assemblies, clipping means searching for the parameters of a plane that removes just the necessary section from a piece such that it becomes valid.

In general, clipping free–form polygon meshes reduces to clipping triangular facets (Paul Bourke [46]). It is also a procedure offered in traditional geometric packages (e.g., CGAL [47], Geometric Tools [48]), and it is an active research problem for general cases (Scherzinger *et al.* [49]).

### 4.4.2 Exact Clipping

The geometries from overlapping pieces contain the information for the clipping planes that make such pieces valid concerning one another. Two overlapping pieces $B_i = \{V_i, F_i\}$ and $B_k = \{V_k, F_k\}$ (example shown in Figure 4.6(a)) are associated to tiles $f_i, f_k \in F$ from the surface tessellation. The tiles have normal vectors $N_i$ and $N_k$ respectively. The overlapping section $B_i \cap^* B_k$ occurs either above or below such tiles. Vertices $v_i \in V_i$ and $v_k \in V_k$ (Figure 4.6(b)) are the closest vertices between the pieces located at the same the half spaces defined by the planes that contain $f_i$ and $f_k$ respectively. Such a pair of vertices is unique due to the orientation both $B_i$ and $B_k$ must have in order to be in Topological Interlocking. If vertices $v_i, v_k$ are in the respective positive half spaces then $\exists f_k^d \in F_k$ incident to $v_k$ such that its normal vector $N_k^d$ points in opposite direction to $N_k$. Otherwise, $\exists f_k^d \in F_k$ incident to $v_k$ such that its normal vector $N_k^d$ points in the same direction to $N_k$.

The plane $P_k = \text{plane}(N_k^d, v_k)$ clips the section from $B_i$ that makes it not valid regarding $B_k$. Similarly, a plane $P_i = \text{plane}(N_i^d, v_i)$ clips the section from $B_k$ that makes it not valid regarding $B_i$, where $N_i^d$ is the normal vector from a face incident to $v_i$ such that its direction points accordingly concerning $N_i$ based on the half space where $v_i$ is located. Figure 4.6(c) shows planes $P_i, P_k$ clipping $B_k, B_i$ thus validating them.



(a)

(b)

(c)

(d)

Figure 4.6. Two–hop piece clipping. a) Original pieces (overlapping), b) Clipping planes, c) Overlapping sections removed, d) Clipped Pieces.

Figure 4.6(d) shows valid pieces $B'_i, B'_k$ after clipping regarding one another. Although clipping only one of the pieces solves the overlapping problem, the non–clipped piece will have additional contact with more than one piece along a single face. Using the example in Figure 4.6, only clipping $B_k$ leaves $B_i$ in contact with both $B_j$ and $B'_k$ along $B_i$'s down motion. The situation does not violate the Topological Interlocking principle but may affect the static equilibrium of the structure (discussed in Chapter 6).

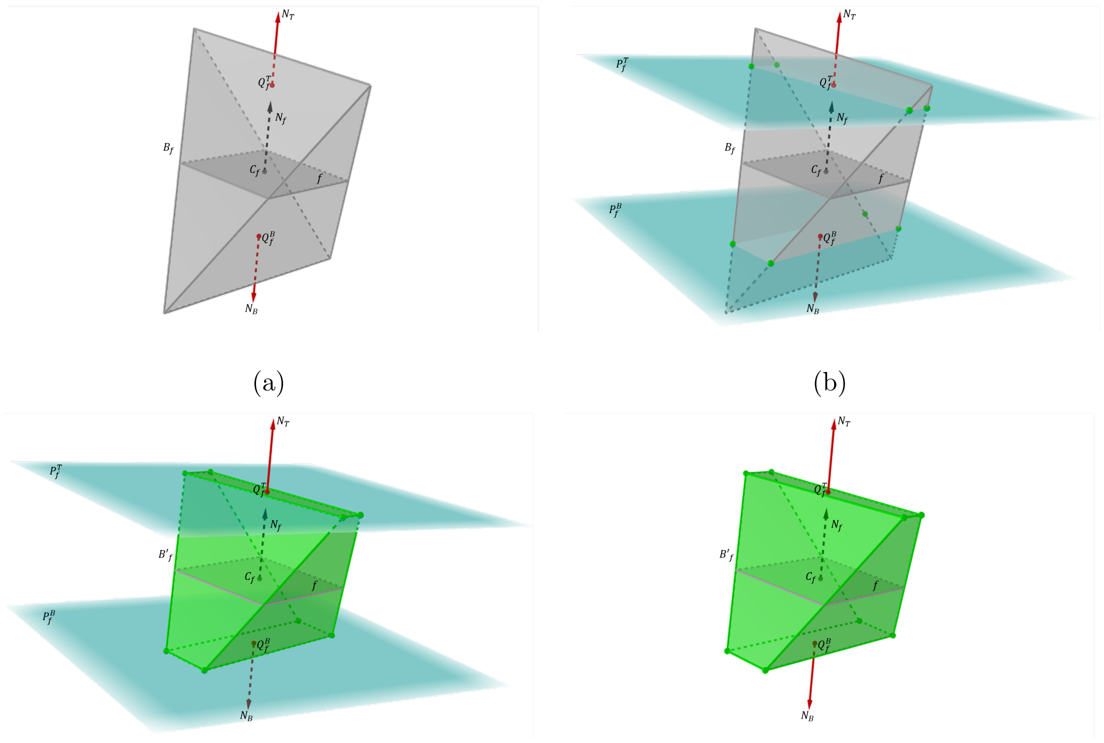
This clipping procedure removes the exact section that makes a piece not valid concerning another one. However, there is no guarantee that such clipping makes a piece to align regarding its neighbors, nor reduce the shape bulkiness caused by both $M$ and $\Psi$ using a generation method $G$.

### 4.4.3 Tile Offset Clipping

The tiles of the surface tessellation provide information for the clipping planes that maintain piece alignment. By offsetting the plane (along its normal vector) that contains a tile, we could define planes to clip the corresponding piece. The clipped sections will align with the orientation of such tile.

A piece $B_f$ is associated to a tile $f \in F$ from the surface tessellation. Let $N_f$ and $C_f$ be the normal vector and a center point from $f$ respectively. The plane $P_f^T = \text{plane}(N_f, Q_f^T)$, with reference point $Q_f^T = C_f + \lambda N_f, \lambda \in \mathbb{R}^+$, clips the section of $B_f$ that exceeds a top offset of $f$. Similarly, plane $P_f^B = \text{plane}(-N_f, Q_f^B)$, with reference point $Q_f^B = C_f - \lambda N_f, \lambda \in \mathbb{R}^+$, clips the section of $B_f$ that exceeds a bottom offset of $f$. Then, $B_f' = \text{TOC}(B_f, \lambda)$ is the resultant piece after applying tile offset clipping on $B_f$ with parameter $\lambda$. Figure 4.7 shows the elements involved in the face offset clipping of $B_f$ into $B_f'$.

Parameter $\lambda$ grants control over the offset clipping planes, it could be uniform for both directions along $N_f$ or relaxed into $\lambda_f^T, \lambda_f^B$ for non–uniform offset clipping along the exterior and interior of tile $f$ respectively. Then, $B_f' = \text{TOC}(B_f, \lambda_f^T, \lambda_f^B)$ is non–uniform offset clipped, where $\lambda_f^T, \lambda_f^B$ are the offset parameters to clip the exterior and interior of $B_f$ respectively. Figure 4.8 shows an example of a TI assembly with clipped pieces using different offset parameters for the exterior and interior of the assembly. Furthermore, these parameters provide design control to adjust the heights of the pieces, which could result in a reduced amount of material to manufacture the pieces. Tile offset clipping returns assemblies with design similar to the Abeille's vault (see Figure 1.1) found in Fallacara [1].



(a)

(b)

(c)

(d)

Figure 4.7. Elements for Tile Offset Clipping. a) Original piece with vectors and reference points, b) Clipping planes intersecting the piece, c) Clipped piece between clipping planes, d) Clipped piece.

Still, any offset parameters $\lambda_f^T, \lambda_f^B$ do not guarantee a piece $B'_f$ will be valid. A small value could clip the piece more than required, while a large value may not remove its overlapping entirely. Optimal offset parameters should remove only the overlapping sections of the pieces. Regular offset parameters remove more than required or leave some overlapping.

Then, for a given assembly $T$ with $n = |T|$ pieces, the optimal offset parameter values $\Lambda^*$ are found using:

$$\begin{aligned} \underset{\Lambda}{\operatorname{argmax}} \quad & g(T) = \sum_{f=0}^{n-1} \lambda_f^T + \lambda_f^B \\ \text{s.t.} \quad & \operatorname{vol}(\operatorname{TOC}(B_i, \lambda_i^T, \lambda_i^B) \cap^* \operatorname{TOC}(B_j, \lambda_j^T, \lambda_j^B)) = 0,\ \forall B_i, B_j \in T, i \neq j \\ & lb \leq \Lambda \end{aligned} \tag{4.4}$$

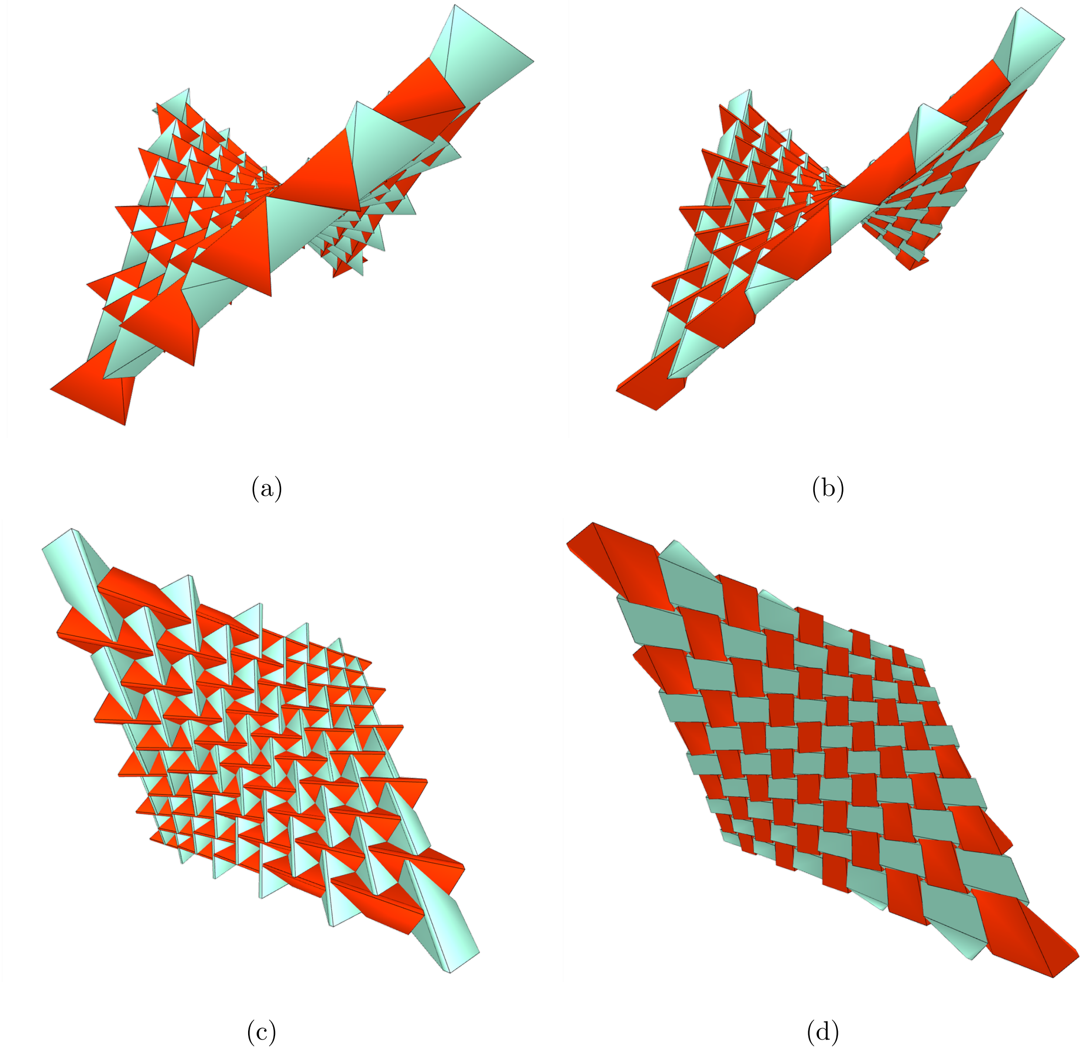


Figure 4.8. TI assembly based on a 10 × 10 saddle surface tessellation, a) Original assembly, b) Clipped pieces using different tile offset clipping parameters, c) Top view, d) Bottom view.

where $\Lambda = \{\lambda_f^T, \lambda_f^B \mid \forall f \in F\}$ is the set of offset parameters for all tiles in the surface tessellation and $lb$ is the lower bounds vector for $\Lambda$.

### 4.4.4 Adaptive Tile Offset Clipping

Given a piece in an assembly, the geometries from its neighboring pieces contain useful information to define the corresponding offset clipping planes. Starting with a center point from the respective tile of a piece, we consider the distances between the point and both top and bottom sections of the piece. The parameters of a clipping plane are a function over such distances from the neighborhood of a piece.

Each piece $B_f$ has two distances from $C_f$ to its respective top and bottom sections. Let $\lambda_f'^T$ be the distance in a piece $B_f$ from $C_f$ to the intersection point between $B_f$ and the ray $C_f + \lambda_f'^T \hat{N}_f$, where $\hat{N}_f$ is the normalized normal vector from tile $f \in F$. Similarly, let $\lambda_f'^B$ be the distance in piece $B_f$ from $C_f$ to the intersection point between $B_f$ and the ray $C_f - \lambda_f'^B \hat{N}_f$ (examples in Figure 4.9).

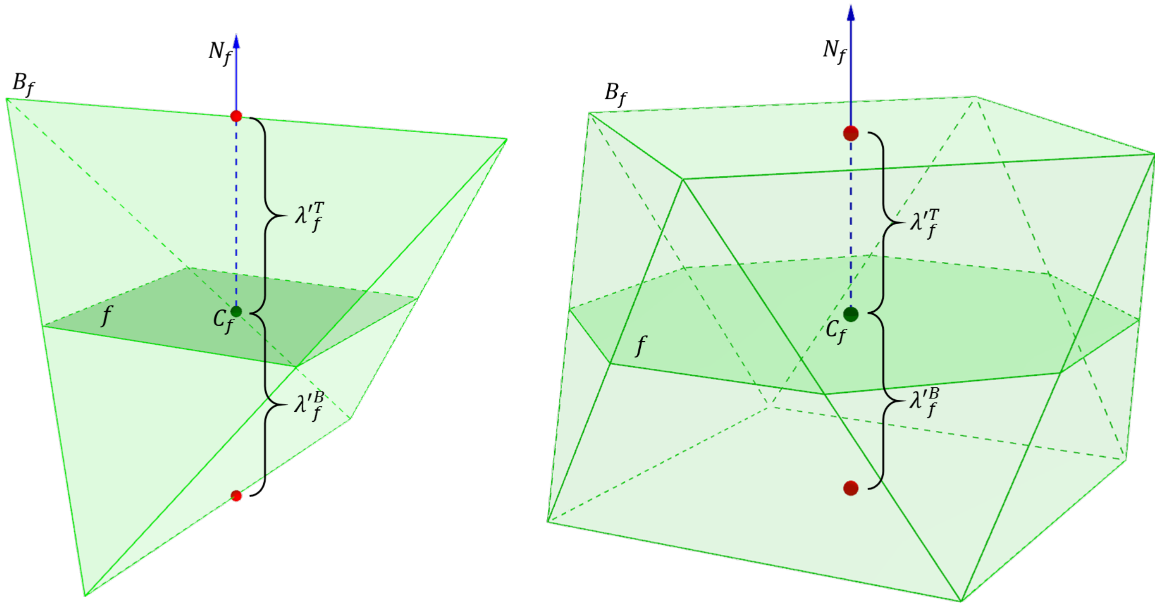
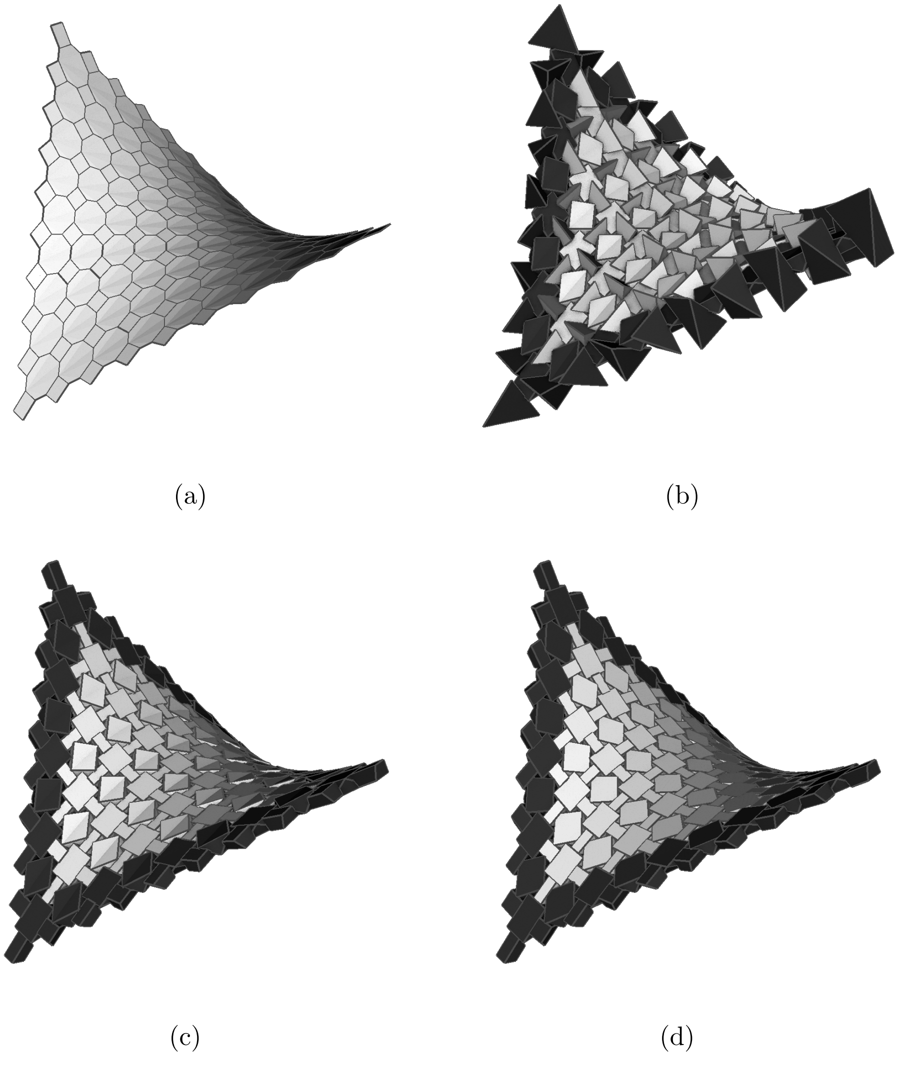


Figure 4.9. Examples of top and bottom distances on a tetrahedron and an antiprism.

The search space for $\lambda_f^T, \lambda_f^B$ reduces to the top and bottom distances from the neighboring pieces of $B_f$. The neighborhood of a piece $B_f$ is the subset of pieces

$T_f \subset T$ such that $\forall B_i \in T_f$ the associated tiles $i, f \in F$ are adjacent in the surface tessellation $M = \{V, F\}$. We can adapt each tile offset value as $\lambda_f^T = h^T(T_f)$ and $\lambda_f^B = h^B(T_f)$ where $h$ is a function (e.g., avg, min) over the top and bottom distances from the neighborhood of $B_f$ respectively. Then, a clipped piece $B'_f = \text{TOC}(B_f, h^T(T_f), h^B(T_f))$. This approach reduces the amount of search through the tile offset space, as specified in 4.4.



(a)

(b)

(c)

(d)

Figure 4.10. Adaptive offset clipping examples. a) Saddle surface tessellation composed of quadrilaterals and octagons, b) Original TI assembly with tetrahedra and 8–sided antiprisms, c) Clipped assembly using $h = \text{avg}$, d) Clipped assembly using $h = \text{min}$.

The adaptive face offset clipping approach becomes useful when the tiles of the surface tessellation have different shapes. In such a case, finding appropriate generation parameters that maintain the alignment of the pieces may be an expensive task. Figure 4.10 shows a TI assembly based on a saddle surface tessellation composed of quadrilaterals and octagons. Both TAM and HBM generation methods return different polyhedra per tile shape. A square tile corresponds to a tetrahedron. A hexagon tile corresponds to an 8–sided antiprism. The evolution sections of each block are at different distances from the respective tile. Such sections come from the intersection of the incident planes from the tile (as discussed in Chapter 3). Then, each piece will have different top and bottom distances regarding its neighborhood. The function $h$ returns a tile offset value that adapts $B_f$ in terms of its neighborhood $T_f$.

## 4.5 Shape Fit Method

We propose a TI generation method to generate a valid assembly when the surface tessellation is non–planar and composed of square tiles; the resultant pieces do not require clipping. The method generates a regular tetrahedron per tile and “fits” the shape of the subset of pieces that overlap due to the dihedral angles between tiles. We name the fitted pieces Quasi–Tetrahedra.

### 4.5.1 Piece Generation

Let $M = \{V, F\}$ be a surface tessellation with a tile $f_i \in F, i = 0, 1, 2, ..., n-1$ a square of side length $l$. Each tile is assigned a binary value $v(f_i) \in \{g, b\}$ such that two adjacent tiles have different values (i.e., checkerboard pattern). Also, $f_i$ has center point $C_i$ and normal vector $N_i$ pointing at the exterior side of the surface tessellation. Let $U_i$ and $V_i$ be two tangential vectors of $f_i$ such that $V_i$ points at an adjacent tile $f_j$ with dihedral angle $\theta_{ij} = 0$, and $U_i = V_i \times N_i$, then $U_i$ points at an adjacent tile $f_k$ where $\theta_{ik} > 0$. The distance from any $C_i$ to both front and back line segments of the respective tetrahedron is $\lambda = \frac{l}{\sqrt{2}}$. The coordinates of the regular tetrahedron $T_i$ associated to tile $f_i$ are:

$$
\begin{aligned}
T_i^A &= \begin{cases} C_i + \lambda N_i + lU_i & , v(f_i) = g \\ C_i + \lambda N_i + lV_i & , v(f_i) = b \end{cases} \\
T_i^B &= \begin{cases} C_i + \lambda N_i - lU_i & , v(f_i) = g \\ C_i + \lambda N_i - lV_i & , v(f_i) = b \end{cases} \\
T_i^C &= \begin{cases} C_i - \lambda N_i + lV_i & , v(f_i) = g \\ C_i - \lambda N_i + lU_i & , v(f_i) = b \end{cases} \\
T_i^D &= \begin{cases} C_i - \lambda N_i - lV_i & , v(f_i) = g \\ C_i - \lambda N_i - lU_i & , v(f_i) = b \end{cases}
\end{aligned}
\tag{4.5}
$$

The generated tetrahedra fit correctly in the direction of $V$ vectors, pieces from tiles with $v(f_i) = b$ overlap in the direction of $U$ vectors. Figure 4.11 shows four regular tetrahedra designed using 4.5, pieces $T_1$ and $T_3$ overlap at the interior of the surface tessellation. Additionally, such pieces do not fit along the direction of $V$ vectors (tangential to the surface tessellation). Then $T = T^g \cup T^b$ has two subsets: one with valid pieces (associated to tiles with $v(f_i) = g$) and one with not valid pieces (associated to tiles with $v(f_i) = b$). Pieces in $T^b$ are fitted within its adjacent pieces in order to make them valid. Let $T_i^m = C_i + \lambda N_i$, $T_i^p = \overline{T_i^m, T_i^m + lU_i} \cap \triangle T_{i+1}^B T_{i+1}^C T_{i+1}^D$ and $T_i^q = \overline{T_i^m, T_i^m - lU_i} \cap \triangle T_{i-1}^A T_{i-1}^C T_{i-1}^D$. The vertices of the fitted piece $T_i'$ are:

$$
\begin{aligned}
T_i'^A =& T_i^p + lV_i \\
T_i'^B =& T_i^q + lV_i \\
T_i'^C =& T_i^p - lV_i \\
T_i'^D =& T_i^q - lV_i \\
T_i'^E =& \overline{T_i^C, T_i^D} \cap \triangle T_{i+1}^B T_{i+1}^C T_{i+1}^D \\
T_i'^F =& \overline{T_i^C, T_i^D} \cap \triangle T_{i-1}^A T_{i-1}^C T_{i-1}^D
\end{aligned}
\tag{4.6}
$$

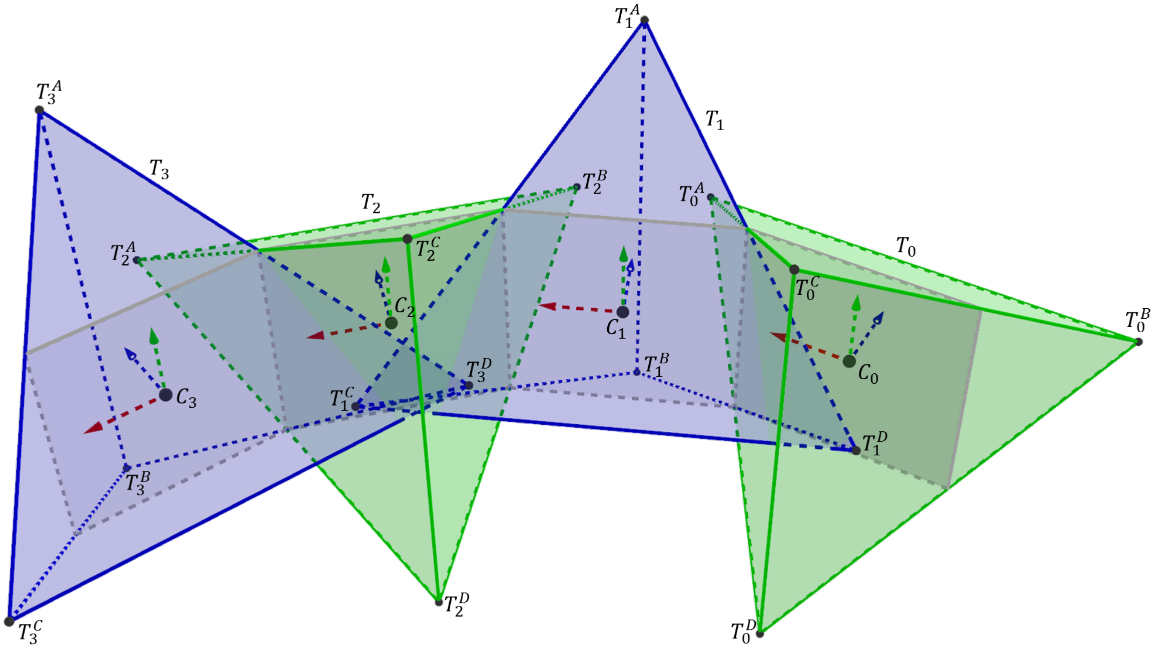


Figure 4.11. Regular tetrahedra from respective squares. Pieces $T_1$ and $T_3$ do not fit concerning $T_0$ and $T_2$, and they overlap to each other. $U$, $V$ and $N$ vectors are shown in red, green and blue respectively.

where $\overline{A, B}$ is the line segment between points $A$ and $B$, and $\triangle ABC$ is the triangle defined by points $A$, $B$, and $C$.

We name the resultant shape Quasi–Tetrahedron. Figure 4.12 shows the elements involved that transform a not valid tetrahedron into a quasi–tetrahedron. The name Quasi–Tetrahedron implies the shape is almost a tetrahedron (it is actually a pentahedron). The shape resembles a tetrahedron as the surface tessellation is flattened. That is, the angle between two adjacent vectors $U_i$ and $U_{i+1}$ gets closer to zero.

### 4.5.2 Cylindrical Assembly

We start with a checkerboard embedded in a cylinder such that each ring of the resultant surface tessellation has an even number of squares. Let $r \in \mathbb{R}^+$ be the radius of the cylinder, and $2m, m \in \mathbb{N}^+$ be the number of squares per ring. The side length

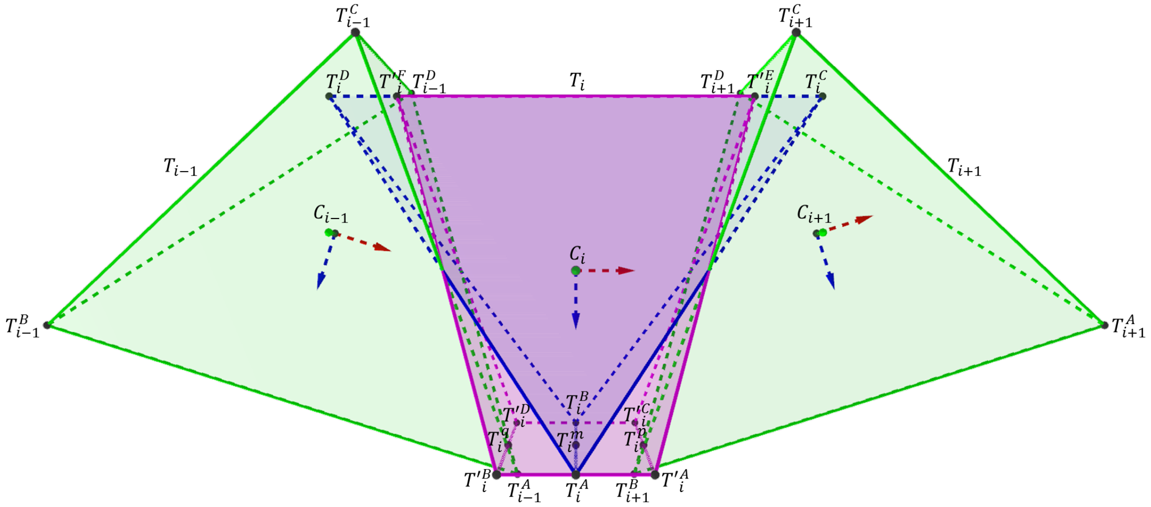


Figure 4.12. Fitting piece $T_i$ (blue) into a Quasi–Tetrahedron $T'_i$ (magenta). $U$, $V$ and $N$ vectors are shown in red, green and blue respectively.

of each square tile is $l = 2r\sin(\pi/2m)$. Then, for $n$ rings the cylinder has height $h = 2nr\sin(\pi/2m)$.

The Topological Interlocking principle holds when $m \geq 6$. In such cases, the intersection line between the left and right faces of a quasi–tetrahedron occurs in front if its respective tile. This line represents the pinch inflicted by the neighboring regular tetrahedra along the ring. When $m < 6$, the intersection line occurs behind the tile, which indicates the neighboring pieces do not lock the pull motion of the quasi–tetrahedron.

Figure 4.13 shows a cylindrical TI generated using the Shape Fitting Method. Such assembly requires two support frames, one at each end of the cylinder. The bottom frame works as the base that aligns the lowest pieces of the assembly. The top frame is the cap that maintains the top pieces closer to each other. Both frames need to be connected internally (e.g., a beam or a stiff spring) such that they become a single structure.

We tested the stability of the structure by using both a 3D printed prototype and physical simulations for general cases. The cylindrical assembly is stable for a small number of rings ($n \leq 4$). However, it becomes unstable with larger values regardless of the radius and number of pieces per ring. We discuss the stability of cylindrical TI assemblies in Chapter 6.

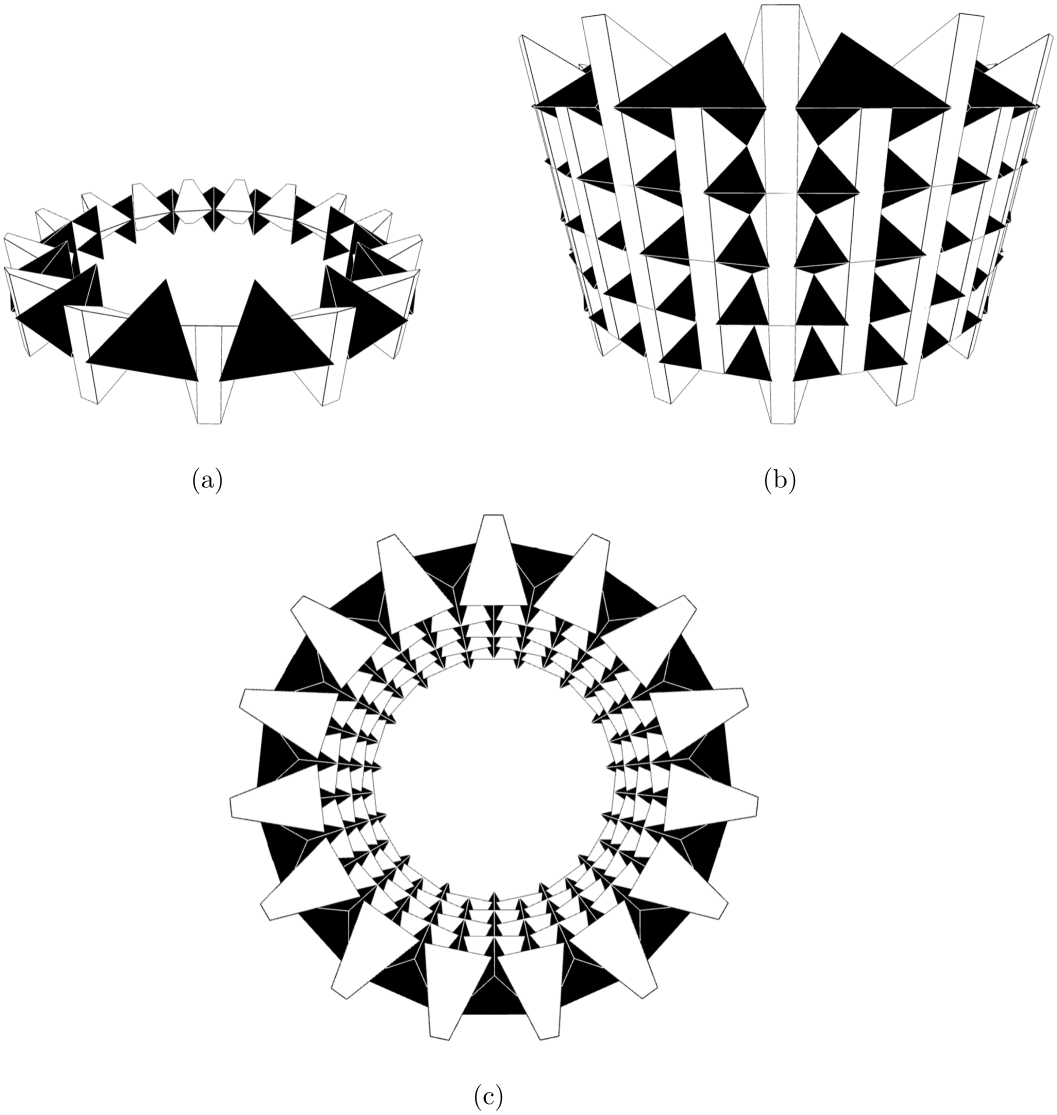


Figure 4.13. Cylindrical TI assembly with 30 pieces per ring, regular tetrahedra in black, quasi–tetrahedra in white. a) One ring, b) Five rings, c) Top view

## 4.6 Discussion

We have introduced a search method for feasible generation parameters $\Psi$ for a surface tessellation $M$ using a generation method $G$. A resultant assembly T $=$ $G(M, \Psi^*)$ has minimized overlapping sections between the pieces. The assembly is valid if $\Psi^*$ is feasible. Still, there are situations when the search procedure does not return a feasible parameter set, then clipping the overlapping from $T$ makes it valid. Our clipping approaches use the geometry of the pieces and the surface tessellation as search spaces for the parameters of the clipping planes. In all cases, clipping is a post–processing step after generating the assembly.

The Shape Fit generation method returns a valid assembly in a single iteration over the surface tessellation. The method requires all tiles to be squares. The generated assembly is a closed form of the surface tessellation where each squared tile has side length $l$. This generation method returns two types of pieces: regular tetrahedron and quasi–tetrahedron. The name quasi–tetrahedron indicates the shape is not precisely a tetrahedron but resembles it as the dihedral angles between their respective tiles gets closer to zero.

# 5 GENERAL MID–SECTION EVOLUTION

## 5.1 Introduction

The Topological Interlocking principle states that two blocks are interlocking if their respective top and bottom evolution sections (formally known as cross–sections) degenerate to a line segment or a point. Dyskin *et. al.* [4] [3] defined this principle, and Kanel–Belov *et. al* [2] later formalized it. We discussed these in Section 2.2.2. The Platonic Solids satisfy such principle since we can find polygons that describe their mid–sections with respective evolution as cross–sections.

Despite such property, the generation methods described so far can only generate three out of five Platonic solids: the tetrahedron, the hexahedron, and the octahedron. Nevertheless, there exist other 3D shapes with TI properties that both TAM and HBM cannot generate (e.g., truncated Platonic solids and the Quasi–tetrahedron). Designers and engineers have to modify the blocks (either manually or computationally) to have more extensive catalogs of TI assemblies.

Our contribution in this chapter is a generalization of the mid–section evolution concept that allows the design of any block with TI properties. We approach the method by including evolution steps for a given polygon such that its evolution along a direction vector produces the vertices of the corresponding block. Each step requires a collection of angle and distance parameters that guide the polygon shape from one evolution step to another. The evolution parameters are similar in nature and purpose to the generation parameters for the TAM and HBM generation methods described in Chapter 3. Additionally, we can obtain identical results to the clipped assemblies introduced in Chapter 4 using specific evolution step parameters.

## 5.2 Overview

The general mid–section evolution method expands the TI generation to infinitely many interlocking block shapes provided the tiles have an even number of sides, and their edges have alternating direction values. We discussed these requirements in Section 3.3.1. We propose a method that generates a polyhedron (i.e., block) $B$ based on a polygon (i.e., tessellation tile) $f$ using evolution step parameters. Such parameters contain a set of tilting angles and a scalar value that guides the evolution process. An evolution step translates and reshapes a polygon into a $n$–polytope (i.e., a point, a line segment or a polygon). The vertices from all evolved $n$–polytopes contain the geometry of the resultant polyhedron. The tilting angle values between two consecutive evolution steps determine whether the polyhedron is either convex or concave.

We first explain the evolutionary step that transforms a polygon into a $n$–polytope. Then, we describe the generation of a polyhedron as the collection of vertices from sequences of multiple consecutive evolution steps. Additionally, we introduce the requirements for such evolution steps to generate polyhedra that satisfy the TI principle. Finally, we show how to generate the Platonic solids and their truncated versions using the general mid–section evolution method starting from their corresponding mid–section polygons.

## 5.3 Polygon Evolution

In this section, we describe the generation of a polyhedron from a polygon using multiple evolution steps. An evolution step is a sweeping procedure that evolves (i.e., reshapes and translates) a polygon into a $n$–polytope, where $n \in \{0, 1, 2\}$ (i.e., a point, a line segment or a polygon respectively). This method is a sweep plane algorithm along a direction vector. The vertices from a sequence of evolved $n$–polytopes are the vertices of the resultant polyhedron. The continuous motion of the polygon as it evolves to the respective $n$–polytope describes the edges of the polyhedron.

### 5.3.1 Evolution Step

Let $f = \{V, E\}$ be a polygon of $n \in \mathbb{N}^+, n \geq 3$ sides, where $V = \{v_0, v_1, ..., v_{n-1}\}$ is a set of vertices and $E = \{(0, 1), (1, 2), ..., (n-2, n-1), (n-1, 0)\}$ is the set of tuples with the vertex indices that define the edges of $f$. The polygon has centroid point $C$ and normalized normal vector $\hat{N}$. We assume $V$ and $E$ walk around $f$ in counterclockwise order to $C$ and $\hat{N}$. Each edge $e = (i, j) \in E$ has a normalized direction vector $\hat{K}_e = \frac{v_j - v_i}{||v_j - v_i||}$ and a midpoint $M_e = \frac{v_i + v_j}{2}$.

An evolution step requires two types of parameters: tilting angles associated with the edges of the polygon, and a scalar value that determines the maximum length allowed for the step. Let $\Theta = \{\theta_e, \forall e \in E \mid \theta_e \in [-\frac{\pi}{2}, \frac{\pi}{2}]\}$ be a set of tilting angles, and $\lambda \in \mathbb{R}^+$ be a scalar value. A tilting angle $\theta_e$ rotates $\hat{N}$ using $\hat{K}_e$ as the rotation axis vector, the rotated vector $\hat{N}_e$ is the normal vector of a plane that contains $e$. Then, $\hat{N}_e = \text{rotate}(\hat{N}, \hat{K}_e, \theta_e)$ where $\text{rotate}(V, A, \theta)$ is the Axis–Angle rotation (as stated in section 3.3.2). The plane $P_e = \text{plane}(\hat{N}_e, M_e)$ passes through $e$. The intersection between the planes from the two edges incident to a vertex $v_i \in V, \forall i = 0, 1, ..., n-1$ defines a line $L_i$; therefore, $L_i$ contains $v_i$. Finally, a point $D = C + \lambda\hat{N}$, and planes $P_C = \text{plane}(\hat{N}, C)$ and $P_D = \text{plane}(\hat{N}, D)$. Figure 5.1 shows an example of the mentioned elements on a hexagon.

We consider the intersection points $v'_{i,j} = L_i \cap L_j, \forall (i, j) \in E$ that occur within the space section delimited by planes $P_C$ and $P_D$. A point $v'_{i,j}$ is within evolution range if $v'_{i,j} \in P_C^+ \cap (P_D^- \cup P_D)$. That is, $v'_{i,j}$ lies at both the positive half–space defined by plane $P_C$ in direction $\hat{N}$ and the negative half–space defined by plane $P_D$ in opposite direction to $\hat{N}$ or $P_D$ itself. Otherwise, we say $v'_{i,j}$ is out of evolution range. In such case we consider the intersection points $v'_{i,D} = L_i \cap P_D$ and $v'_{j,D} = L_j \cap P_D$ which are within evolution range. Figure 5.2 shows an evolved $n$–polytope from the intersection point located at plane $P_D$. Let $V'$ be the set of intersection points that are within evolution range; then, $V'$ defines the endpoints of the evolved $n$–polytope $f'$. Let $E'$ be the set of edges of $f'$, each tuple $e' \in E'$ is defined by the indices of consecutive

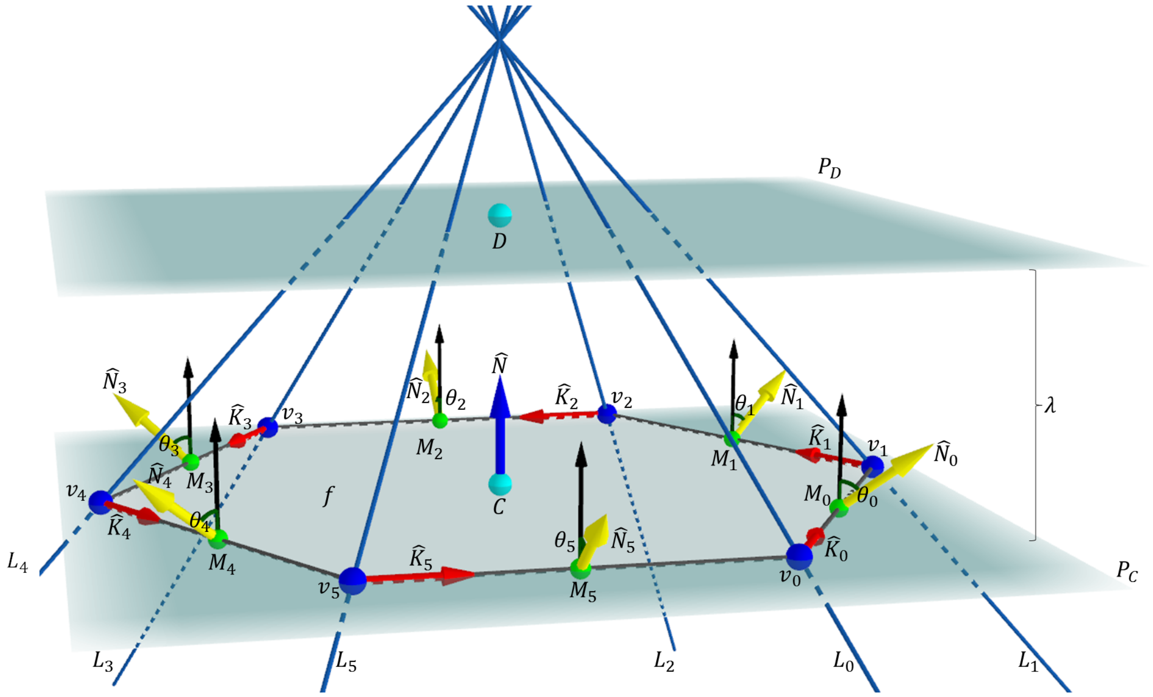


Figure 5.1. Evolution step elements for a hexagon.

vertices in $V'$ as they are calculated. The last tuple in $E'$ must connect both the last and the first vertices in $V'$. We call $f' = \text{Evolve}(f, \Theta, \lambda)$ to the described procedure that evolves $f = \{V, E\}$ into $f' = \{V', E'\}$. We refer to $f$ as the seed polygon when used as a parameter in an evolution step.

The cardinality of $V'$ determines the type of $n$–polytope of $f'$. Polytope $f'$ is a single point if $|V'| = 1$, a line segment if $|V'| = 2$, or a polygon with $n' = |V'|$ sides if $|V'| \geq 3$. When $|V'| > 3$ we need to check all points in $V'$ are coplanar. If at least one point $v'_i \in V'$ is not coplanar then $f'$ is a degenerated evolution of $f$ (i.e., the sequence of edges $E' = \{(0, 1), (1, 2), ..., (n' - 2, n' - 1), (n' - 1, 0)\}$ does not describe a planar polygon). In such a case, adjusting either $f$, $\Theta$, or $\lambda$ such that the resultant $n$–polytope $f' = \{V', E'\}$ describes either a point, a line segment or a planar polygon solves the issue. Figure 5.3. shows examples of evolved $n$–polytopes from regular seed polygons.

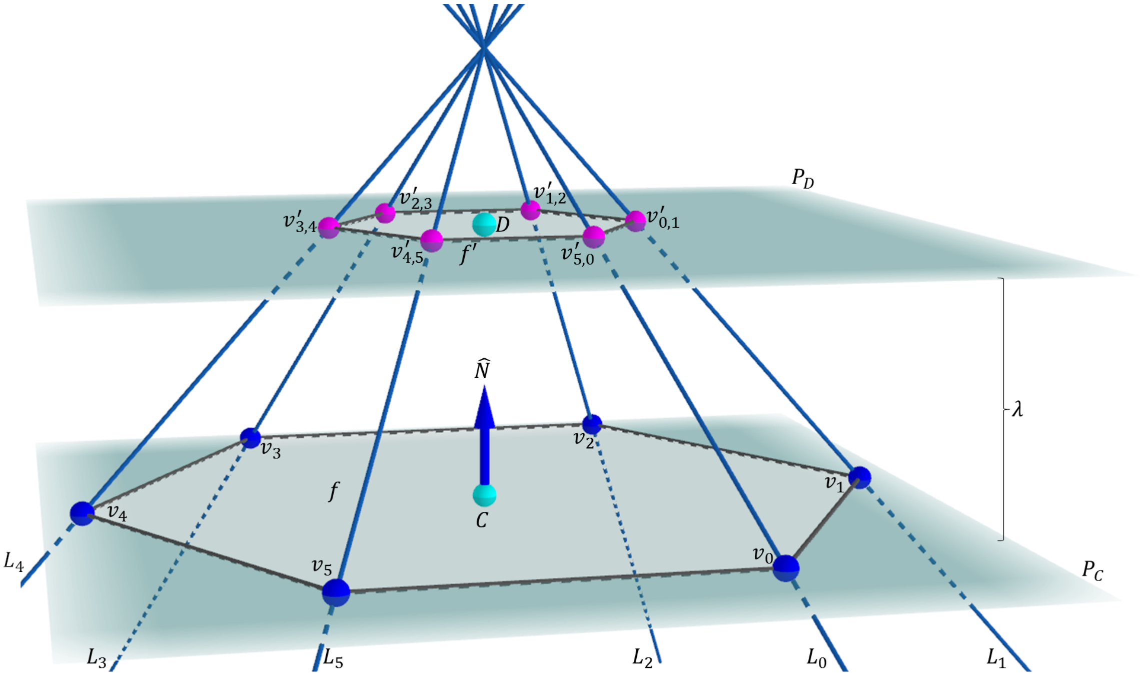


Figure 5.2. $n$–polytope from the intersection points at $P_D$.

Table 5.1.
Prismatoid families based on the original vertices $V$, evolved vertices $V'$ and tilting angle parameters $\Theta$ using a single evolution step.

| Family | Vertices | Angles |
|---|---|---|
| Pyramids | $\lvert V\rvert \geq 3, \lvert V'\rvert = 1$ | $\theta > 0, \forall\theta \in \Theta$ |
| Wedges | $\lvert V\rvert \geq 3, \lvert V'\rvert = 2$ | $\theta > 0, \forall\theta \in \Theta$ |
| Parallelepipeds | $\lvert V\rvert = \lvert V'\rvert = 4$ | $\theta_0 = \theta_2 = 0, \theta_1 = \theta_3$ |
| Prisms | $\lvert V\rvert = \lvert V'\rvert \geq 3$ | $\theta = 0, \forall\theta \in \Theta$ |
| Cupolae | $\lvert V\rvert > \lvert V'\rvert \geq 3$ | $\theta > 0, \forall\theta \in \Theta$ |
| Frusta | $\lvert V\rvert = \lvert V'\rvert \geq 3$ | $\theta > 0, \forall\theta \in \Theta$ |

An evolution step generates the geometry of a prismatoid whose vertices are $\{V \cup V'\}$. Such set of vertices along with the set of tilting angles $\Theta$ describe the type of

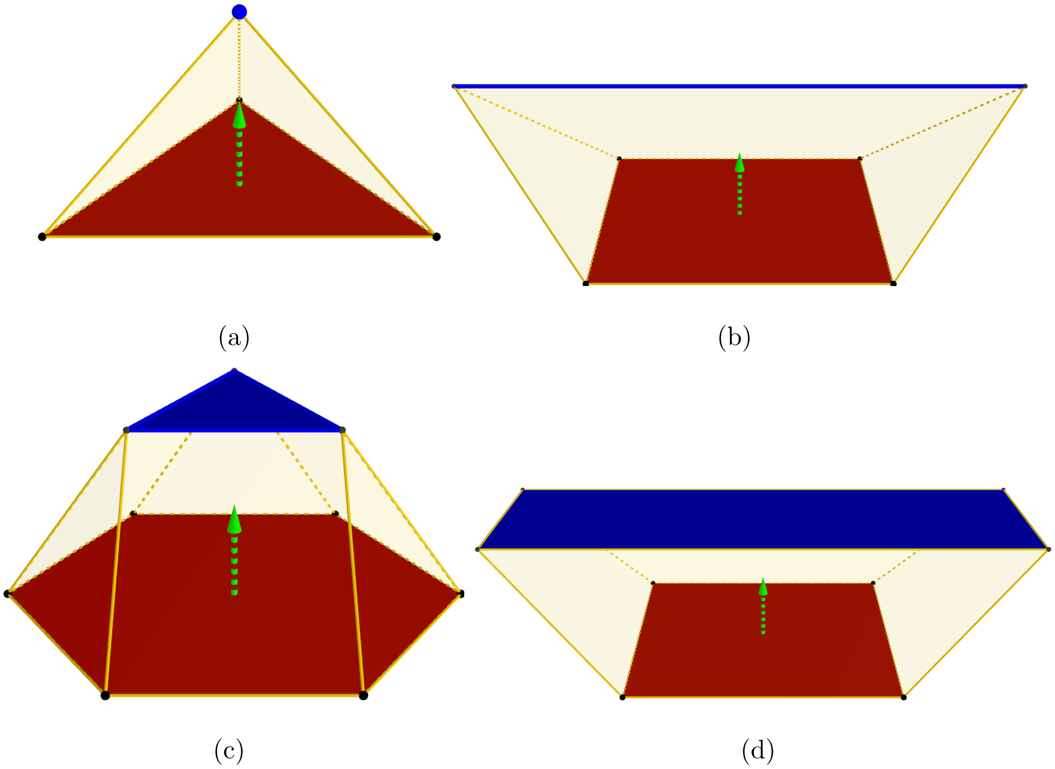


Figure 5.3. Evolved $n$–polytopes from polygons. Seed polygons in red, evolved $n$–polytopes in blue. a) Point, b) Line Segment, c) Polygon with less sides than the seed polygon, d) Polygon with the same number of sides as the seed polygon.

the generated prismatoid. Table 5.1 shows the requirements in terms of vertices and angles to describe an specific polyhedron from the prismatoids family.

### 5.3.2 Single Direction Polygon–Polyhedron Evolution

We consider using evolved $n$–polytopes as the seed polygons for subsequent evolution steps. An evolved $n$–polytope $f' = \text{Evolve}(f, \Theta, \lambda)$ where $f' = \{V', E'\}$ with $|V'| \geq 3$ becomes the seed polygon for another evolution step. Such new step requires its own set of evolution parameters $\{\Theta', \lambda'\}$ with $|\Theta'| = |V'|$ (i.e., the number of tilting angles for the new evolution step is equal to the number of sides of $f'$). A new evolved $n$–polytope $f'' = \text{Evolve}(f', \Theta', \lambda')$ is then an additional step in the evolution sequence from the original seed polygon $f = \{V, E\}$ into a $n$–polytope $f'' = \{V'', E''\}$. Considering the set of vertices $\{V, V', V''\}$ from polygons $f, f'$ and $n$–polytope $f''$, such set contains the vertices of a polyhedron described by the two–step evolution of $f$ into $f''$. Figures 5.4 and 5.5 show an example of the elements for the evolution of a polygon into a single point through two evolution steps.

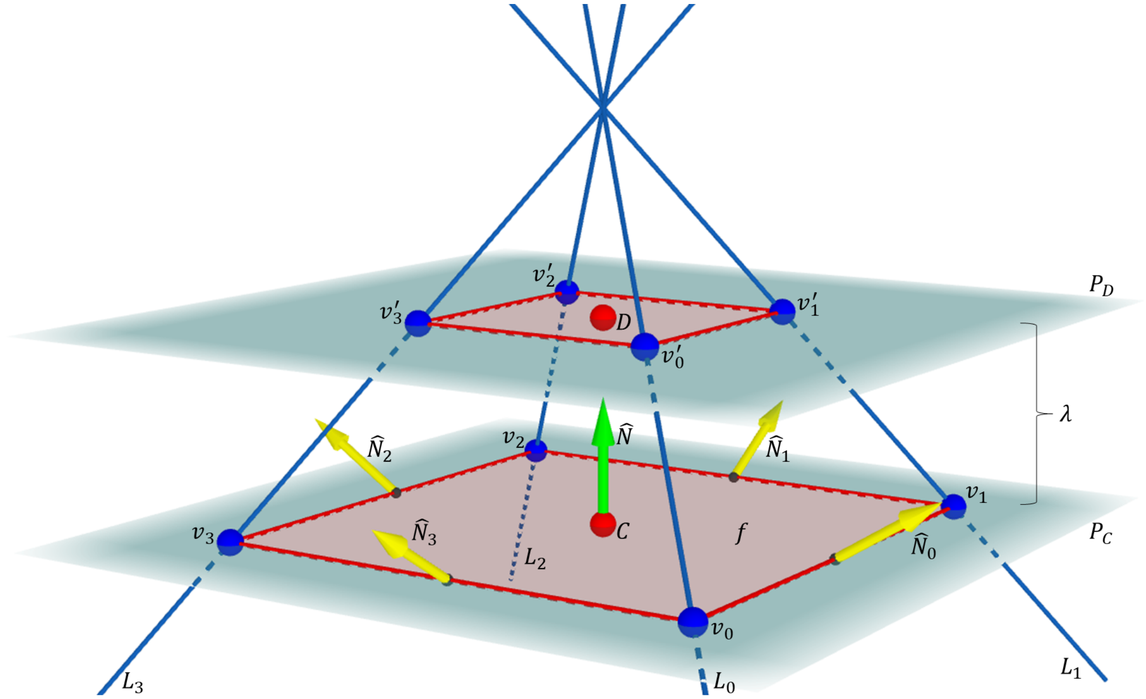


Figure 5.4. Elements for a first evolution step from a square to a single point.

A sequence of at most $m \geq 2$ evolution steps generates a polyhedron $B$ from a polygon $f$. Let $f^0 = f$ be the seed polygon, the sequence of evolved $n$–polytopes $f^{i+1} = \text{Evolve}(f^i, \Theta^i, \lambda^i)$, where $f^i = \{V^i, E^i\} \, \forall i = 0, 1, ..., n-1$ contains the geometric information of $B$. The evolution process stops after $m$ evolution steps or a $n$–polytope $f^i$ is either a point or a line segment. The set $V_B = \bigcup_{i=0}^{m-1} V^i$ is the set of vertices of $B$. The set $F_B$ is the set of vertex indices that define the faces of $B$, The correspondences between the edges of $f^i$ and the respective evolved elements from

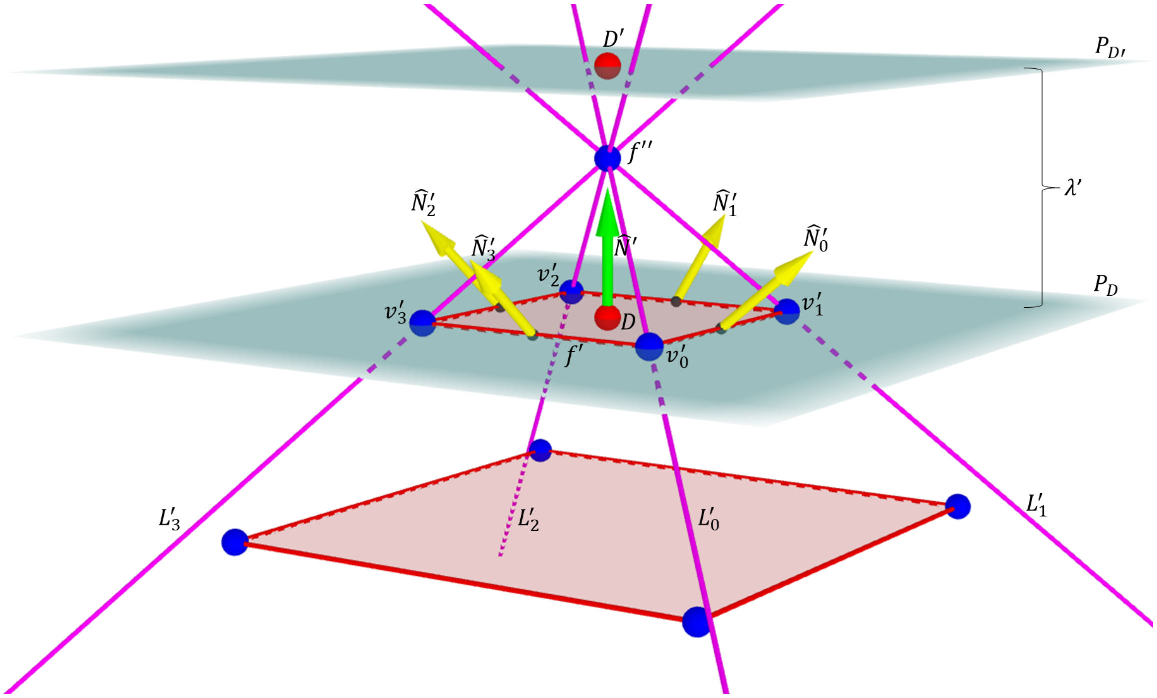


Figure 5.5. Elements for a second evolution step from a square to a single point.

$f^{i+1}$ define $F_B$. Figure 5.6 shows the resultant polyhedron based on the vertices from the seed polygon and evolved $n$–polytopes as shown in figures 5.4 and 5.5.

### 5.3.3 Double Direction Polygon–Polyhedron Evolution

We expand the family of evolved polyhedra by considering negative evolution steps. Let us start with a seed polygon $f$ with a normalized normal vector $\hat{N}$. An additional sequence of evolution steps along with $-\hat{N}$ allows the generation of polyhedra that cannot be described by a single direction polygon–polyhedron evolution. A positive evolution step is one that occurs above $f$ along $\hat{N}$, and a negative evolution step is one that occurs below $f$ along $-\hat{N}$.

Let $f_0^+ = f$ be the seed polygon for the first positive evolution step, the sequence of $m^+ \in \mathbb{N}^+$ positive evolution steps $f_{i+1}^+ = \text{Evolve}(f_i^+, \Theta_i^+, \lambda_i^+)$, $i = 0, 1, ..., m^+ - 1$

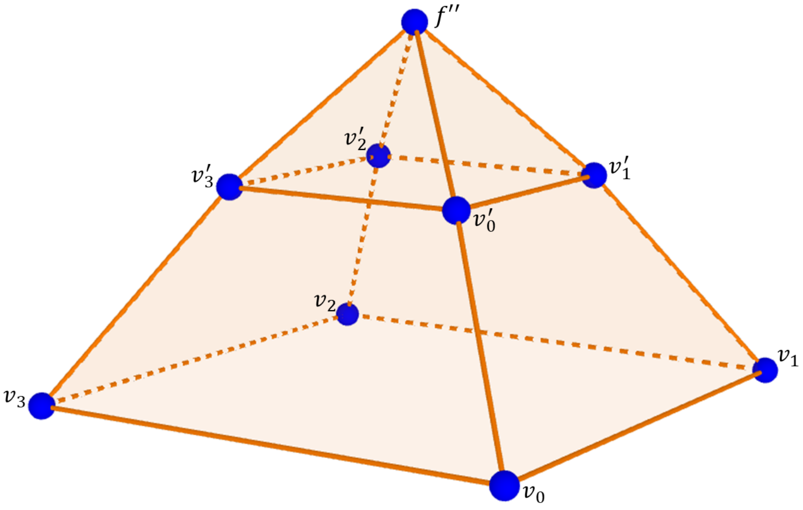
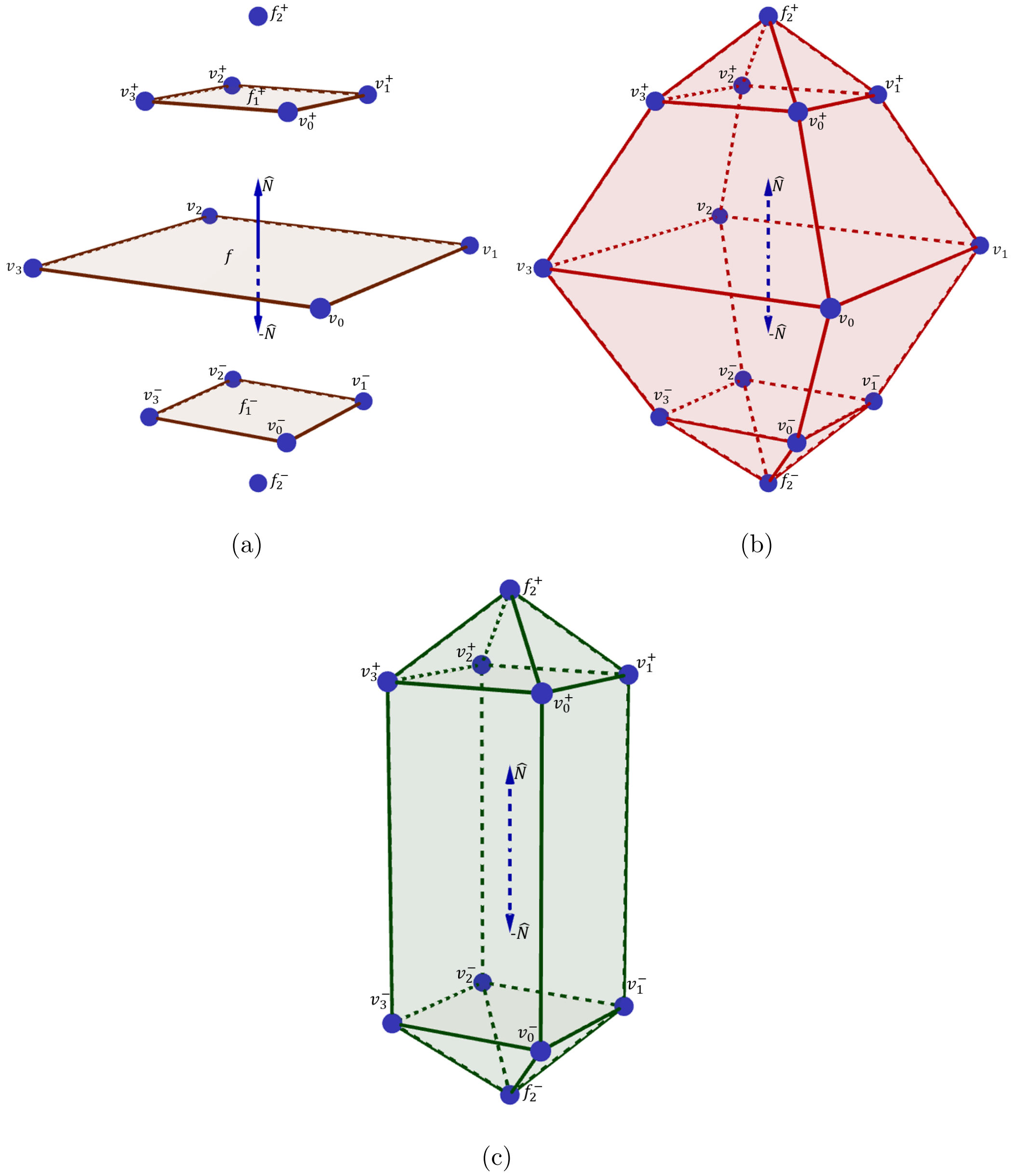


Figure 5.6. Resultant polyhedron based on the vertices from the seed polygon and evolved $n$–polytopes.

describes the upper half of the polyhedron (i.e., the section that lies above $f$ in direction $\hat{N}$). Similarly, let $f_0^- = \text{flip}(f)$ be the seed polygon for the first negative evolution step, where flip($p$) changes the vertex indices order of the edges from a polygon $p$ (i.e., swaps the orientation of the front face of the polygon). Flipping the seed polygon for the first negative evolution step allows the evolution of $f$ along $-\hat{N}$. This adjustment lets us use the evolution step method without any modification for negative evolution steps. The sequence of $m^- \in \mathbb{N}^+$ negative evolution steps $f_{j+1}^- = \text{Evolve}(f_j^-, \Theta_j^-, \lambda_j^-)$, $j = 0, 1, ..., m^- - 1$ describes the lower half of the polyhedron (i.e., the section that lies below $f$ in direction $-\hat{N}$).

The geometry of an evolved polyhedron $B$ comes from the vertices of the seed polygon and the evolved $n$–polytopes along with the positive and negative directions. The vertices from the seed polygon $f = \{V, E\}$, the positive evolved $n$–polytopes



(a)

(b)

(c)

Figure 5.7. Resultant polyhedra from Double Direction Polygon–Polyhedron Evolution from a polygon $f$. a) Seed Polygon and $n$–polytopes from evolution steps, b) Polyhedron described by vertices from $f, f_1^+, f_2^+, f_1^-, f_2^-$, c) Polyhedron described by vertices from $f_1^+, f_2^+, f_1^-, f_2^-$.

$f_i^+ = \{V_i^+, E_i^+\}, \forall i = 0, 1, ..., m^+ - 1$ and the negative evolved $n$–polytopes $f_j^- = \{V_j^-, E_j^-\}, \forall j = 0, 1, ..., m^- - 1$ describe the vertices of the polyhedron $B$ generated from both positive and negative evolution sequences. The set $V_B = V \cup \bigcup_{i=0}^{m^+-1} V_i^+ \cup \bigcup_{j=0}^{m^--1} V_j^-$ contains the vertices of $B$. The set $F_B$ is the set of vertex indices that define the faces of $B$. Similarly to the single direction evolution, the correspondences between the edges of the seed polygons and their respective evolved $n$–polytopes define $F_B$. Figure 5.7(b) shows an example of a polyhedron obtained from evolving a seed polygon $f$ along $\hat{N}$ and $-\hat{N}$.

By omitting the vertices of the seed polygon $f$, we can generate a different polyhedron $B$ using only the information from both evolution sequences. The set $V_B = \bigcup_{i=0}^{m^+-1} V_i^+ \cup \bigcup_{j=0}^{m^--1} V_j^-$ is then the vertices from both positive and negative evolved $n$–polytopes exclusively. Figure 5.7(c) shows an example of a polyhedron obtained from evolving a seed polygon $f$ along $\hat{N}$ and $-\hat{N}$ without including the vertices of $f$.

### 5.3.4 Reciprocal Evolution Steps

We consider rotational symmetries between evolved $n$–polytopes on double direction polygon–polyhedron evolution. Let $f = \{V, E\}$ be a polygon and seed polygons $f_0^+ = f, f_0^- = \text{flip}(f)$ for both positive and negative evolution sequences respectively. A positive evolved $n$–polytope $f_{i+1}^+ = \text{Evolve}(f_i^+, \Theta_i^+, \lambda_i^+), i \in \{0, 1, ..., m^+ - 1\}$ is reciprocal to a negative evolved $n$–polytope $f_{j+1}^- = \text{Evolve}(f_j^-, \Theta_j^-, \lambda_j^-), j \in \{0, 1, ..., m^- - 1\}$ if $|V_i^+| = |V_j^-|$ , $|V_i^+|$ and $|V_j^-|$ are even numbers, and the geometries of $f_{i+1}^+$ and $f_{j+1}^-$ are rotational symmetric to one another of order $O = \frac{|V_i^+|}{2}$. Otherwise, the evolved $n$–polytopes are non–reciprocal to each other. Finally, two sets of parameters $\{\Theta_i^+, \lambda_i^+\}, \{\Theta_j^-, \lambda_j^-\}$ for $i \in \{0, 1, ..., m^+ - 1\}$ and $j \in \{0, 1, ..., m^- - 1\}$ are reciprocal if the respective evolved $n$–polytopes $f_{i+1}^+, f_{j+1}^-$ are reciprocal to each other. Figure 5.8 shows two octahedra generated using both reciprocal and non–reciprocal evolution steps.

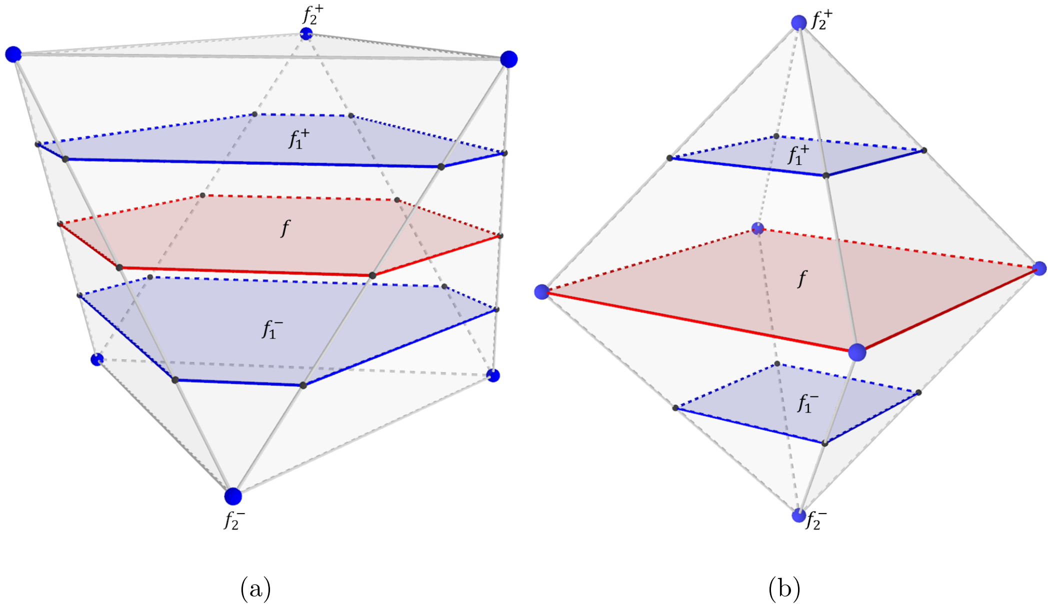
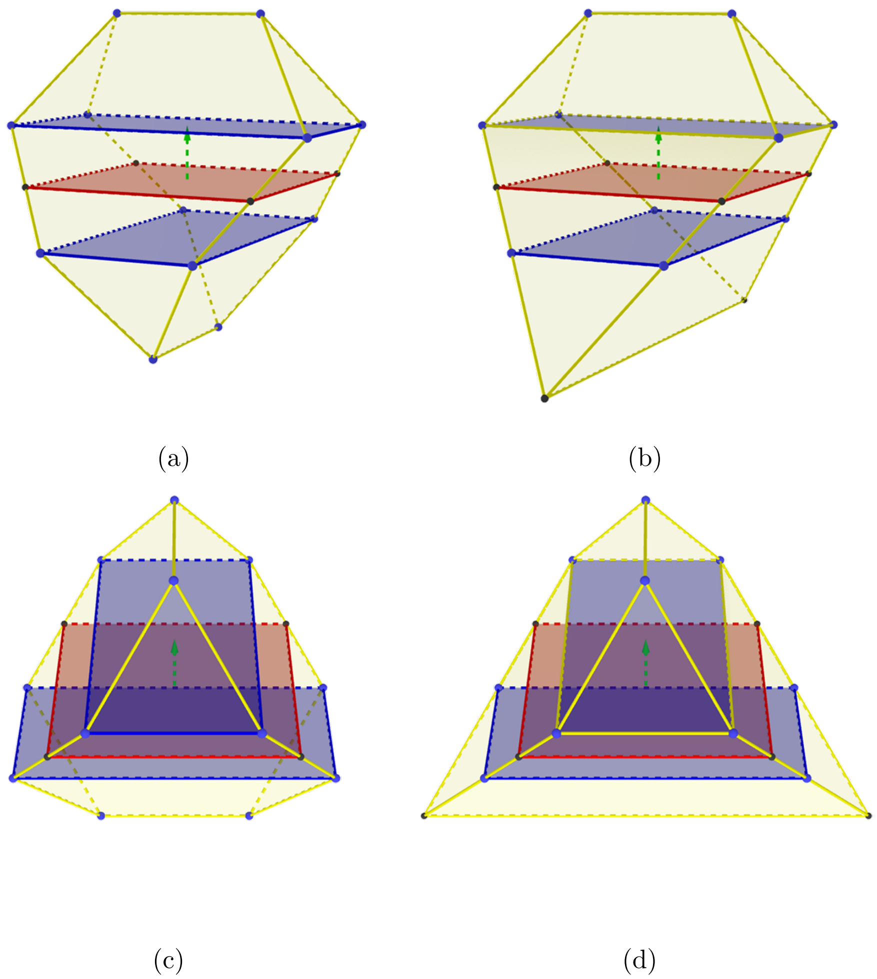


Figure 5.8. Evolved octahedra based on reciprocal and non–reciprocal evolution steps. a) Reciprocal evolution steps $f_1^+$ with $f_1^-$, and $f_2^+$ with $f_2^-$, b) Non–reciprocal evolution steps.

### 5.3.5 Uniform Evolution

The double direction evolution through a sequence of correspondent reciprocal evolution steps between both positive and negative evolution sequences generates a different set of polyhedra. Let $f = \{V, E\}$ be a polygon, seed polygons $f_0^+ = f, f_0^- = \text{flip}(f)$ for both positive and negative evolution sequences respectively, a positive evolution sequence with $m^+$ evolution steps and a negative evolution sequence with $m^-$ evolution steps. Both evolution sequences are uniform evolution of $f$ if $m^+ = m^-$ (i.e., the number of evolution steps for both positive and negative evolution sequences is the same), and the evolved $n$–polytopes $f_{i+1}^+ = \text{Evolve}(f_i^+, \Theta_i^+, \lambda_i^+)$ and $f_{i+1}^- = \text{Evolve}(f_i^-, \Theta_i^-, \lambda_i^-)$ are reciprocal to each other $\forall i = 0, 1, ..., m^+ - 1$. Figure 5.9 shows two polyhedra generated with both uniform and non–uniform evolution. The uniform evolution generates a truncated tetrahedron.



(a)

(b)

(c)

(d)

Figure 5.9. Uniform and non–uniform evolution. Seed polygons in red, evolution polygons in blue. a) Uniform evolution (overview), b) Non–uniform evolution (overview), c) Uniform evolution (lateral), d) Non–uniform evolution (lateral).

## 5.4 General Mid–Section Evolution

We propose an evolutionary approach that defines the blocks of a TI assembly based on a given surface tessellation. Each tile in the tessellation evolves into a polyhedron by a double direction evolution. Although each tile evolves independently from each other, the parameters of at least its first positive and negative evolution steps must match with the respective first step parameters from the neighboring tiles. Such requirement guarantees the evolved polyhedra have common interfaces that allow them to comply with the TI principle (discussed in Section 2.4). The evolved polyhedra are then the blocks of the interlocking assembly.

### 5.4.1 Generation Method

Let $M = \{V, F\}$ be a surface tessellation with $V$ the set of vertices and $F$ the set of tiles. Each tile $f \in F$ must have an even number of sides (this guarantees the edges will have alternating direction values as indicated in Section 3.3.1). A tile $f \in F$ evolves into a polyhedron $B$, which becomes the respective block in the TI assembly. The geometry of $B$ comes from the double direction polygon–polyhedron evolution of $f$ with $f_0^+ = f$ and $f_0^- = \text{flip}(f)$ the respective seed polygons for the positive and negative evolution sequences respectively. Each evolution sequence has a specific number of steps. The positive sequence has $m^+ \in \mathbb{N}^+$ steps while the negative sequence has $m^- \in \mathbb{N}^+$ steps.

An evolution step takes a set of evolution parameters $\Psi_k^d = \{\Theta_k^d, \lambda_k^d\}, \forall k = 0, 1, ..., n^d - 1$ where $d \in \{+, -\}$ is the evolution sequence to which the step belongs to. The set $\Theta_k^d$ contains the tilting angles for the edges of the seed polygon $f_k^d$. Such set must satisfy $|\Theta_k^d| = |V_k^d|$ (i.e., the number of tilting angles for the evolution step must be the same as the number of sides of the respective seed polygon). Additionally, two consecutive angles $\theta_i, \theta_j \in \Theta_k^d$ must comply with $\text{sign}(\theta_i) = -\text{sign}(\theta_j)$ (i.e., tilting angles toggle their directions along the edges of the seed polygon). The toggling directions are equivalent to the edge directions required to generate interlocking blocks using TAM or HBM (discussed in Section 3.3.1). The scalar value $\lambda_k^d \in \mathbb{N}^+$ is the evolution range for the respective evolution step. Each $f_{k+1}^d = \text{Evolve}(f_k^d, \Theta_k^d, \lambda_k^d), \forall d \in \{+, -\}, \forall k = 0, 1, ..., n^d - 1$ defines the vertices and faces of the evolved polyhedron.

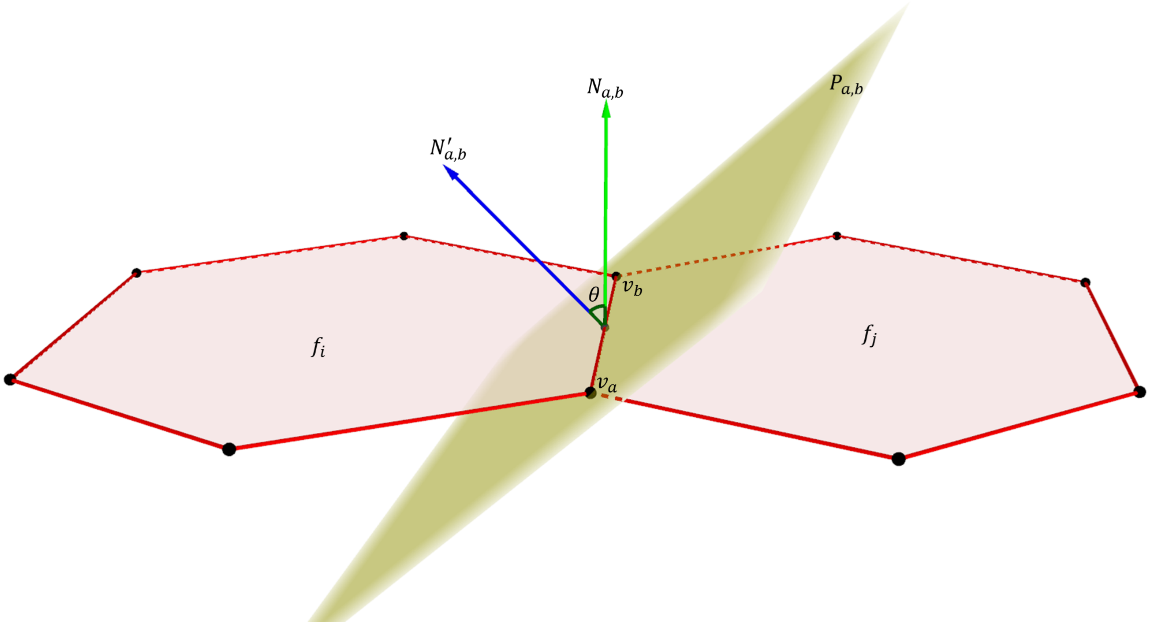


Figure 5.10. Matching angle $\theta_{ba} = -\theta_{ab}$ for the tilted vector associated to the shared edge between two neighboring tiles $f_i, f_j$ in a surface tessellation.

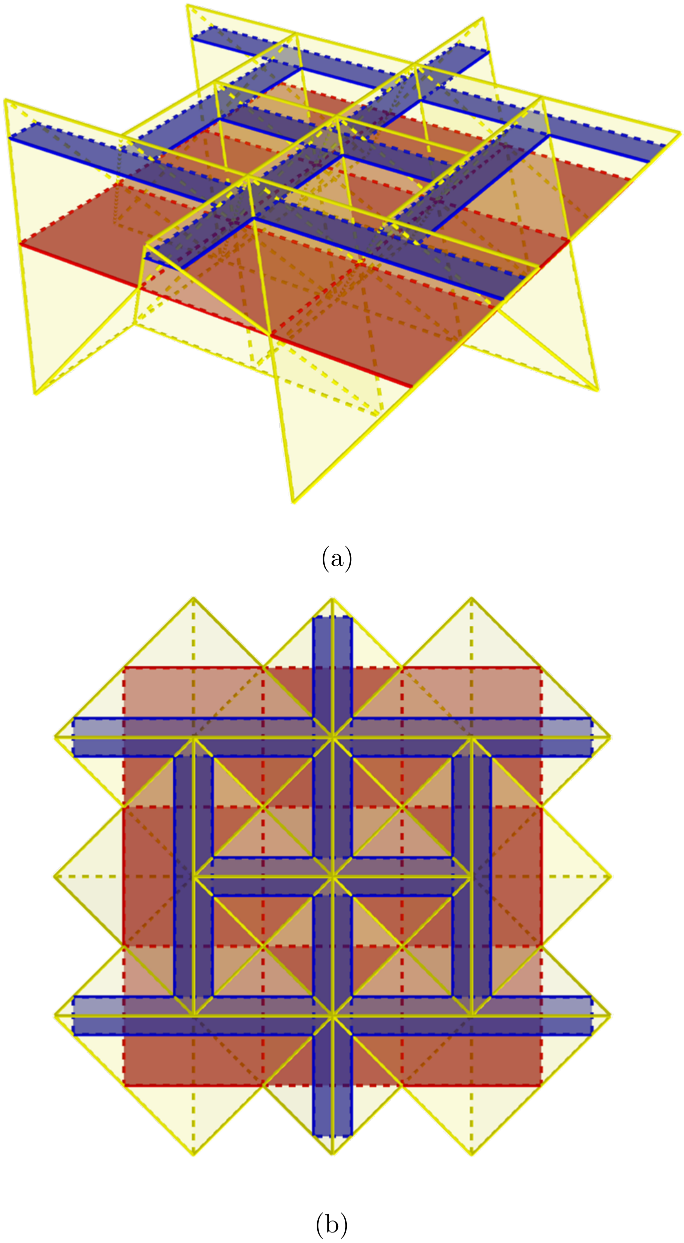
### 5.4.2 Fundamental TI generation Requirement

Two neighboring polyhedra must have a common planar interface between them. Such an interface is the intersection of the respective faces from both polyhedra that are mutually coplanar. Let $f_i, f_j \in F$ be two neighboring tiles in a surface tessellation with shared edges represented as half–edges $ab, ba$ for the respective tile. The tilting angles $\theta^+_{ab} \in \Theta^+_0$ and $\theta^-_{ba} \in \Theta^-_0$ (only for the first step from both evolution sequences) must be $\theta^+_{ab} = -\theta^-_{ba}$. That is, the tilting angles rotate the normal vector associated with the shared edge such that it defines the same tilted plane $P_{a,b}$ that contains the edge. Figure 5.10 shows an example of the angle matching for the tilting angle on a shared edge between two neighboring tiles in a surface tessellation.

This requirement satisfies the cross–section criteria [3] for TI arrangements. For a planar tessellation composed of square tiles, each tile is the mid–section of the respective block in the assembly. A tile evolves into rectangles as it moves upward in the direction of its normal vector. Eventually, the rectangles collapse into a line segment when the evolving tile reaches the top–most section of the block. Simultaneously, the tile also evolves into rectangles as it moves downwards in the opposite direction of its normal vector, and it will collapse into a line segment when the rectangles reach the bottom–most section of the block. The rectangles moving towards the top of the block are rotated $\frac{\pi}{2}$ angles concerning the rectangles moving towards the bottom of the block. Figure 5.11 shows an example of squares evolving into rectangles in an assembly of interlocked tetrahedra. Kanel–Belov *et al.* in [2] introduced the Topological Interlocking criterion using such evolution principle.

The polyhedron section contained between its first evolved $n$–polytopes, along with both positive and negative directions, contribute to the TI behavior of the resultant block. Such a section contains the faces that are in contact between the block and its neighbors. Its geometry is an antiprism without both top and bottom faces. This geometry guarantees the cross–section criteria of the block in the resultant assembly. The polyhedron sections defined after the first evolution steps (along with both positive and negative directions) can contribute to the TI behavior if their respective tilting angles are the same as the tilting angles from the first evolution steps.

Figures 5.9(a) and 5.9(b) show two different polyhedra evolved from the same seed polygon. Both positive evolution sequences are identical. The negative evolution sequences differ in the last step on which the evolving rectangle collapses into a line segment. Such change alters the common interface with a neighboring block in the assembly (if any). Furthermore, it breaks the uniformity of the block without sacrificing the TI property. Furthermore, the volumes of the polyhedra become different (assuming uniform density) from one another. Changes in the evolution step parameters have implications in the distribution of forces required to maintain the static equilibrium of an assembly. We discuss this topic in Chapter 6.


(a)

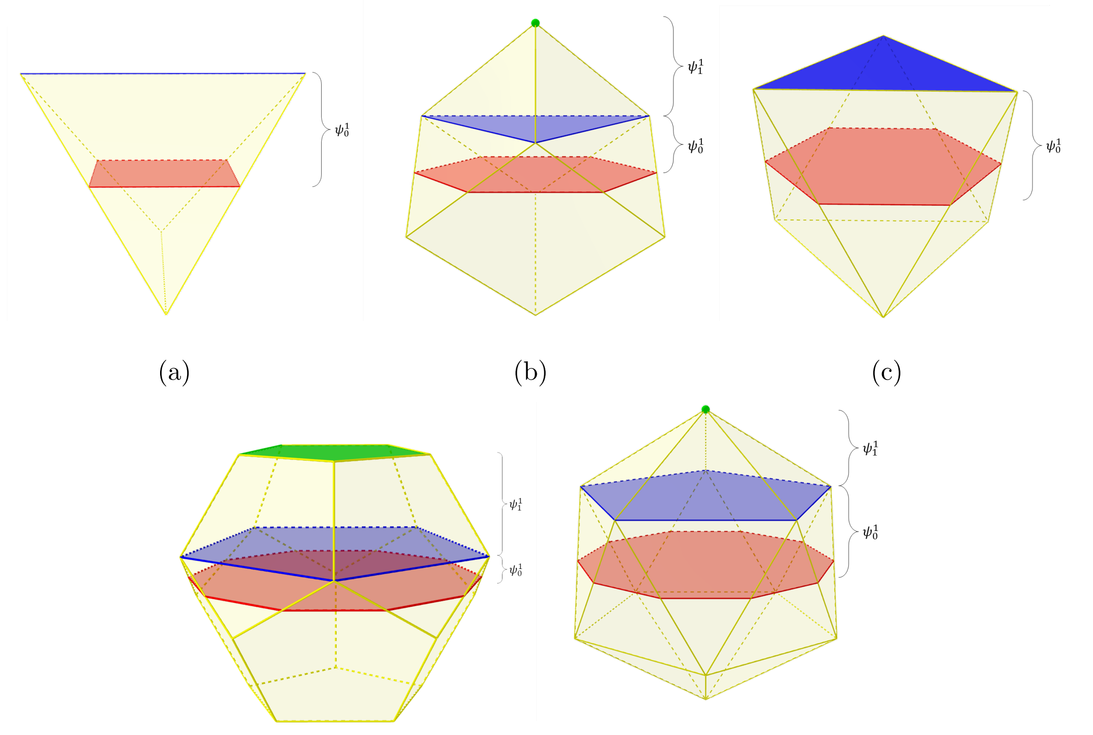
(b)

Figure 5.11. Squares evolving into rectangles while generating tetrahedra. a) Assembly overview, b) Top view showing the rectangles as cross–sections.

### 5.4.3 Platonic Solids

Dyskin *et. al.* [3] showed that all five Platonic Solids have TI properties when placed in a way that the tiles of the tessellation fit as the mid–sections of the solids.

Specifically, a square tile produces a tetrahedron. A hexagon tile produces either a hexahedron, an octahedron, or a dodecahedron. A decagon tile produces an icosahedron. If a tile is regular, then the respective polyhedron can be regular as well. Still, the TAM and HBM generation methods can only generate tetrahedra and octahedra. Generating the cube requires an additional step that intersects the planes tilted towards the same direction on the hexagon. There is no combination of intersecting planes that generate all vertices from a hexagon to a dodecahedron, or from a decagon to an icosahedron.



(a) (b) (c)

(d) (e)

Figure 5.12. Positive evolution steps of the Platonic Solids. Seed polygons in red, evolved $n$–polytopes in blue and green. a) Tetrahedron, b) Cube, c) Octahedron, d) Dodecahedron, e) Icosahedron.

We can use the Uniform Evolution to generate the Platonic solids starting with a respective polygon. Each solid requires the same number of positive and negative evolution steps. The tetrahedron starts with a square and requires $n = 1$ step. The cube starts with a regular hexagon and requires $n = 2$ steps. The octahedron starts with a regular hexagon and requires $n = 1$ step. The dodecahedron starts with a regular decagon and requires $n = 2$ steps. The icosahedron starts with a regular decagon and requires $n = 2$ steps. The last steps for the cube, dodecahedron, and icosahedron must collapse the penultimate evolution polygon into a point. Figure 5.12 shows the positive evolution steps along $\hat{N}$ for each Platonic solids, negative evolution steps are reciprocal steps. The required evolution parameters are functions of the side length $l$ from the respective tile. We list the specific generation parameter values and for each Platonic solid in the appendix section.

## 5.5 Discussion

The Generalized Mid–Section Evolution describes the shape each block in an assembly as the result of a seed polygon evolving along with both directions of its normal vector, the evolved shape is a function of a given set of evolution step parameters. This method can generate any shape returned by the TAM and HBM generation methods. Furthermore, the evolution step parameters can also determine the shape of clipped blocks. The blocks will maintain TI concerning one another, provided the fundamental TI generation requirement is satisfied. The number of generation parameters provides more considerable flexibility to generate a diverse set of TI blocks. A change in such parameters may affect the geometry of the interfaces between neighboring blocks, block volumes, and centers of mass. Such changes are relevant for the static equilibrium analysis of the interlocking assembly.

Using evolving mid–sections helps to describe either convex or concave polyhedra. In particular, we can generate the Platonic solids, their respective truncated versions, antiprisms, and other types of polyhedra with TI properties. It is left to the designer to set the respective parameter values $\Psi_d^k = \{\Theta_k^d, \lambda_k^d\}, d \in \{+, -\}, \forall k = 0, 1, ..., n^d - 1$ that evolve a polygon into a polyhedron.

# 6 SURFACE TESSELLATION PROCESSING AND STATIC EQUILIBRIUM ANALYSIS

## 6.1 Introduction

A surface tessellation is the primary source of information to generate a TIC. The tessellation represents the overall shape that an assembly must resemble. Furthermore, its tiles provide information of the mid-sections for each block. Different tile shapes result in a diversity of block shapes in the assembly. Such shape diversity is an essential factor for the study of TICs when considered as structures or materials.

Knowing the geometries of the blocks opens a path to consider an analysis of the structure in a physical setting. With the geometry of a block, we can calculate its center of mass and volume. If we assume each block has uniform density and the gravity constant, then we can weigh the blocks. Finally, assuming a static friction coefficient allows us to calculate the forces (e.g., compression, tension, and tangential) between the blocks. All this information is relevant for analyzing the static equilibrium of a TIC.

This chapter focuses on two goals: reshaping the tiles in a surface tessellation and analyzing the static equilibrium of a TIC. We first describe methods to reshape the tiles (e.g., subdivisions and dual). This approach helps us to design more complicated surface tessellations (regular, semi-regular, and irregular) to generate TI assemblies based on a variety of block shapes. Second, we use the Static Equilibrium Analysis proposed by Whiting *et al.* [5] to determine if a compression-only structure is stable under gravity. Such analysis introduces a measure of infeasibility based on the magnitude of the tension forces required for the structure to maintain static equilibrium. Finally, we propose a pipeline to find the set of parameters that generate an assembly that requires minimum tension forces. We apply this approach to both parametric and free–form surface tessellations.

## 6.2 Overview

The design of the surface tessellation is the first step to generate a TI assembly. It indicates both the overall shape the assembly must represent and the mid-section of the blocks. The shape of the tiles of the tessellation determines the geometry of the interlocking block. That is, quadrilateral tiles generate tetrahedra, hexagonal tiles generate hexahedra, octahedra, or dodecahedra. Then, editing the surface tessellation translates into different tiles that represent the same overall shape of the original one.

After generating an assembly, it is essential to determine if it can maintain static equilibrium under loads. If so, then the structure is functional and may have useful applications. We run a static equilibrium analysis on the assembly, which returns the required forces (compression, tension, and tangential) to maintain the structure in an equilibrium state. Such forces allow us to adjust the generation parameters until the tension magnitudes are minimum. We explain how this analysis becomes part of the TIC generation pipeline to generate optimal assemblies.

## 6.3 Surface Tessellation Processing

We regard surface tessellations as the primary source of information to generate TI assemblies. A surface tessellation may represent the image of a parametric equation, a NURB, or a 3D mesh. The tiles of the tessellation must have an even number of sides to be useful to generate a TI assembly. In some cases, we need to pre-process the tessellation such that all tiles have an even number of sides. Even more, editing the tiles to have diverse shapes translates into assemblies made of a variety of block shapes. In this section, we discuss approaches to edit the tiles of a surface tessellation that resemble the overall shape of the original one.

### 6.3.1 3D Mesh Quadrangulation

The Topological Interlocking principle (Kanel–Belov *et al.* [2]) requires that each tile in the surface tessellation must have an even number of sides. Such requirement guarantees the direction values at the edges alternate between the inside and the outside of the tile. We have control of the tile shapes when generating the image of a parametric equation. However, traditional 3D meshes for computational purposes use triangles, which do not comply with the requirement.

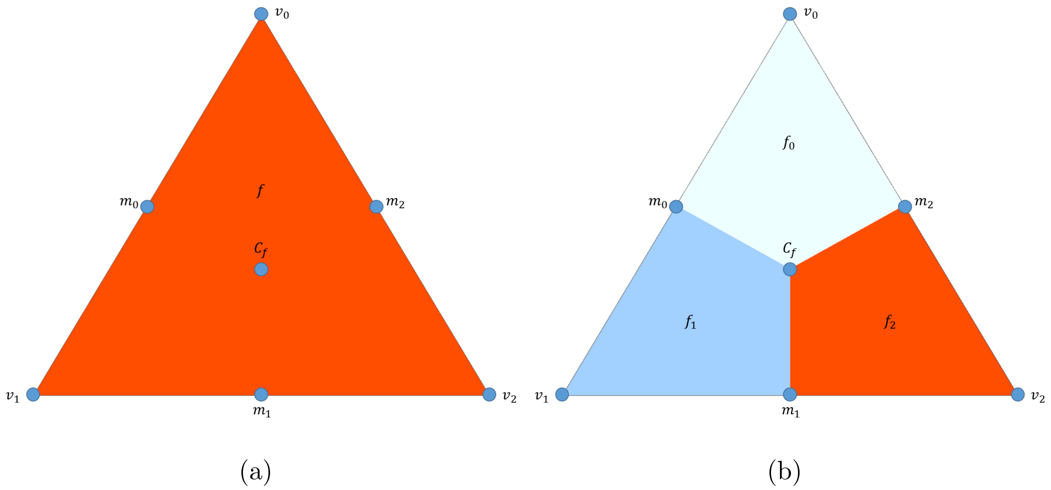
Triangle 3D meshes require an all-convex quadrangulation remeshing. Remeshing algorithms “capture the global structure of the input geometry by aligning groups of elements to the dominant geometric features” [50]. Then, quadrangulation algorithms describe such elements as aligned convex quadrilaterals. These algorithms keep the overall shape of a tessellation using quadrilateral tiles. Then, each tile will have an even number of sides which satisfies the requirement mentioned before.

Mesh quadrangulation is an active topic of research that applies to multiple problems such as Animation, Computer–Aided Geometric Design, Domain Decomposition, and Finite Element Analysis. Recent quadrangulation algorithms are Mixed Integer Quadrangulation (Bommes *et al.* [51]), Instant Field-Aligned Meshes (Jakob *et al.* [52]), Functional cross field design for mesh quadrangulation (Azencot et al. [53]), QuadriFlow (Huang *et al.* [54]) and Quadrangulation through Morse–Parameterization hybridization (Fang *et al.* [55]). The work from Botsch *et al.* [56] [50] and Bommes *et al.* [57] expand the discussion in the topic.

### 6.3.2 Basic Midpoint Subdivision

The basic midpoint subdivision method splits a tile into quadrilaterals (Armstrong *et al.* [37], Hoffmann [58]). Let $M = \{V, F\}$ be a surface tessellation and a tile $f = \{v_0, v_1, ...v_{n-1}\}, f \in F$, where $n$ is the number of vertices in $f$ (which is also the number of edges). The method subdivides $f$ into $n$ quadrilaterals by defining new edges from a center point $C_f \in f$ (e.g., centroid) to the midpoints $m_0, m_1, ..., m_{n-1}$

from each edge of the tile. This subdivision transforms $f$ into $f_0, f_1, ..., f_{n-1}$ tiles. Furthermore, it introduces at most $n+1$ new vertices to $M$ per original tile (the center point and the edge midpoints). Figure 6.1 shows an example of a subdivided equilateral triangle into three quadrilaterals.



(a) (b)

Figure 6.1. Basic midpoint subdivision of an equilateral triangle. a) Original tile with center point and mid–edge points, b) Subdivision tiles.

There are geometric features from a surface that often go missing (e.g., creases) after generating the respective TIC. The geometry of the blocks that correspond to the tiles adjacent to such features occludes their visualization. Furthermore, larger tiles (when compared against its neighbors) correspond to larger blocks in the assembly. Such a situation may not compromise the Topological Interlocking principle but results in artifacts that reduce the resemblance of the assembly to the respective surface tessellation.

We use the basic midpoint subdivision to increase the level of detail from the surface tessellation captured by the assembly. This subdivision approach adds redundancy to the surface tessellation. That is, the subdivision method generates smaller tiles to describe a larger tile from the surface. The results are larger assemblies made of smaller blocks. However, such assemblies highlight features from the surface tessellation better than the assemblies generated from the original ones. Figure 6.2 shows a 3D mesh based on quadrilaterals and its respective TI assembly.

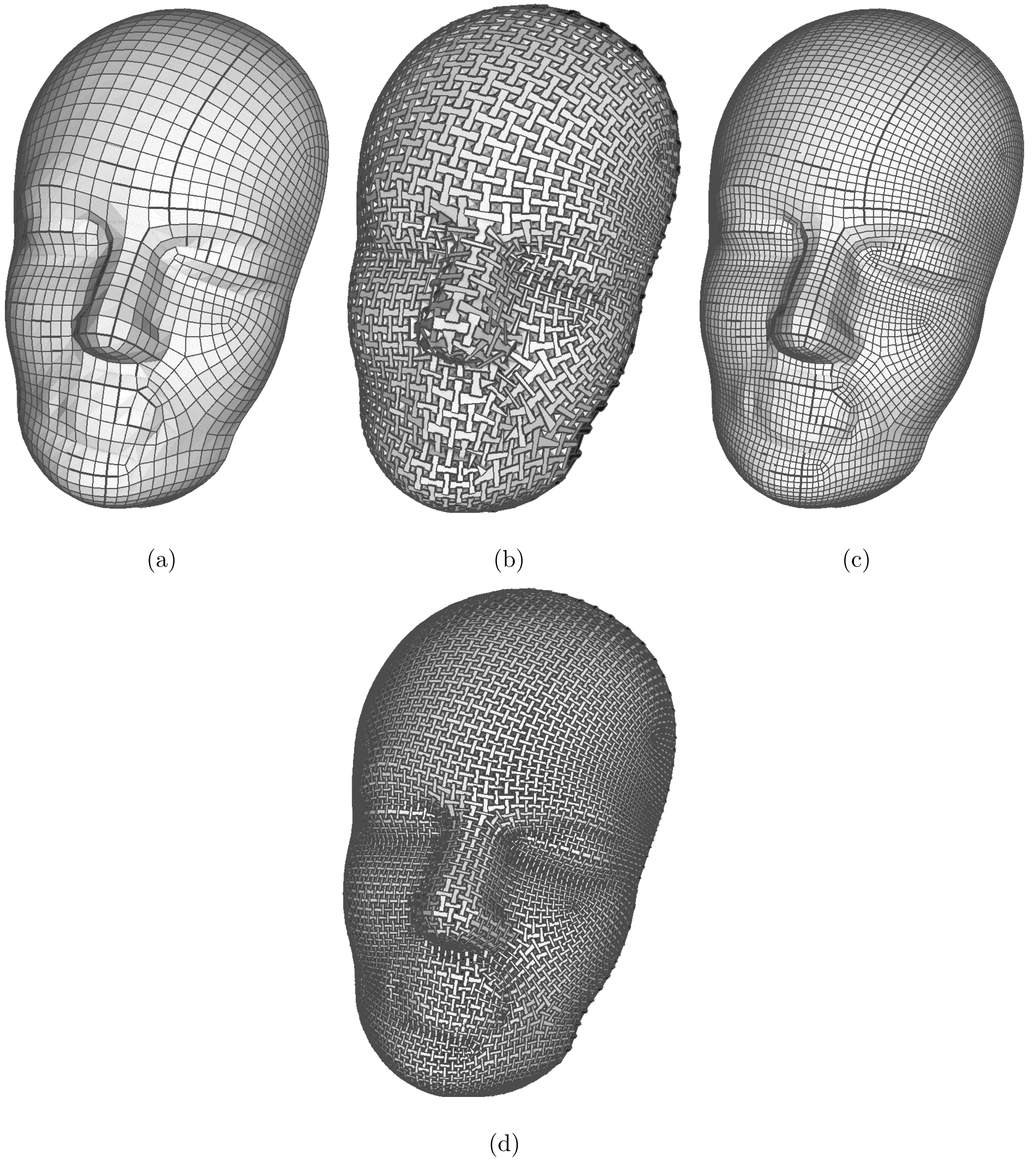
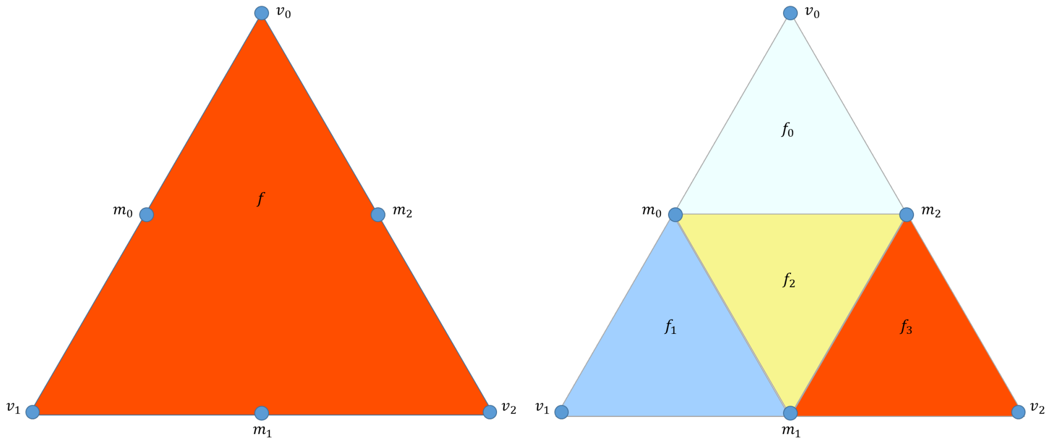


Figure 6.2. Subdivided tessellation using basic midpoint subdivision and their respective assemblies. a) Original (1384 vertices, 1332 tiles and 2715 edges), b) TI based on the original tessellation, c) Subdivided (5431 vertices, 5328 tiles and 10758 edges), d) TI based on the subdivided tessellation.

### 6.3.3 Mid–Edge Points Subdivision

The Mid–Edge Points subdivision method splits a tile into a smaller version of the original tile plus a set of triangles. Let $M = \{V, F\}$ be a surface tessellation and a tile $f = \{v_0, v_1, ...v_{n-1}\}, f \in F$, where $n$ is the number of vertices of $f$ (which is also its number of edges). The method subdivides $f$ into a smaller version of the tile and triangles by defining edges between consecutive midpoints $m_0, m_1, ..., m_{n-1}$ (the midpoints from two consecutive edges in a tile). This subdivision generates $f_0, f_1, ..., f_{n-1}, f_n$ new tiles. Furthermore, it introduces at most $n$ new vertices to $M$ per original tile (the edge midpoints). Figure 6.3 shows an example of a subdivided equilateral triangle into four triangles.



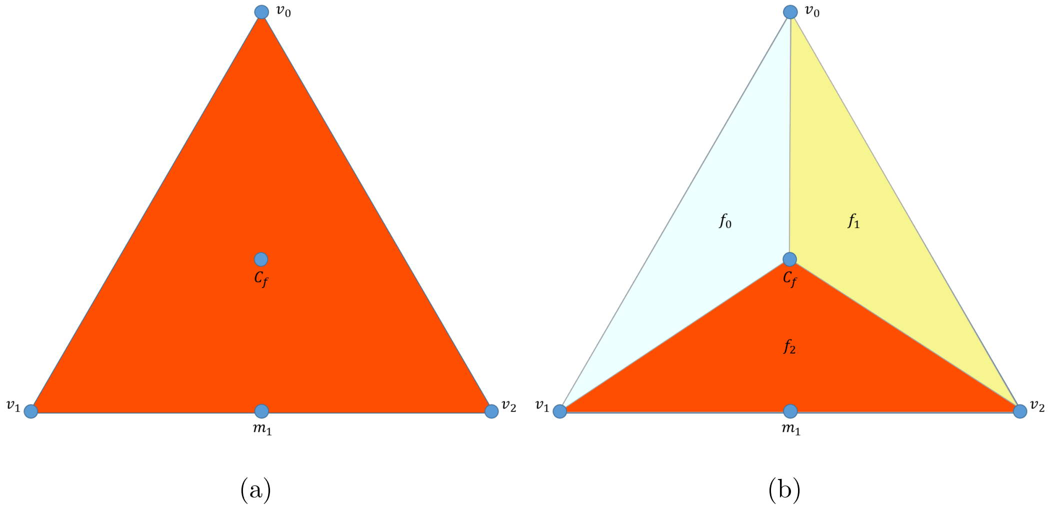
(a)

(b)

Figure 6.3. Mid–Edge Point subdivision example. a) Original tile with mid–edge points, b) Subdivision tiles.

### 6.3.4 Triangular Subdivision

The Triangular Subdivision method splits a tile into triangles. Let $M = \{V, F\}$ be a surface tessellation and a tile $f = \{v_0, v_1, ...v_{n-1}\}, f \in F$, where $n$ is the number of vertices in $f$ (which is also its number of edges). The method subdivides $f$ into triangles by defining new edges from a center point $C_f \in f$ (e.g., centroid) to the vertices of $f$. This subdivision generates $f_0, f_1, ..., f_{n-1}$ tiles. Furthermore, it introduces $n$ new vertices to $M$ per original tile (the center points). Figure 6.4 shows an example of a subdivided equilateral triangle into three triangles.



(a)

(b)

Figure 6.4. Triangular subdivision example. a) Original tile with center point, b) Subdivision tiles.

### 6.3.5 Dual Surface Tessellation

We use the dual of a surface tessellation to change its tiling and generate TI assemblies with a diverse family of blocks. In particular, duals are useful to generate tiles with an even number of sides (based on the valence from the original vertices). A surface tessellation and its dual describe the same geometric domain. Then, we can generate TI assemblies that differ in the geometry of their blocks but still resemble the original surface.

Using the tile subdivision methods along with the dual of the surface tessellation make possible to generate tessellations similar to the ones used by Weizmann *et al.* in [16] [17] [18] to generate TI catalogues. We redesign the tiling of a surface tessellation using the subdivision methods and the dual. Such procedures facilitate the generation of regular and semi-regular tessellations. Figures 6.5, 6.6, and 6.7 show examples of such tessellations and how to generate them using the subdivision methods and the dual.

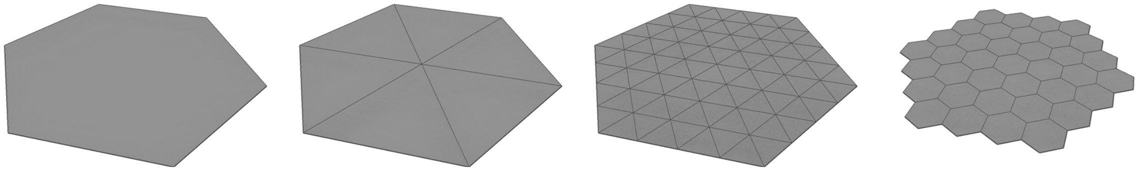


Figure 6.5. Generating a surface tessellation based on hexagonal tiles. From left to right: Single hexagon tile, triangular subdivision, mid–edge points subdivision (x2), dual tessellation.

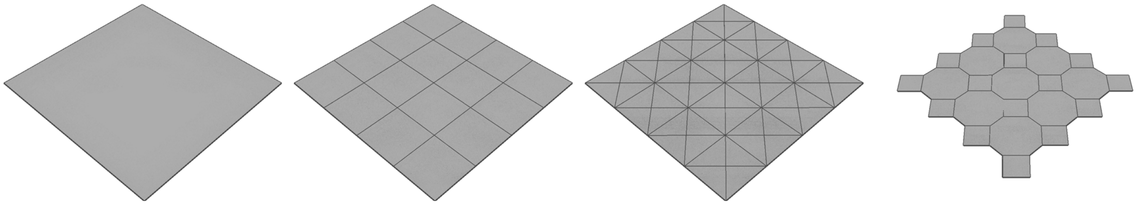


Figure 6.6. Generating a surface tessellation based on square and octagon tiles. From left to right: Single square tile, basic midpoint subdivision (x2), triangular subdivision, dual tessellation.

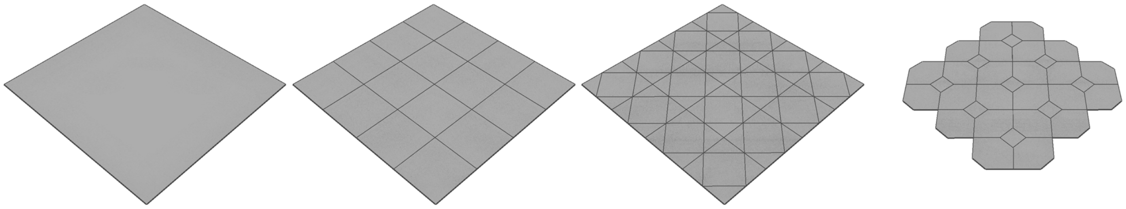


Figure 6.7. Generating a surface tessellation based on square and hexagon tiles. From left to right: Single square tile, basic midpoint subdivision (x2), mid–edge points subdivision, dual tessellation.

We apply such methods to 3D tessellations to generate elaborated interlocking assemblies. As an example, Figure 6.8 shows the TI assemblies from surface tessellations that represent the same geometric domain. The original tessellation is a quad mesh. The re-meshed version starts with the triangular subdivision of the original tiles. Then we get its dual to generate a tessellation composed of quadrilaterals and octagons. The generated assemblies resemble the original surface tessellation.

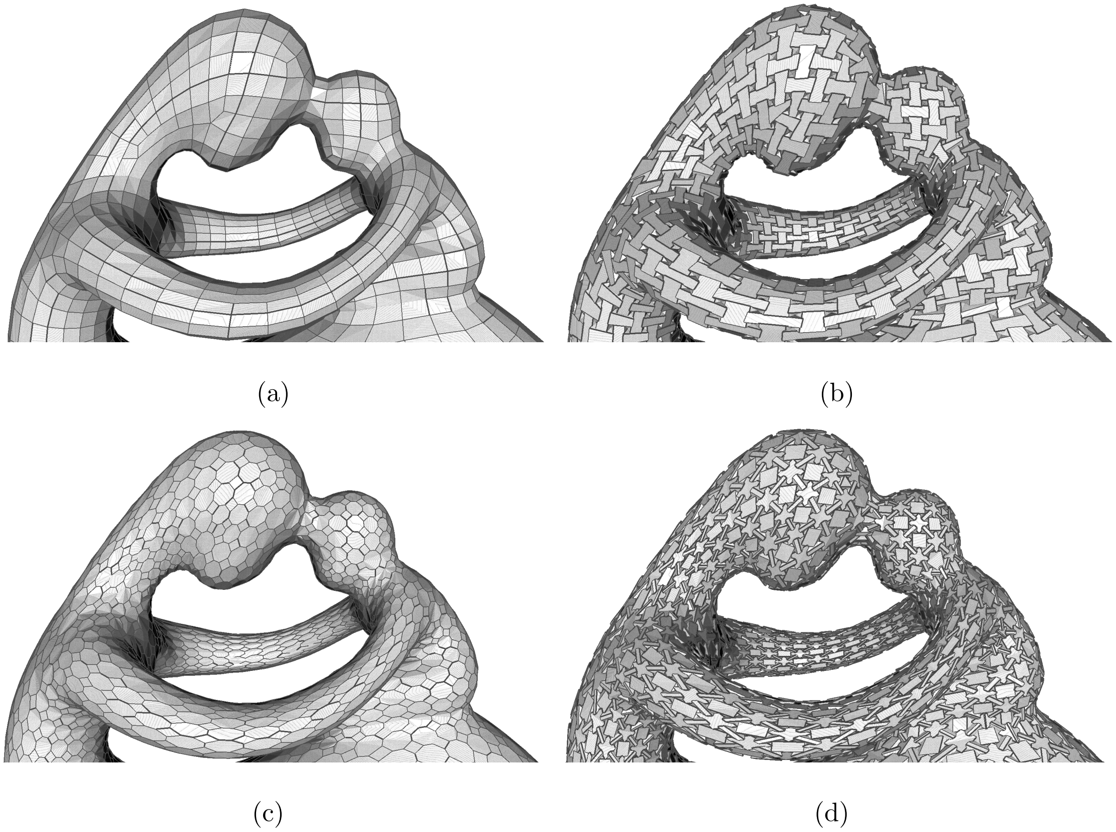


Figure 6.8. Assemblies from a tessellation and re-meshed. a) Original quad mesh, b) TI assembly from quad mesh, c) Subdivided tessellation (triangular subdivision and dual), d) TI assembly from re–meshed tessellation

## 6.4 Static Equilibrium Analysis

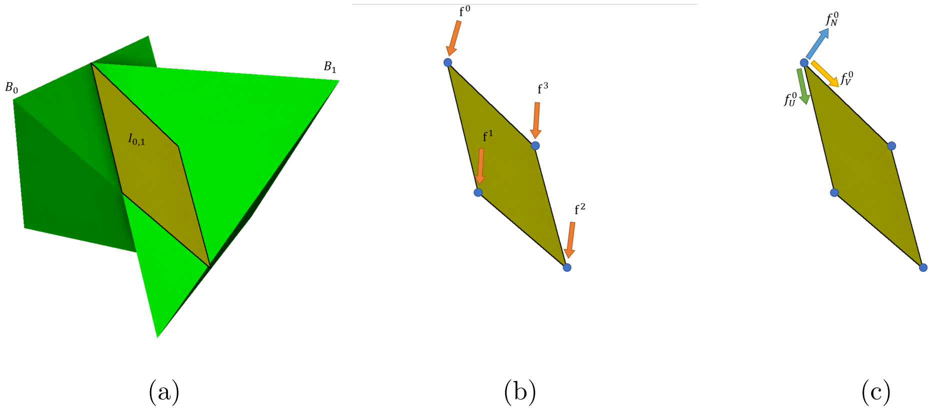
We analyze the stability of the assembly using the measure of infeasibility for compression-only structures under gravity loads introduced by Whiting *et al.* [5]. This measure follows after the Static Equilibrium Analysis described by Livesley [59]. It defines a system of linear constraints representing the distribution of forces (axial and tangential) required to keep a structure in a static equilibrium state. Livesley's analysis determines whether the structure is feasible or not. That is, the geometry of the blocks, along with the physical assumptions (block density, gravity, and static friction coefficient) guarantees the structure maintains equilibrium. Otherwise, the structure is infeasible. Whiting *et al.* relaxed the constraints by decomposing the axial forces into compression and tension components. Their approach minimized the magnitude of the required tension forces using a nonlinear optimization system. The resultant distribution of forces indicates the interfaces between blocks that require tension to maintain static equilibrium.

This analysis method was later expanded to describe tensions introduced by support cables (Whiting *et al.* [60]) and additional constraints for optimal assembly steps (Deuss *et al.* [61]). Recent applications of this method are the decomposition and design of self–supporting discrete structures (Frick *et al.* [62]), space filling structures based on scutoids [63] (Subramanian *et al.* [64]), and the stability analysis on dry–stacked structures focusing on kerns (shrunk interface polygons) as a safety factor (Liu *et al.* [65]).

### 6.4.1 Measure of Infeasibility

The static equilibrium analysis based on the measure of infeasibility require specific assumptions regarding the physical conditions in the structure: blocks are rigid bodies with uniform density, there is static friction between the blocks, and blocks have gravity load.

We describe the elements to calculate the measure of infeasibility in the context of TI assemblies. As defined by Whiting *et al.* [5], a structure $T$ is an assemblage of rigid blocks. The interface $I_{i,j}$ between two adjacent blocks $B_i, B_j \in T$ is a planar polygon representing the contact region between such blocks. A 3D force $\mathrm{f}^k$ is positioned at each vertex $v_k \in I_{i,j}$, the forces are modelled as a linear force distribution. Each force $\mathrm{f}^k$ is decomposed into $f_N^k, f_U^k, f_V^k$ representing the normal component perpendicular to $I_{j,k}$ and two orthogonal friction forces tangential to the interface. Figure 6.9 shows the interface between two tetrahedra, its force distribution, and the force decomposition of one of such forces.



(a) (b) (c)

Figure 6.9. Interface between two blocks and respective force distribution. a) Interface $I_{0,1}$ between blocks $B_0$ and $B_1$, b) Force distribution at the vertices of $I_{0,1}$, c) Force decomposition of $\mathrm{f}^0$ into normal and tangential components (top vertex only).

Each force component is subject to respective constraints. Normal force components must be non–negative, then

$$f_N^k \geq 0 \tag{6.1}$$

Tangential components are constrained to the friction cone defined by the normal component $f_N^k$, and the friction coefficient $\mu$ of the interface. For practical purposes, we assume this coefficient is the same for all interfaces between the blocks of the assembly. A four–sided friction pyramid approximates the friction cone. Then

$$|f_U^k|, |f_V^k| \leq \frac{\mu}{\sqrt{2}} f_N^k \tag{6.2}$$

Multiplying $\mu$ by $\frac{1}{\sqrt{2}}$ we define a conservative friction pyramid that fits within the actual friction cone. Each normal component $f_N^k$ is relaxed by expressing it in terms of a compression $f_N^{k+}$ and a tension $f_N^{k-}$ components such that

$$f_N^k = f_N^{k+} - f_N^{k-} \tag{6.3}$$

Such components must be non–negative, then

$$f_N^{k+}, f_N^{k-} \geq 0 \tag{6.4}$$

The static equilibrium condition states that the net force and net torque for each block in the assembly must be equal to zero. That is

$$\mathrm{A}_{eq} \cdot \mathrm{f} + \mathrm{W} = 0 \tag{6.5}$$

where $\mathrm{A}_{eq}$ is the matrix of coefficients for the equilibrium equations of the blocks, f is the vector of forces at the interfaces between blocks, and W is the vector of force loads at each block. This system has 6 rows per block (3 equations for net force and 3 equations for net torque). Matrix $\mathrm{A}_{eq}$ is sparse due to the small number of interfaces per block (a block $B_i$ will have at most $n_i$ interfaces, where $n_i$ is the number of faces of the block).

The friction constraints for each block in the assembly are represented as a linear system of inequalities, then

$$\mathrm{A}_{fr} \cdot \mathrm{f} \leq 0 \tag{6.6}$$

where $\mathrm{A}_{fr}$ is the matrix of coefficients for the inequalities, and f is the vector of forces at the interfaces between blocks.

The measure of infeasibility is the sum of the squared tension components from all force vectors at each interface between the blocks of the assembly. For compression–

only structures, tension forces are non–existent, then the goal is to penalize the tension forces that make the structure stable. The system is stated as a quadratic program:

$$\begin{aligned} y(\Psi) = \min_{\mathrm{f}} & \sum_{n}^{i=0} \left(f_N^{i-}\right)^2 \\ \text{s.t.} \quad & \mathrm{A}_{eq} \cdot \mathrm{f} + \mathrm{W} = 0 \\ & \mathrm{A}_{fr} \cdot \mathrm{f} \leq 0 \\ & f_N^{i+}, f_N^{i-} \geq 0 \end{aligned} \tag{6.7}$$

where $\Psi$ is the set of parameters to generate $T$ using a generation method $G$ (e.g., Tilting Angle Method, Height–Bisection Method or General Mid–Section Evolution method) based on a surface tessellation $M$. The expression indicates that the measure of infeasibility is a function of the generation parameters $\Psi$ for an assembly $T$. That is, a change in $\Psi$ represents a change in the geometry of the blocks (at least one block changes). Such a change alters the volume and centroid of the block. Furthermore, it modifies the geometry of the interfaces between the blocks. Then, the force distribution at the interfaces changes as well since the coefficients of the net force and net torque equations are a function of the volume and centroid of the respective blocks.

Expression 6.7 can be generalized for all force components (compression, tension and tangential forces) such that the solution of the system returns the minimum force magnitudes required for an assembly to have static equilibrium. That is,

$$\begin{aligned} y(\Psi) = \min_{\mathrm{f}} & \frac{1}{2}\mathrm{f}^T \mathrm{H} \mathrm{f} \\ \text{s.t.} \quad & \mathrm{A}_{eq} \cdot \mathrm{f} + \mathrm{W} = 0 \\ & \mathrm{A}_{fr} \cdot \mathrm{f} \leq 0 \\ & f_N^{i+}, f_N^{i-} \geq 0 \end{aligned} \tag{6.8}$$

where H is the diagonal matrix with the penalization coefficients for each force component (high values for tension forces and low values for the remaining). System 6.8 will have an optimal solution if H is positive semi–definite.

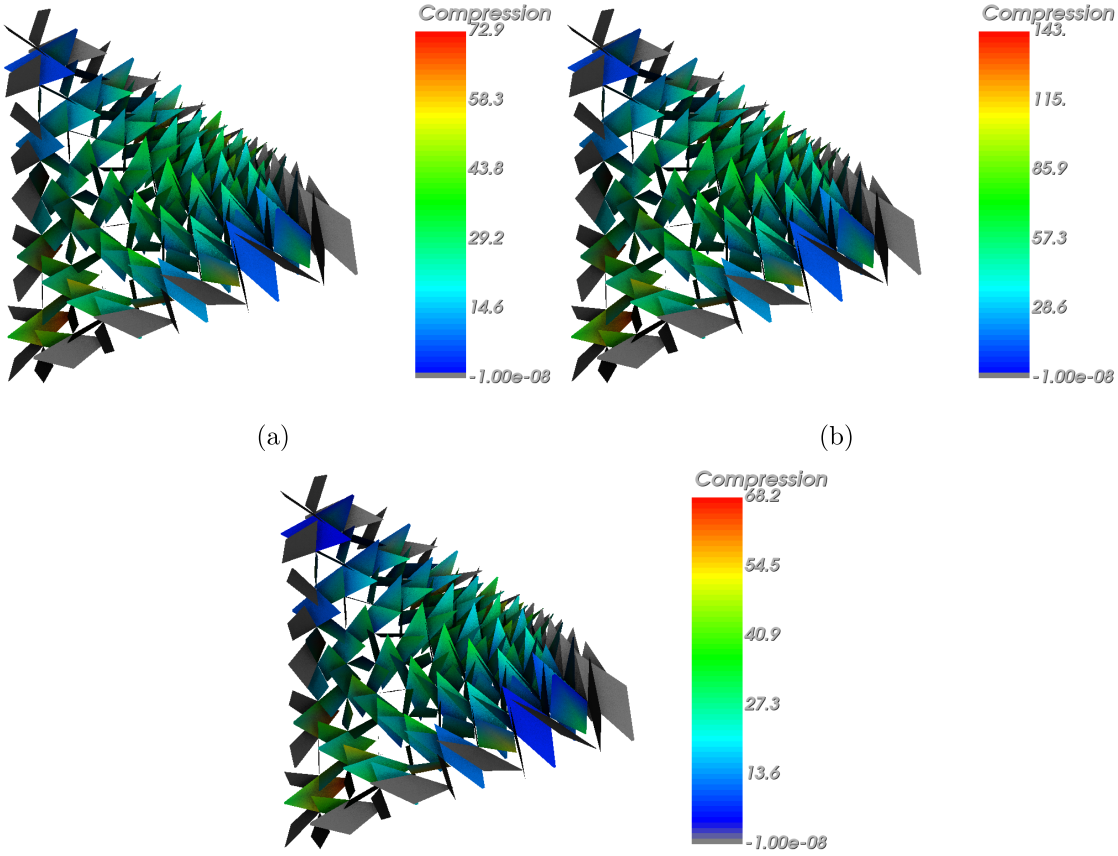
### 6.4.2 Normalized Force Magnitudes

The force magnitudes vector f found using the Expression 6.8 has units based on the indicated block density and gravity values. This approach allows the study of the static equilibrium of the assembly using different material properties and sizes. We can normalize the resultant force magnitudes by dividing them by the density value. By doing this, the static equilibrium analysis only requires the static friction coefficient $\mu$ and gravity value to calculate the required forces for the structure to maintain static equilibrium. Then, we can obtain specific forces per material (assuming rigid blocks and uniform block density) by multiplying the normalized force magnitudes by the specified density value. Figure 6.10 shows an example of the compression force values (using different density values) and normalized force values.

### 6.4.3 Optimal Interlocking

We use the expression 6.8 to calculate the minimum forces required for an assembly $T$ to maintain static equilibrium. Still, small changes in the generation parameters $\Psi$ may result in reduced tension forces for $T$ to maintain equilibrium. The optimal interlocking is the state of $T$, such that no changes in $\Psi$ reduce the magnitude of the tension forces.

Figure 6.11 shows the overall pipeline we follow to generate TI assemblies that require minimum tension forces to maintain static equilibrium under gravity load. This pipeline is an extension over the one in Figure 4.1 to generate valid assemblies (discussed in Section 4.2). The extended pipeline adds the equilibrium analysis over a valid assembly. It may happen that a valid assembly does not have optimal interlocking (i.e., making slight changes in the generation parameters reduce the magnitude of the tension forces). In practice, this pipeline can be interactive by allowing the user to specify the number of iterations to search for an assembly with optimal interlocking. By doing so, the equilibrium parameters $\Psi'$ would be closer to feasible parameters $\Psi^*$ reducing the compromise between validation and interlocking.



(a)

(b)

(c)

Figure 6.10. Normalized compression forces using friction coefficient $\mu = 0.4$. a) Compression force magnitudes using density $\rho = 1.07$ g/cm$^3$, b) Compression force magnitudes using density $\rho = 2.1$ g/cm$^3$, c) Normalized force magnitudes.

### 6.4.4 Support Frame Design

TI assemblies require support frames that prevent lateral motions of the pieces at the boundary. Without such support, the blocks will fall apart due to their pose for interlocking purposes. A support frame could be an additional structure explicitly designed to hold an assembly. In such a case, we require a new design process that could not be entirely related to surface tessellation.

We consider support frames made of blocks from an assembly. As an example, we consider the sequence of blocks at the periphery of the assembly for such a frame.

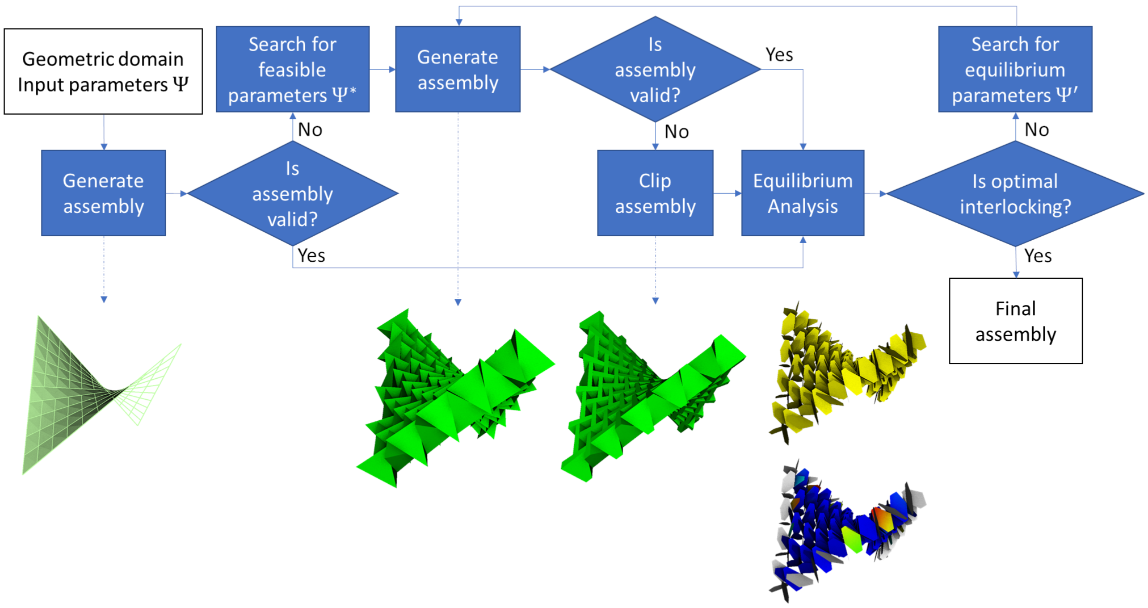


Figure 6.11. Pipeline to generate stable TI assemblies.

Doing so removes the issue of lateral motions with such blocks. Additionally, the support frame design comes after the actual design of the assembly. Which means no new generation process runs to generate the support structure. Another support frame consists of the sequence of blocks along with the interfaces with larger tension magnitudes. There are assemblies where the tension forces have larger magnitudes due to the shape of the surface tessellation (e.g., concave sections). Then, the support frame consists of the sequence of blocks between the interface with larger tension forces.

A competent frame is one that provides support to the assembly and requires a low number of blocks. Using the system 6.8, we find the tension magnitudes at the interfaces between blocks. The geometries from the blocks with interfaces that require larger tension magnitudes become part of the frame. It may be possible to have an isolated high tension section in the assembly. We connect such sections along with the existing support frame (if any) using the shortest path between them. The resultant support frame must include the blocks at an extreme side of the assembly concerning the direction of the force loads. As an example, having gravity load requires to use the blocks at the bottom of the assembly as the frame.

## 6.5 Results

We designed multiple TI assemblies using different types of surface tessellations (e.g., parametric surfaces, solids, and 3D meshes). The tile subdivision methods allowed us to generate different tessellations that represent the same geometric domain. Each tile subdivision results in different interlocking blocks, which affect the result of the feasibility of the respective assembly.

We use TIGER (Topological Interlocking GEneratoR) to design and analyze the TICs listed below. TIGER is a tool of our own developed on C++, where we implemented the generation methods and the static equilibrium analysis mentioned in this thesis. We use VTK for rendering geometries and displaying the force magnitudes. We use the Gurobi solver [66] to implement the Static Equilibrium Analysis and find the force magnitudes required to reach an equilibrium state in each TI assembly.

For the Static Equilibrium Analysis, we consider ABS plastic and Masonry materials since they are typical for 3D Printing and building, respectively. In Table 6.1, we list the parameters for the materials considered during the Static Equilibrium Analysis. Finally, Table 6.2 lists the descriptions of the equilibrium analysis for each result discussed in this section.

Table 6.1.
Parameters of the materials considered during the Static Equilibrium Analysis.

| Material | Density $\rho$ | Static Friction $\mu$ |
|---|---|---|
| ABS Plastic | 1.07 g/cm$^3$ | 0.4 |
| Masonry | 2.1 g/cm$^3$ | 0.7 |

### 6.5.1 Planar TICs

First, we considered the traditional planar TIC based on tetrahedra. Figure 6.12 shows a planar 7 × 7 assembly made of regular tetrahedra. The support frame consists of an additional sequence of tetrahedra put together as a single unit. Figure 6.13 shows the force magnitudes of the compression and tension forces at the interfaces between the blocks. A color gradient represents the magnitudes of such forces as calculated by the static equilibrium analysis of the assembly. The blue color indicates zero magnitudes, while the red color is the respective highest magnitude. The blocks next to the center of each boundary require tension to support the weight from the internal blocks. In both cases, the support frame provides the required tensions.

A variation of the planar TIC considers an Archimedean tessellation to generate the blocks. Figure 6.14 shows the elements to generate a planar TIC based on clipped tetrahedra and square antiprisms. The area of the interfaces between blocks (see Figure 6.14(d)) is smaller when compared against the interfaces from the previous assembly (see Figure 6.12(d)) due to the shape of the blocks. However, the number of interfaces increases along with the number of vertices per interface. Figure 6.15 shows the force magnitudes for the planar Archimedean assembly are smaller concerning the tetrahedra assembly. The support frame comes from the blocks at the periphery of the assembly put together as a single unit.

### 6.5.2 Curvilinear TICs

Next, we considered TICs based on curvilinear tessellations. These kinds of TI assemblies are of interest to architects and engineers, as discussed in Chapter 2. However, the dynamics of such assemblies are unknown to the best of our knowledge.

The first curvilinear tessellation represents a saddle surface from the equation $z = xy$. Figure 6.16 show the elements to generate the respective assembly. We design the saddle tessellation using the same number of tiles from the first planar

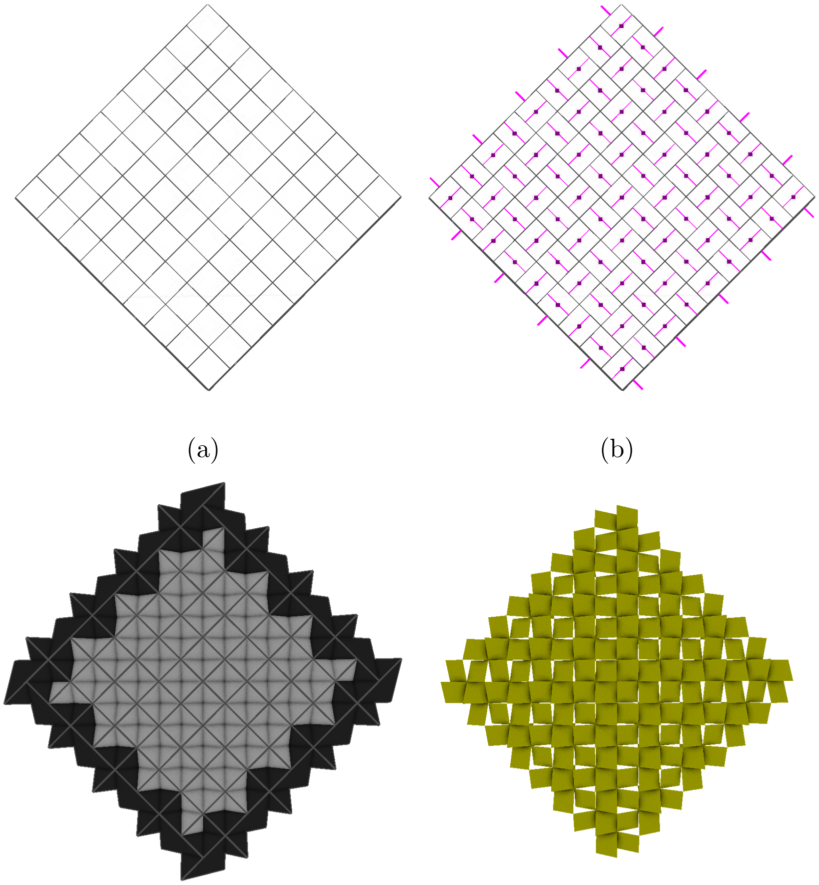
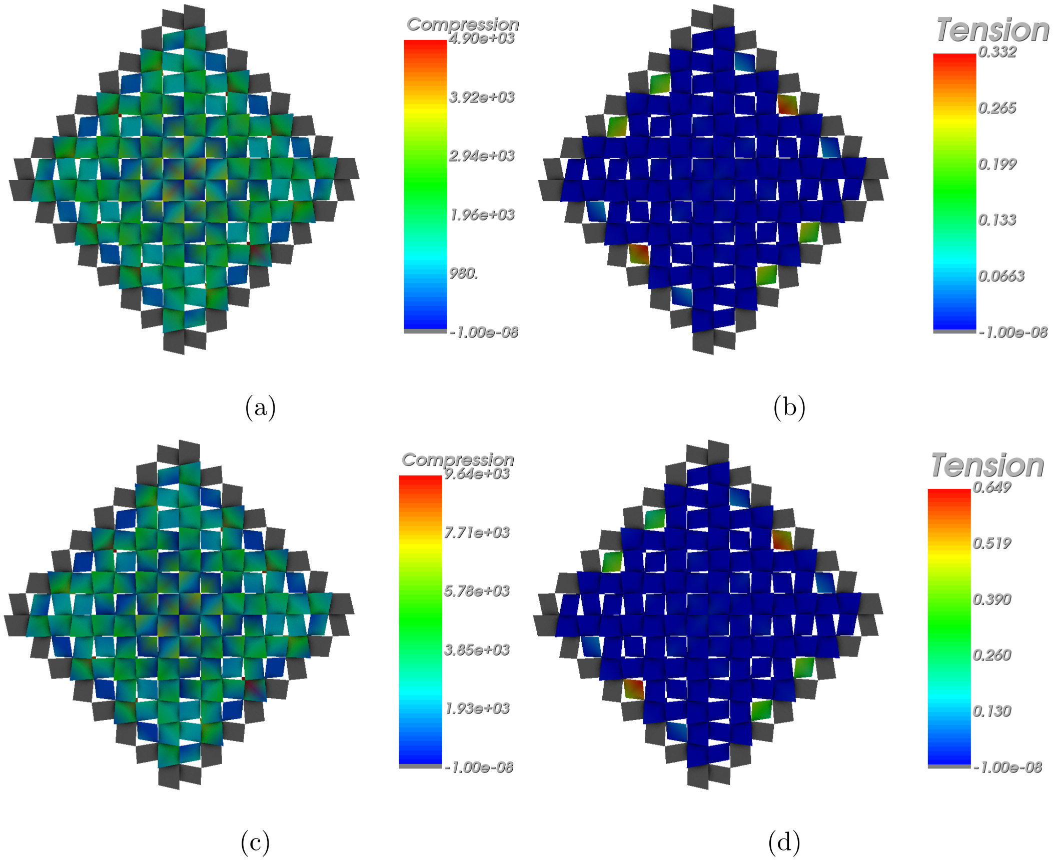

(a)
(b)
(c)
(d)

Figure 6.12. Elements for a planar TIC based on tetrahedra. a) Tessellation, b) Tile centers and edge direction vectors, c) Assembly, d) Interfaces between blocks.

result to maintain correspondences as much as possible. Similarly to the planar assemblies, the blocks at the periphery of the assembly make up the support frame.

Figure 6.17 show the compression and tension magnitudes in the assembly. The distribution of the compression forces in the saddle assembly differs concerning the planar assemblies. The interfaces at the top locations of the assembly have the lowest compression magnitudes, while the interfaces at the bottom have the highest ones. The tension magnitudes behave similarly concerning the planar assemblies. In this case, higher tension magnitudes locate at the top of the assembly. Meanwhile, higher



(a) (b)

(c) (d)

Figure 6.13. Compression and tension force magnitudes on a planar TIC based on tetrahedra. a) Compression forces (ABS plastic), b) Tension forces (ABS plastic), c) Compression forces (Masonry), d) Tension forces (Masonry).

tension magnitudes locate at the central sections of the support frame in the planar assemblies.

We consider a second saddle tessellation composed of quadrilateral and octagonal tiles. First, we subdivide the tiles of the original tessellation (see Figure 6.16(a)) into triangles. Then, we get the dual tessellation, which is composed of the mentioned tile shapes. Figure 6.18(a) shows the resultant saddle tessellation. Similarly to the planar counterpart (see Figure 6.14(a)), the block shapes are clipped tetrahedra and square antiprisms.

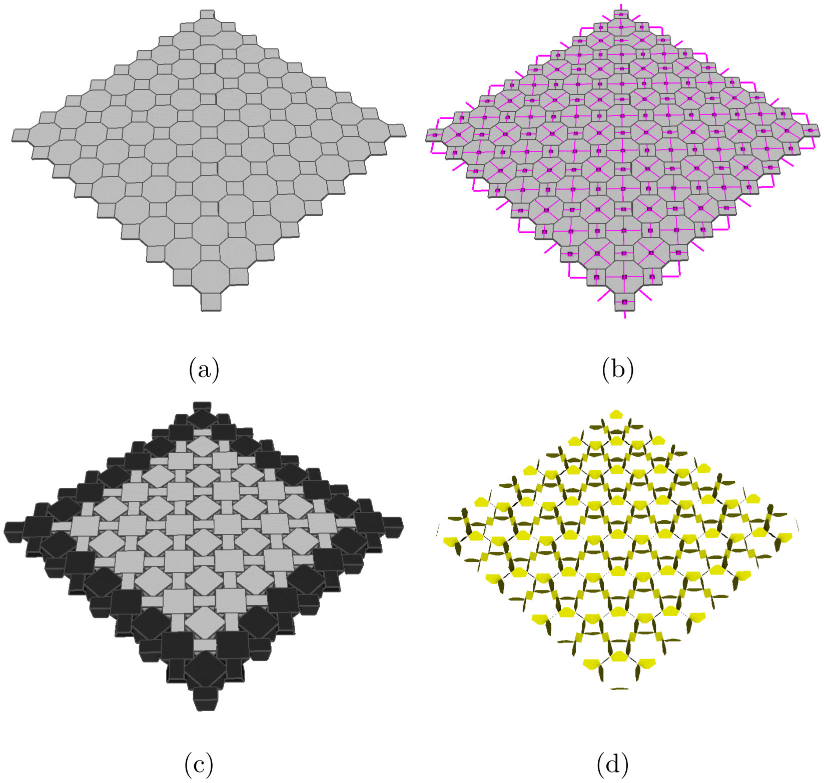
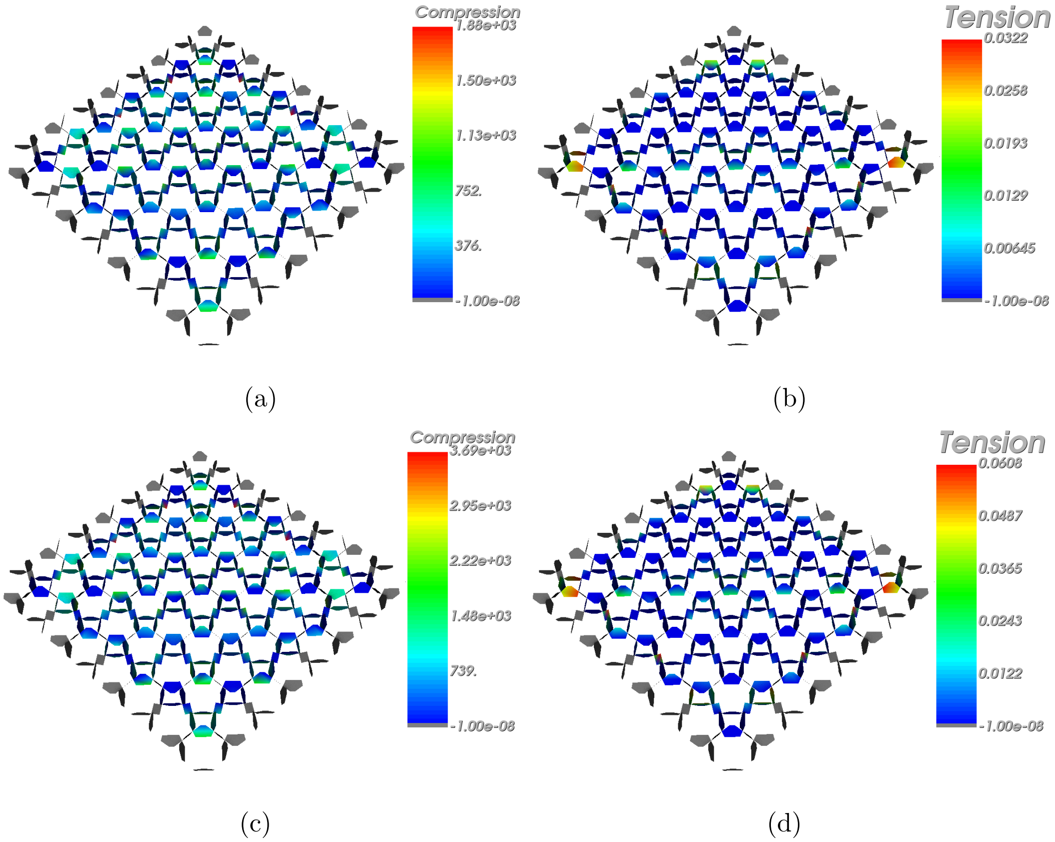


Figure 6.14. Elements for a planar TIC based on diverse antiprisms. a) Tessellation, b) Tile centers and edge direction vectors, c) Assembly, d) Interfaces between blocks.

Figure 6.19 show the compression and tension magnitudes for the TI assembly based on clipped tetrahedra and square antiprisms. The distribution of the forces is similar to the distribution for the initial saddle assembly. The lower compression magnitudes locate at the top locations of the assembly, while the higher magnitudes locate at the bottom. Similarly, the tension forces locate at the top locations next to the support frame.

In Chapter 4, we discussed the generation of cylindrical TICs using the Shape–Fit Method. Here, we approach the same type of TICs using the generation and analysis pipeline shown in Figure 6.11. A polygonal approximation of a cylinder using squares has length $l = 2rn \sin\left(\frac{\pi}{m}\right)$, where $r \in \mathbb{R}^+$ is the radius of the cylinder, $n \in \mathbb{N}^+$ is the number of rings of the cylinder, and $m \in \mathbb{N}^+$ is the number of blocks per ring. As



(a)

(b)

(c)

(d)

Figure 6.15. Compression and tension force magnitudes on a planar TIC based on diverse antiprisms. a) Compression forces (ABS plastic), b) Tension forces (ABS plastic), c) Compression forces (Masonry), d) Tension forces (Masonry).

mentioned in Section 4.5.2, the blocks of a cylindrical TIC hold to the interlocking principle when $m \geq 6$. Figure 6.20 shows the elements to generate a TIC using a cylindrical tessellation made of square tiles.

The support frame for the cylindrical TIC comes from the blocks at both boundaries. That is, the blocks at the endpoints of the cylinder behave as one single unit that contains the assembly. In a physical setting, both endpoint sections may be put together through an internal pole or spring.

Figure 6.21 shows the force magnitudes for the cylindrical TIC. The compression forces located at the bottom of the assembly. Meanwhile, the tension forces locate at

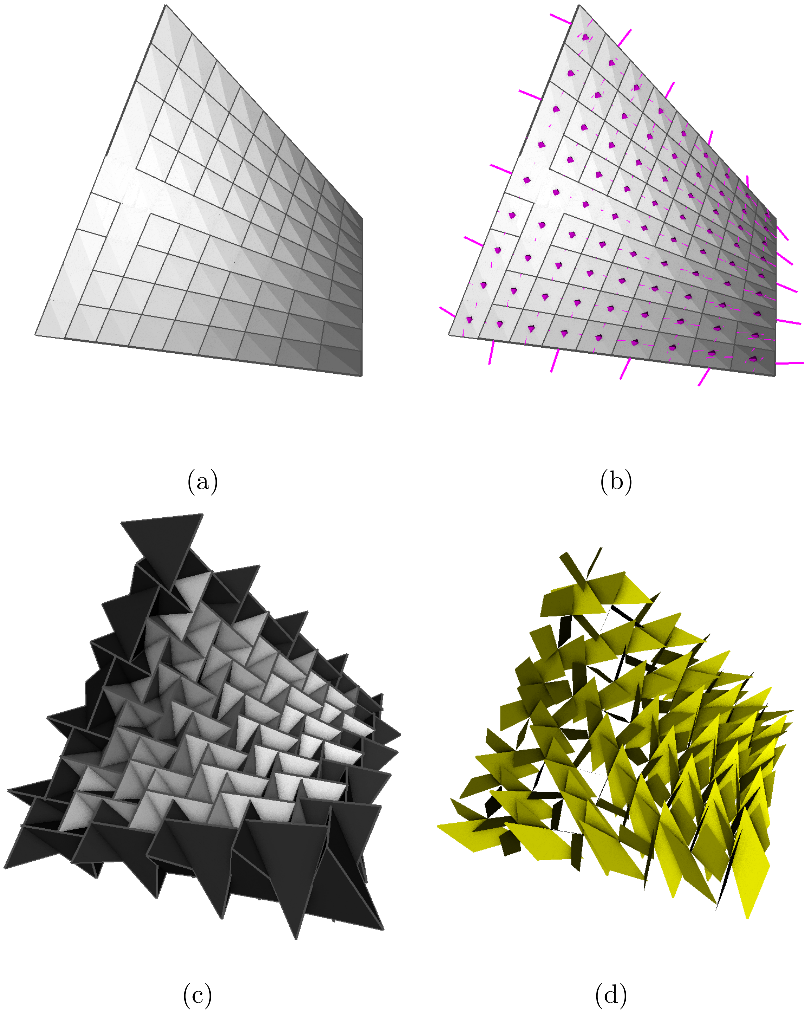
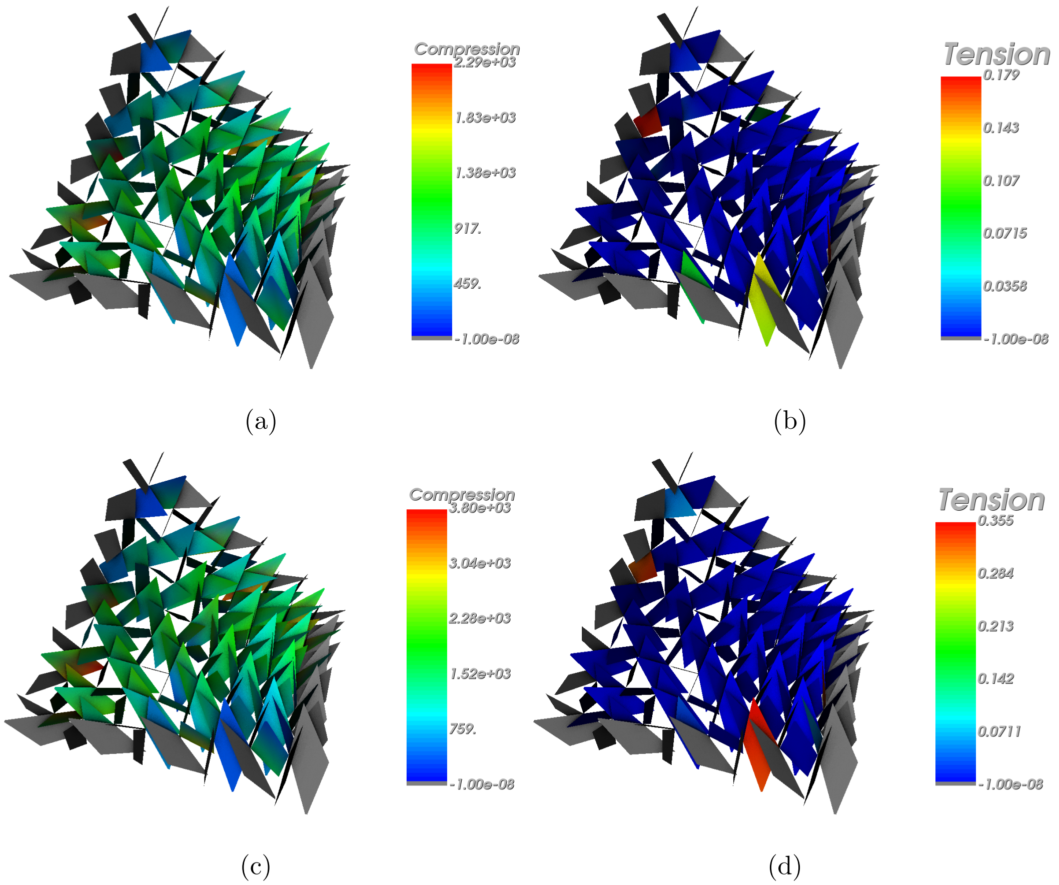


Figure 6.16. Elements for a saddle TIC based on tetrahedra. a) Tessellation, b) Tile centers and edge direction vectors, c) Assembly, d) Interfaces between blocks.

the middle and top locations of the assembly. In this assembly, the orientation of the blocks lock push and pull motions (along the normal vectors of the tiles) but do not prevent lateral strain. Then, the static equilibrium analysis returns higher tension magnitudes in order to keep the blocks in their respective location and maintain an equilibrium state.

We also considered a cylindrical TIC in a horizontal position. Figure 6.22 shows the elements to generate the assembly. The support frame holds to the same obser-



(a)

(b)

(c)

(d)

Figure 6.17. Compression and tension force magnitudes on a saddle TIC based on tetrahedra. a) Compression forces (ABS plastic), b) Tension forces (ABS plastic), c) Compression forces (Masonry), d) Tension forces (Masonry).

vations from the vertical case. The compression and tension forces follow a different distribution in this case (see Figure 6.23). The top section of the assembly resembles a barrel vault design, as shown in Figure 1.1. Such structures are known to behave as compression-only structures. However, this particular assembly requires tension forces in the middle of the assembly due to the location of the support structure (the endpoints of the cylinder). The bottom sections of the assembly resemble a membrane-like structure. Its blocks require larger tension magnitudes to stay in place and reach an equilibrium state.

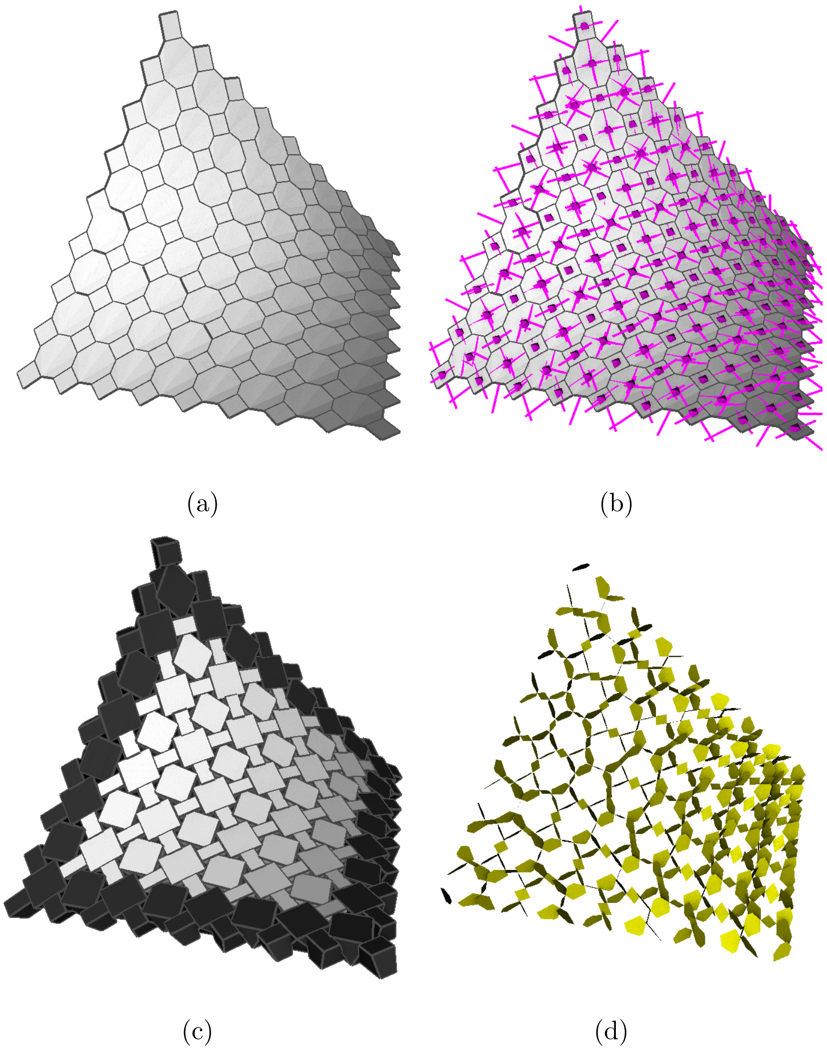
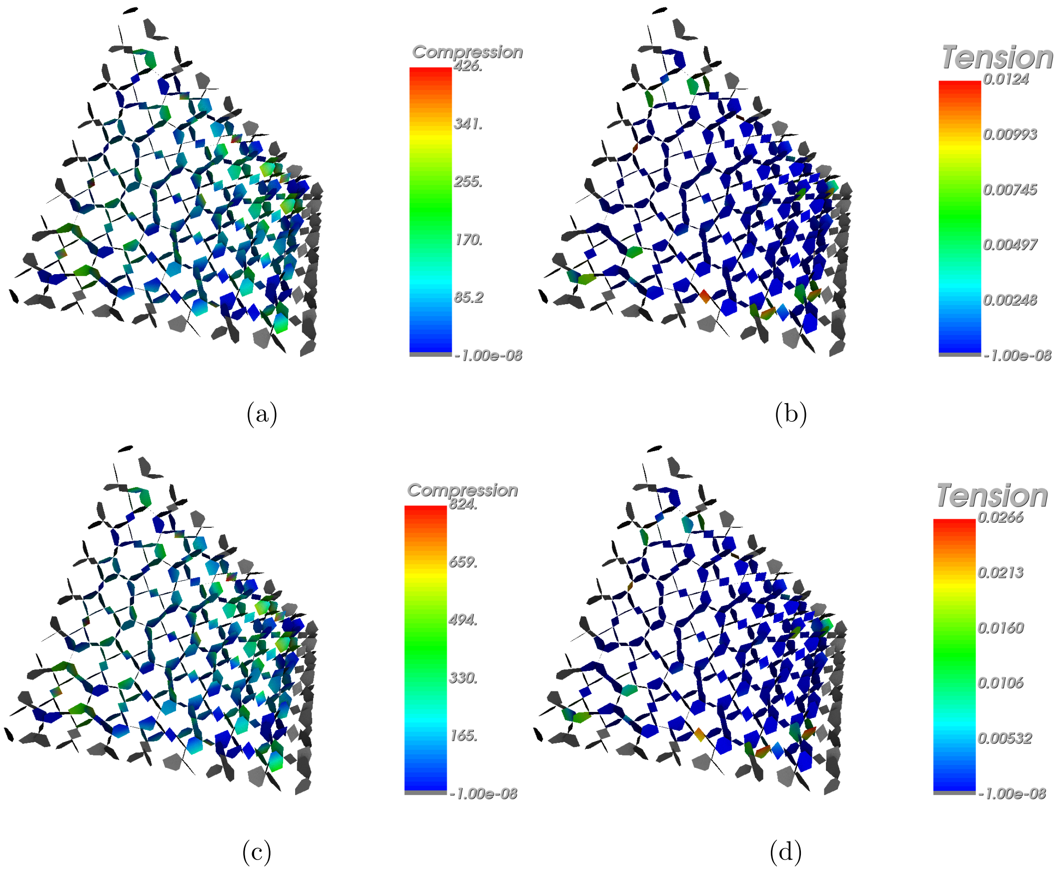

(a)
(b)
(c)
(d)

Figure 6.18. Elements for a saddle TIC based on diverse antiprisms. a) Tessellation, b) Tile centers and edge direction vectors, c) Assembly, d) Interfaces between blocks.

### 6.5.3 Closed TICs

Closed surfaces are relevant to our study since the respective assemblies do not have a boundary. We designed spherical tessellation by subdividing the faces of regular solids. Figure 6.24 comes from an octahedron whose faces were subdivided by midpoints three times, and by quadrilaterals once. Then, we normalize the coordinates of the vertices to the desired radius of the sphere. We generate the blocks of



(a) (b)

(c) (d)

Figure 6.19. Compression and tension force magnitudes on a saddle TIC based on diverse antiprisms. a) Compression forces (ABS plastic), b) Tension forces (ABS plastic), c) Compression forces (Masonry), d) Tension forces (Masonry).

the assembly and clip them using the Tile Offset approach. The result is a smooth spherical assembly.

The Static Equilibrium Analysis requires fixed blocks that act as support. For that purpose, we disable the blocks at the bottom of the assembly. We run the Static Equilibrium Analysis using ABS plastic and masonry parameters, Figure 6.25 shows the required compression and tension magnitudes for the spherical assembly to reach an equilibrium state. The higher compression magnitudes locate at the bottom of the assembly next to the support structure. The blocks around and below the equator of the sphere require higher tensions magnitudes in order to maintain static equilibrium.

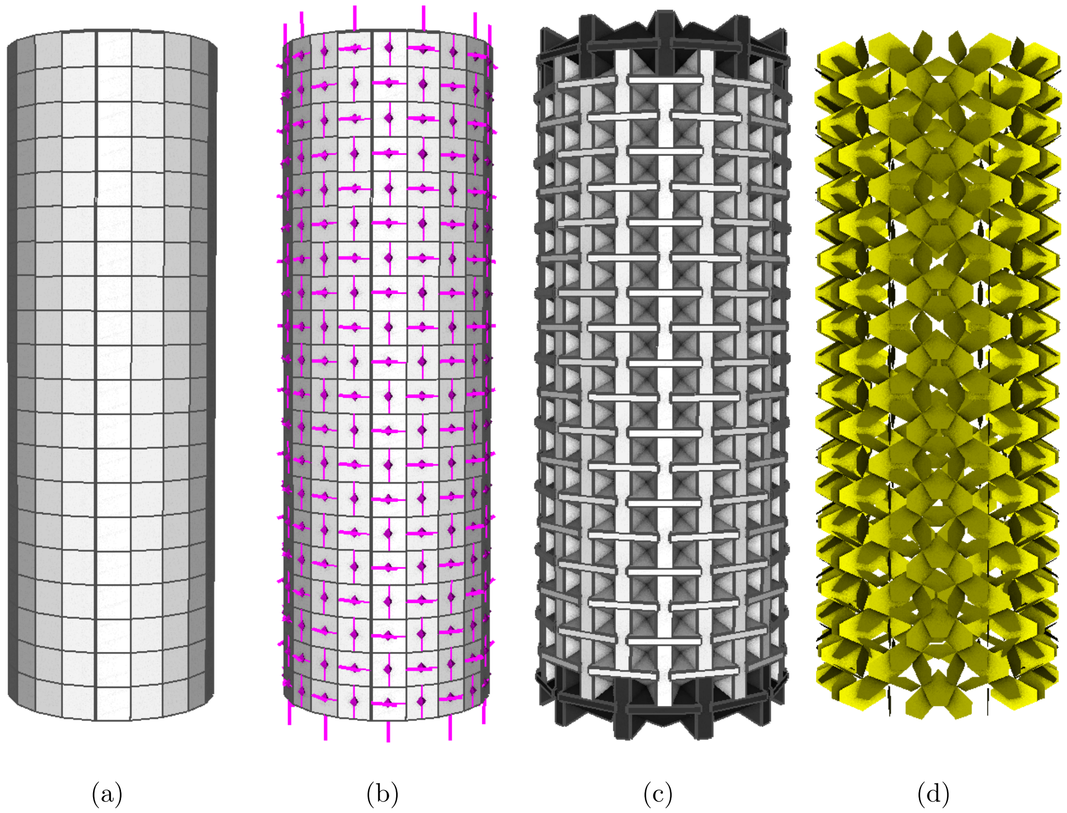
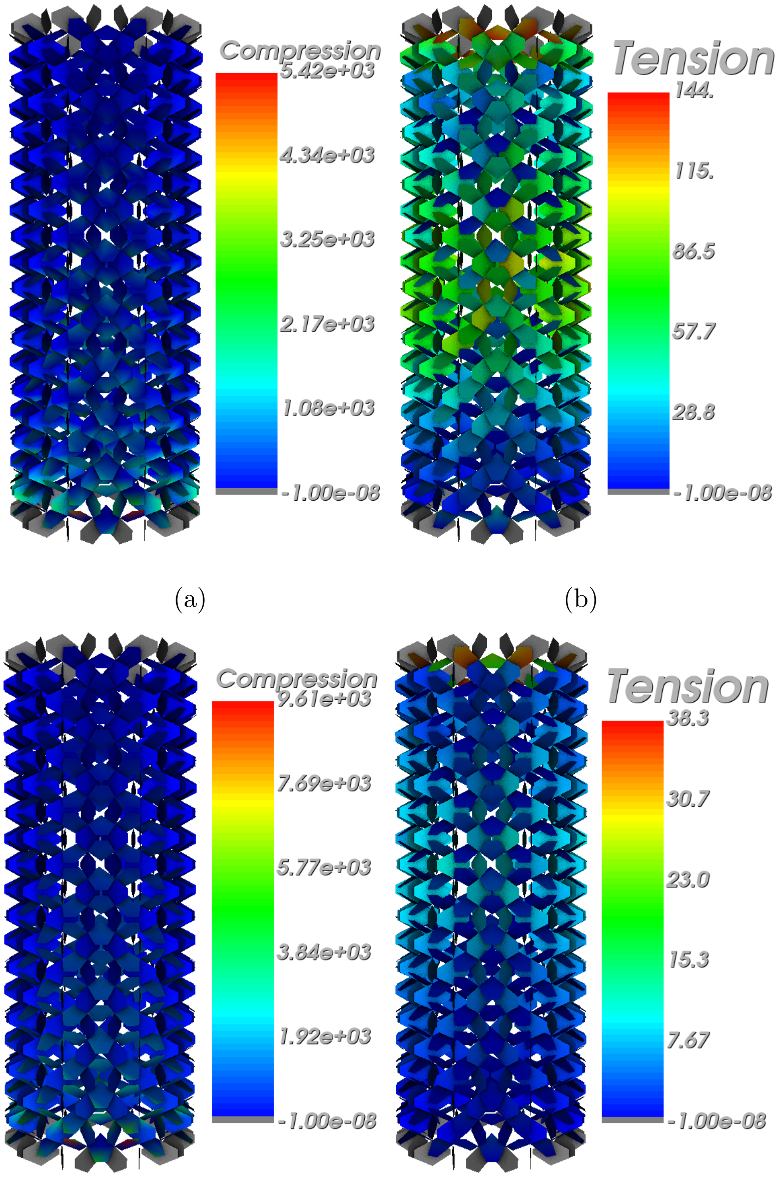


Figure 6.20. Elements for a vertical cylindrical TIC based on clipped tetrahedra. a) Tessellation, b) Tile centers and edge direction vectors, c) Assembly, d) Interfaces between blocks.

The blocks above the equator form a vault structure. As expected, such blocks require less tension magnitudes as the blocks approach the apex of the assembly.

### 6.5.4 Free–Form Based TICs

Finally, we consider TICs based on free–form 3D meshes. Given their non–uniform design, some assemblies require further clipping until all overlapping get removed. As mentioned earlier in this section, we use quadrangulated meshes as the base tessellations. Each face in the mesh corresponds to a tile of the tessellation. We focus our results on two free–form meshes: an open palm hand, and the traditional Stanford Bunny. The hand mesh has sections that resemble some of the TICs previously



(a) (b)

(c) (d)

Figure 6.21. Compression and tension force magnitudes on a vertical cylindrical TIC based on clipped tetrahedra. a) Compression forces (ABS plastic), b) Tension forces (ABS plastic), c) Compression forces (Masonry), d) Tension forces (Masonry).

discussed. The Stanford Bunny is an unofficial benchmark in Computer Graphics

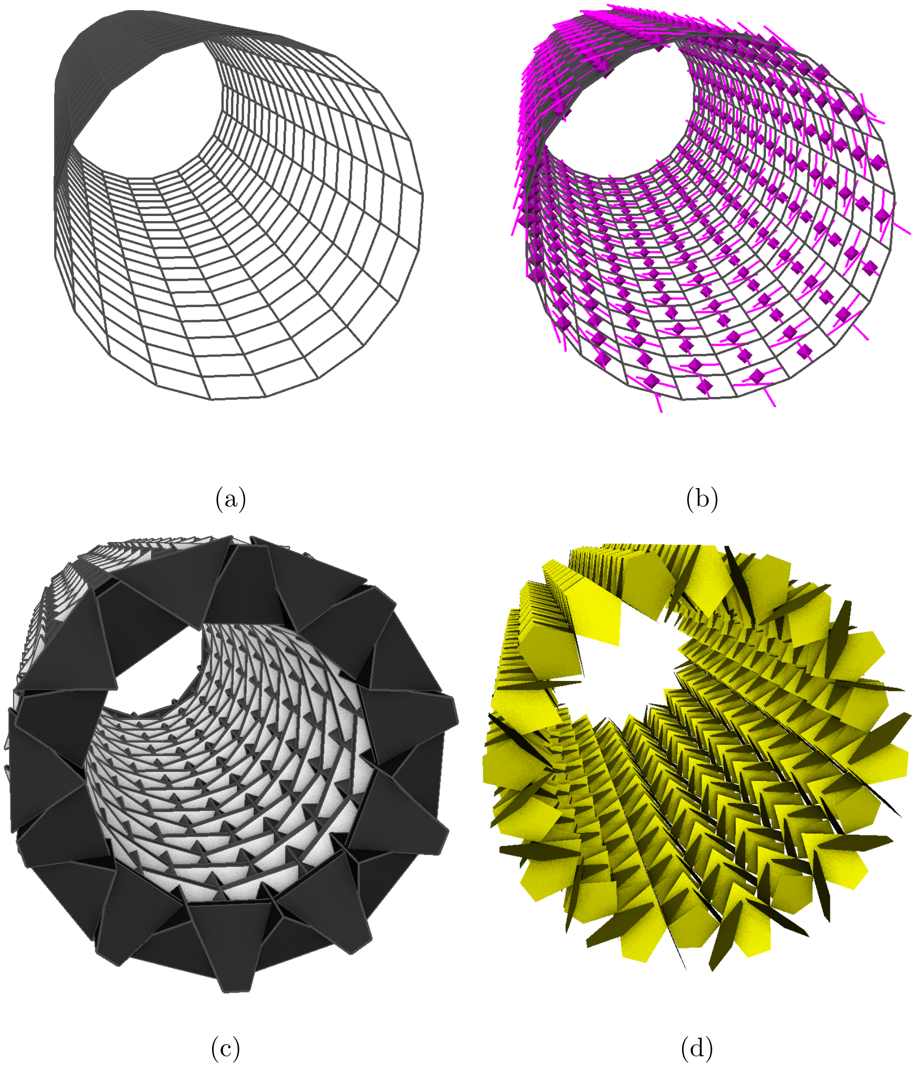


Figure 6.22. Elements for a horizontal cylindrical TIC based on clipped tetrahedra. a) Tessellation, b) Tile centers and edge direction vectors, c) Assembly, d) Interfaces between blocks.

due to its non–trivial geometry and a large number of vertices and faces. The 3D quadrangulated meshes come from the set provided by Fang *et al.* available in [67].

The hand tessellation is open at the bottom. Which means, the resultant assembly will have a support structure only at the bottom of the configuration. Additionally, the hand has pronounced valleys between the fingers and peaks at the fingertips. The blocks of the assembly must fit adequately to describe such regions without

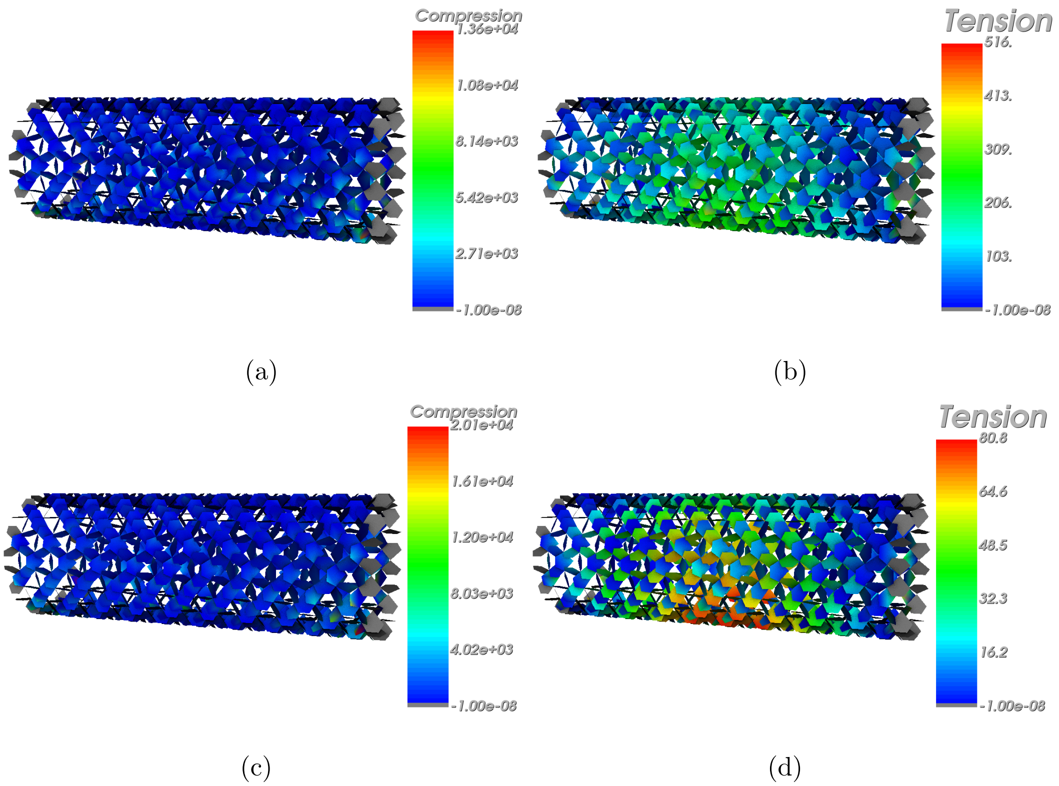


Figure 6.23. Compression and tension force magnitudes on a horizontal cylindrical TIC based on clipped tetrahedra. a) Compression forces (ABS plastic), b) Tension forces (ABS plastic), c) Compression forces (Masonry), d) Tension forces (Masonry).

overlapping. Figure 6.26 shows the elements to generate a TI assembly based on a hand tessellation using clipped tetrahedra. Figure 6.27 shows a zoomed–in section of the assembly for better visualization of the resultant blocks between the thumb, index finger, and palm sections.

The static equilibrium analysis on the assembly indicates that the structure needs high tension magnitudes to reach an equilibrium state. Figure 6.28 shows the compression and tension force magnitudes for such a state. The compression force magnitudes locate at the bottom of the assembly next to the support structure. The tension forces distribute at different locations of the assembly. In particular, the sections of the palm and the valleys between the fingers.

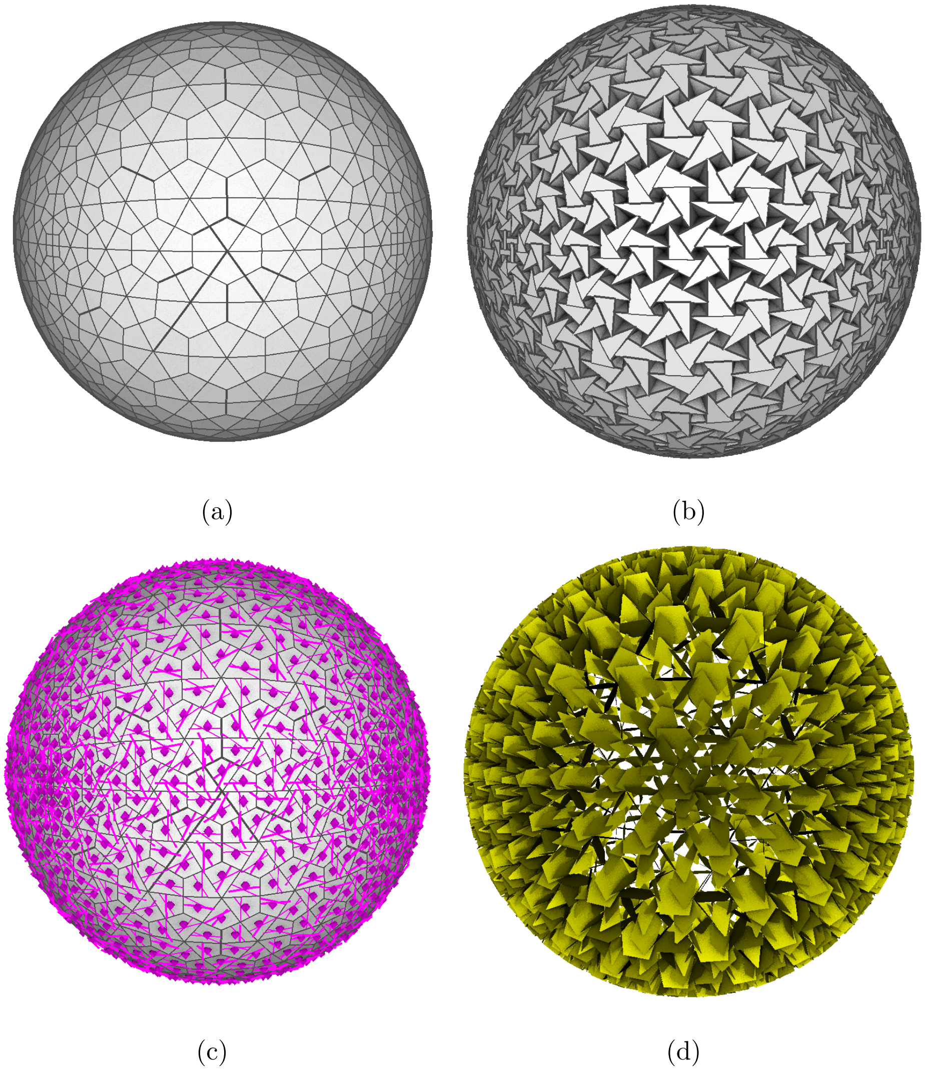


Figure 6.24. Elements for a spherical TIC based on clipped tetrahedra. a) Tessellation, b) Tile centers and edge direction vectors, c) Assembly, d) Interfaces between blocks.

We also considered a different tessellation of the hand mesh. Using the same approach from the second saddle case, the shape of the tiles change from only quadrilaterals to quadrilaterals and other polygons. Large sections of the new tessellation resemble an Archimedean pattern based on squares and octagons. Figure 6.29 shows the elements to generate the respective assembly. Similarly to the original hand tessellation case, the blocks at the bottom of the assembly are the only support structure

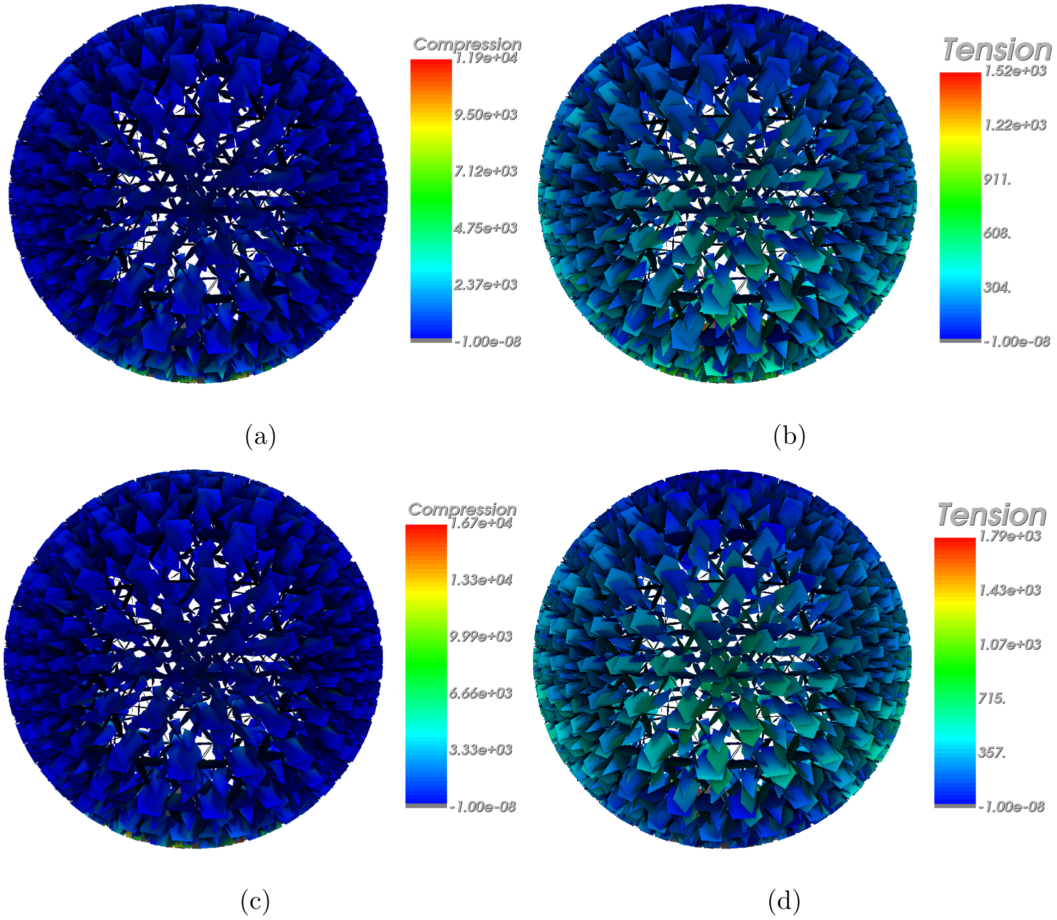
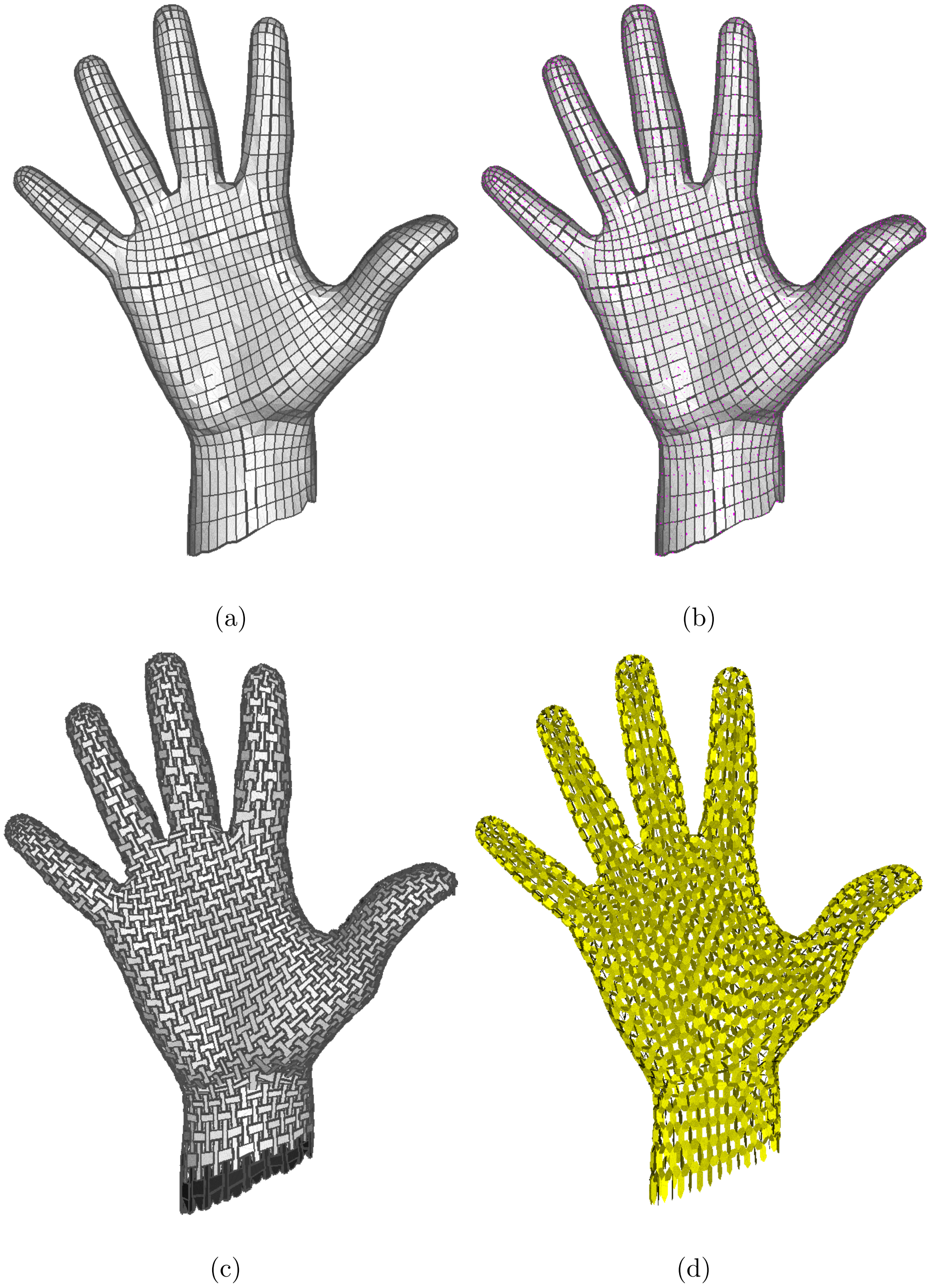


Figure 6.25. Compression and tension force magnitudes on a spherical TIC based on clipped tetrahedra. a) Compression forces (ABS plastic), b) Tension forces (ABS plastic), c) Compression forces (Masonry), d) Tension forces (Masonry).

of the configuration. Figure 6.30 shows the respective section of the assembly that corresponds to the section shown in Figure 6.27.

Figure 6.31 shows the distribution of the compression and tension forces through the assembly. Similarly to the previous case, higher compression forces locate at the bottom of the assembly next to the support structure. The tension forces are distributed throughout the assembly. In this case, the palm section gather most of the tension required to reach an equilibrium state.



(a)

(b)

(c)

(d)

Figure 6.26. Elements for a hand–like TIC based on clipped tetrahedra. a) Tessellation, b) Tile centers and edge direction vectors, c) Assembly, d) Interfaces between blocks.

For the Stanford Bunny mesh, an assembly based on the quadrangulated mesh was reasonable for most of the body (e.g., face, belly, legs, feet, back, and tail).

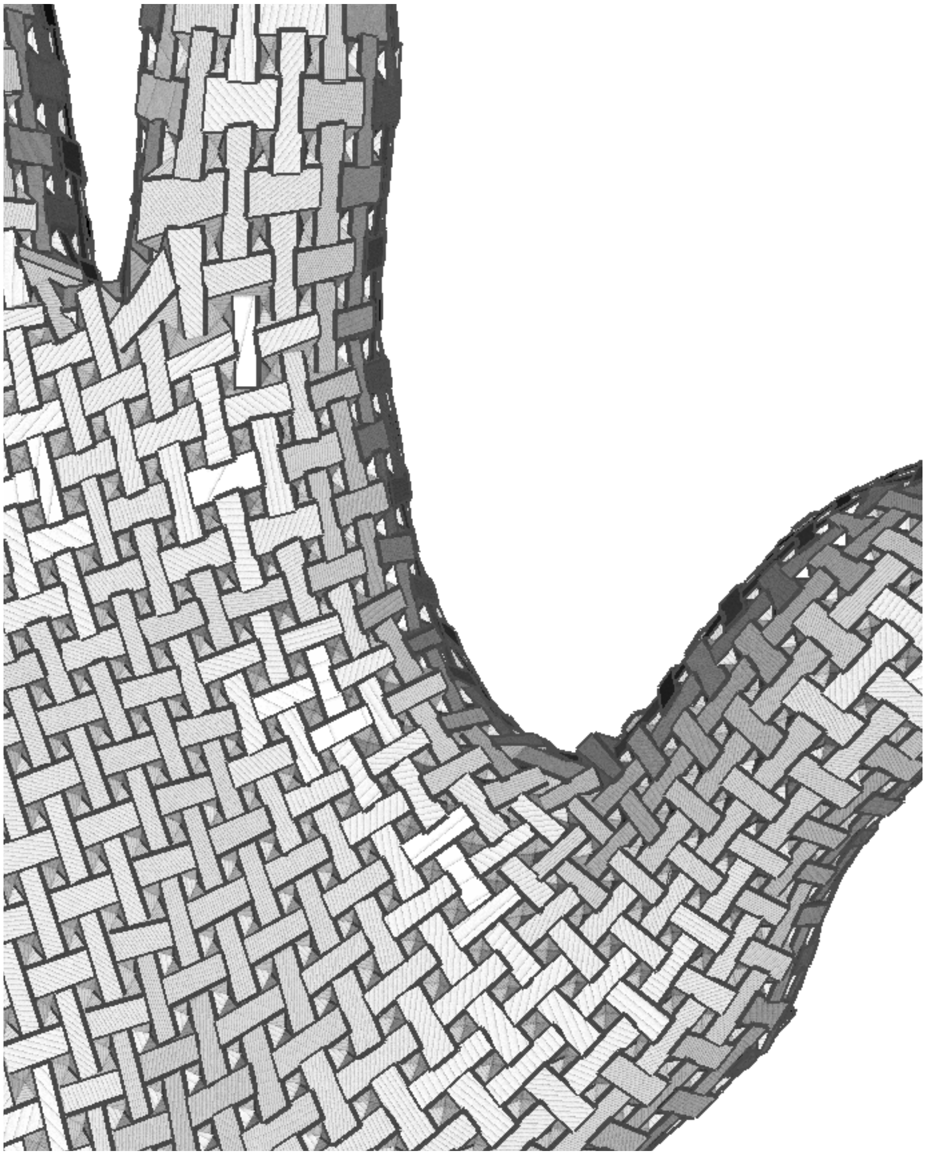
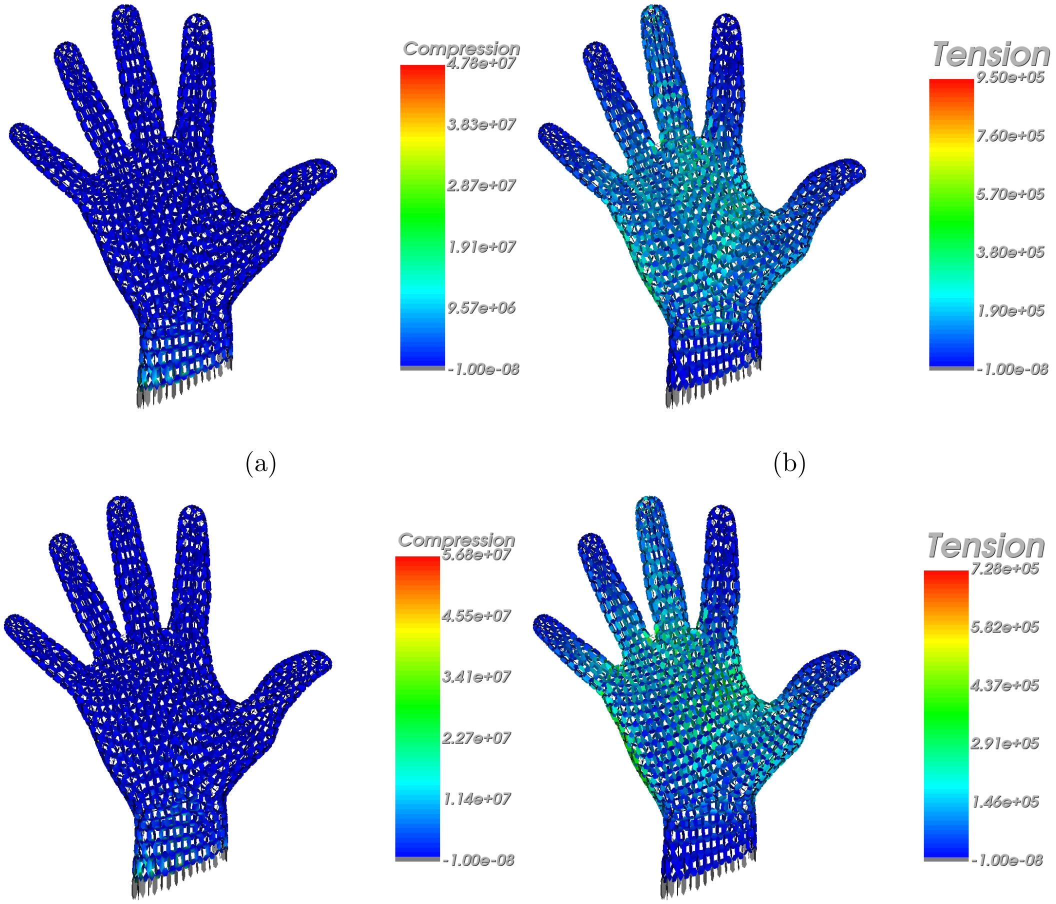


Figure 6.27. Section of the hand–like TIC based on clipped tetrahedra.

However, the blocks for the ears produced artifacts that could not be removed using the proposed pipeline or any of the clipping methods described in Chapter 4. We got satisfactory results when using an Archimedean–like tessellation similar to the second hand–based assembly. Figure 6.32 show the elements to generate the respective assembly. The resultant blocks are clipped tetrahedra, square antiprisms, and decagon antiprisms. Figure 6.33 shows a zoomed–in section of the assembly corresponding to



(a)

(b)

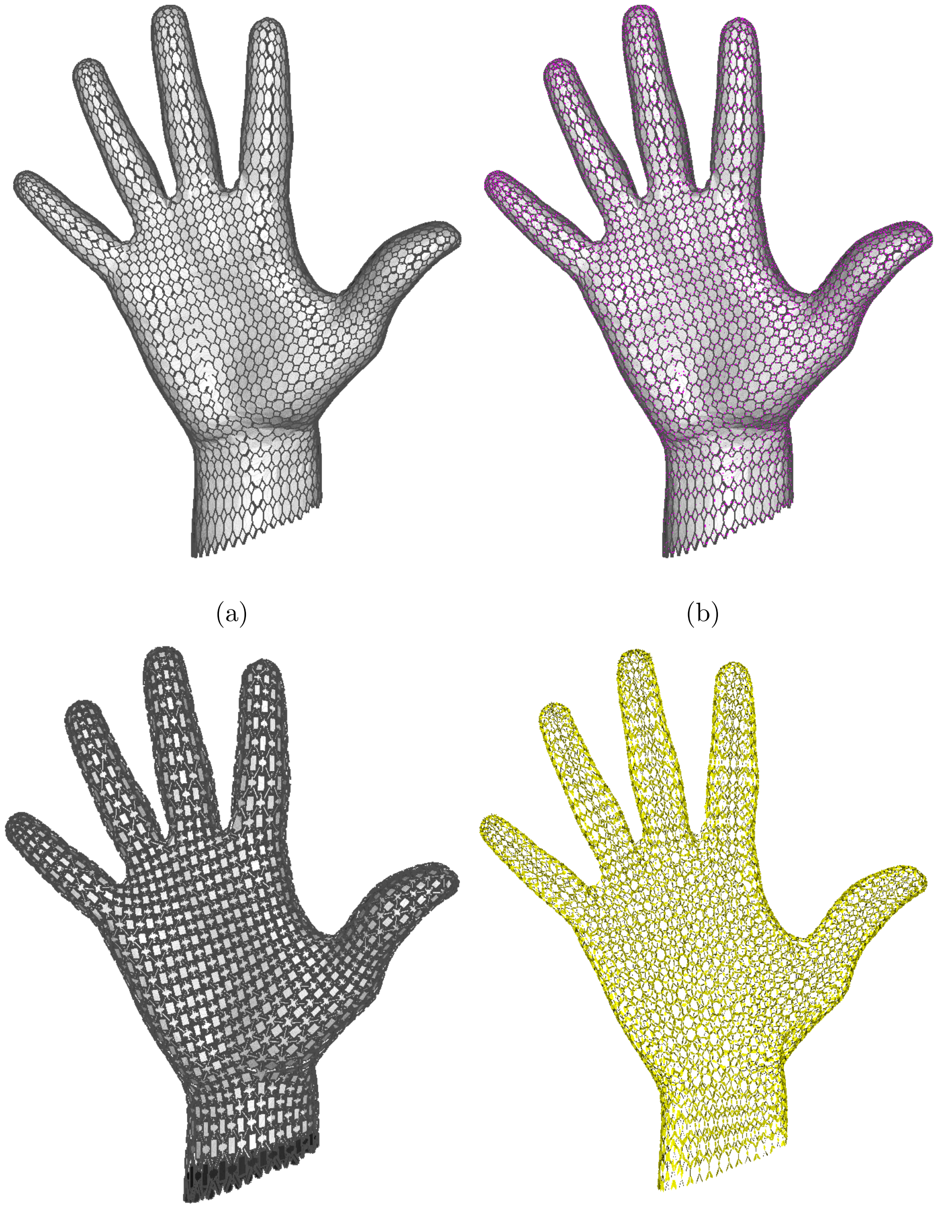
(c)

(d)

Figure 6.28. Compression and tension force magnitudes on a hand–like TIC based on clipped tetrahedra. a) Compression forces (ABS plastic), b) Tension forces (ABS plastic), c) Compression forces (Masonry), d) Tension forces (Masonry)

the face and the base of the ears. Since the bunny is a closed mesh, we used the blocks at the bottom of the assembly as the support structure. This approach is the same we considered for the spherical TIC case.

Figure 6.34 shows the compression and tension force magnitudes required for the assembly to reach an equilibrium state. As expected, the higher compression forces locate at the bottom of the assembly next to the support structure. The tension



(a)

(b)

(c)

(d)

Figure 6.29. Elements for a hand–like TIC based on clipped tetrahedra and antiprisms. a) Tessellation, b) Tile centers and edge direction vectors, c) Assembly, d) Interfaces between blocks.

forces scatter throughout the assembly. In particular, the base of the ears, belly and legs contain most of the required tensions.

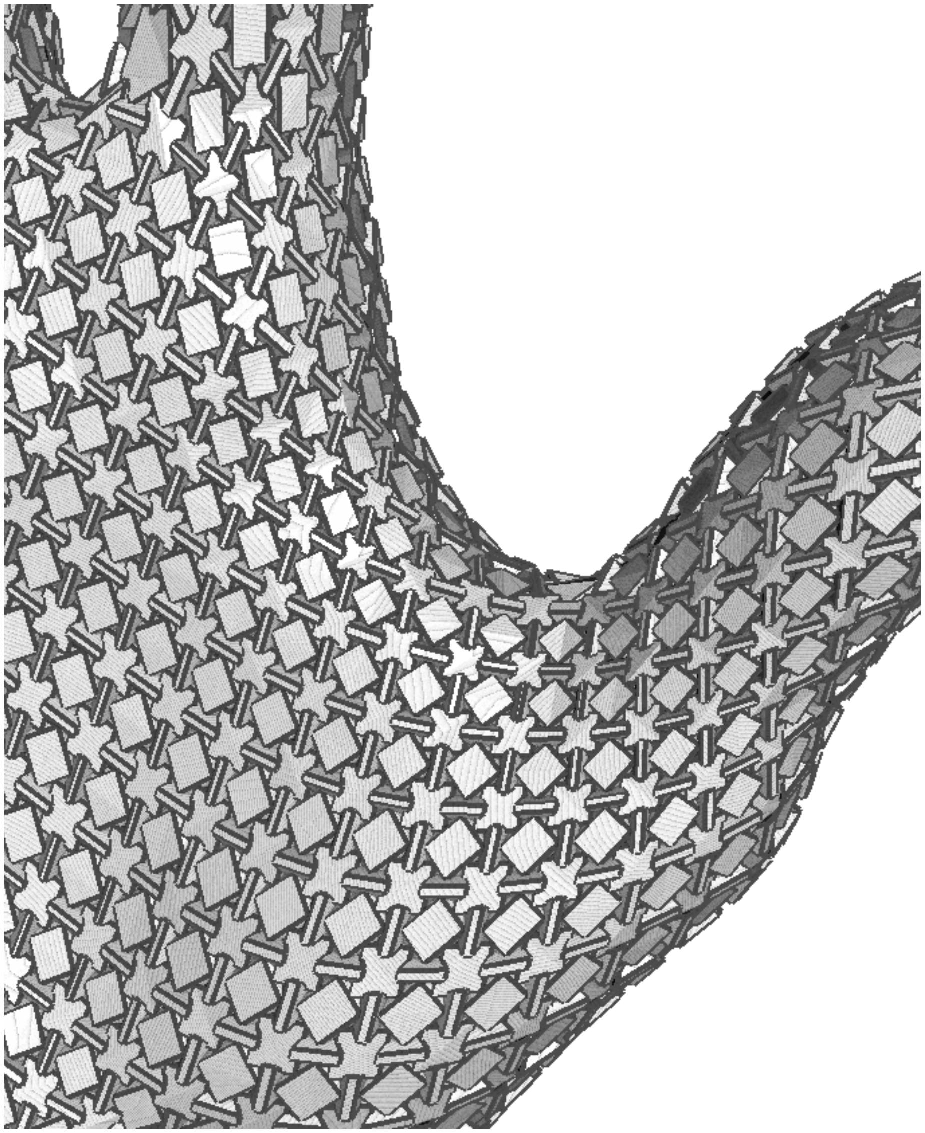
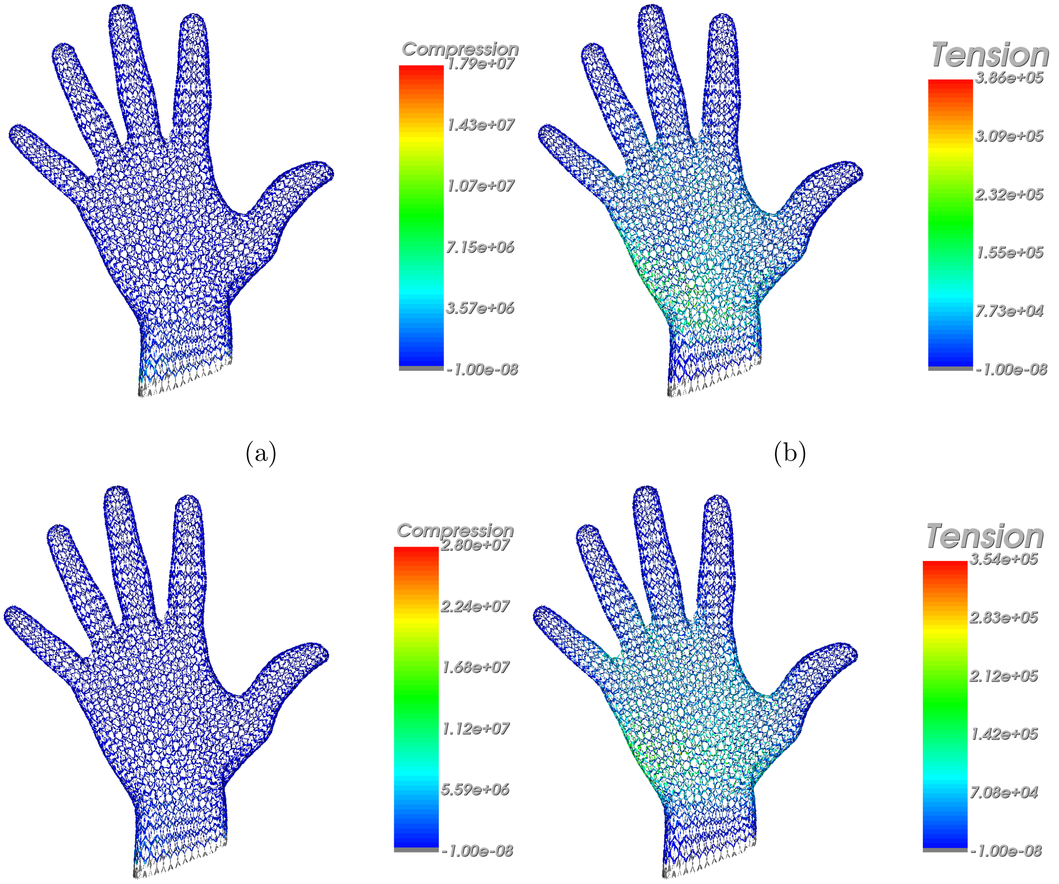


Figure 6.30. Section of the hand–like TIC based on clipped tetrahedra and antiprisms.

## 6.6 Discussion

We have discussed approaches to redesign the tiles of a surface tessellation and analyze the equilibrium of a TI assembly. Changing the tiling of a tessellation allows the generation of diverse block shapes. The static equilibrium analysis returns the



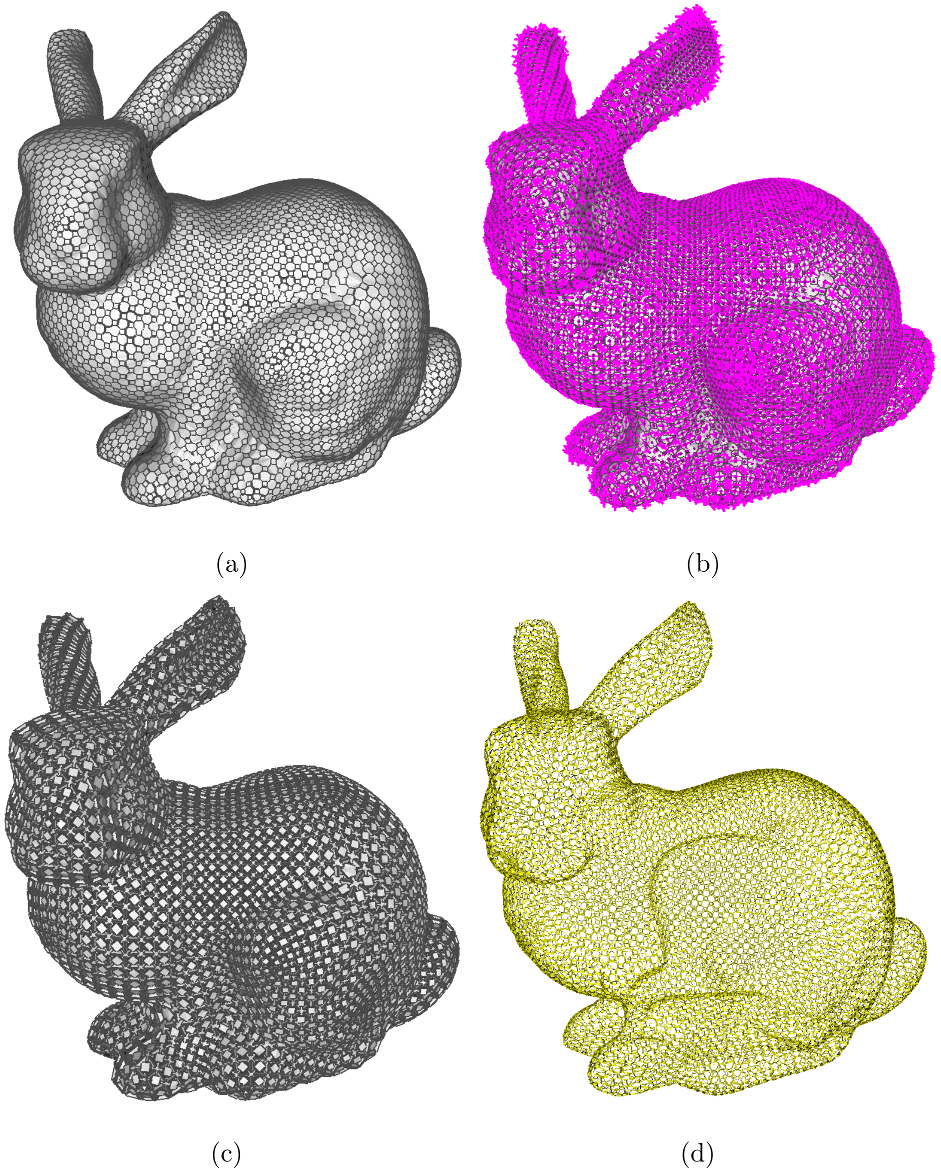
(a) (b)

(c) (d)

Figure 6.31. Compression and tension force magnitudes on a hand–like TIC based on clipped tetrahedra and antiprisms. a) Compression forces (ABS plastic), b) Tension forces (ABS plastic), c) Compression forces (Masonry), d) Tension forces (Masonry)

required forces (compression, tension, and tangential) to maintain an assembly in static equilibrium.

The proposed redesign methods involve subdividing the tiles and the dual surface tessellation. Both approaches alter the tiling of the tessellation, therefore changing the mid-sections of the blocks. By doing so, we expand the catalog of possible TI assemblies that resemble the surface shape. The only requirement to generate a TI assembly is that all tiles must have an even number of sides.



(a)

(b)

(c)

(d)

Figure 6.32. Elements for a bunny TIC based on clipped tetrahedra and antiprisms. a) Tessellation, b) Tile centers and edge direction vectors, c) Assembly, d) Interfaces between blocks.

Finally, the static equilibrium analysis provides a powerful tool to determine the location and magnitude of the forces required for the assembly to be in equilibrium when considered as a compression-only structure. We use the results of such analysis to find the generation parameters that reduce the tension forces. We can determine a

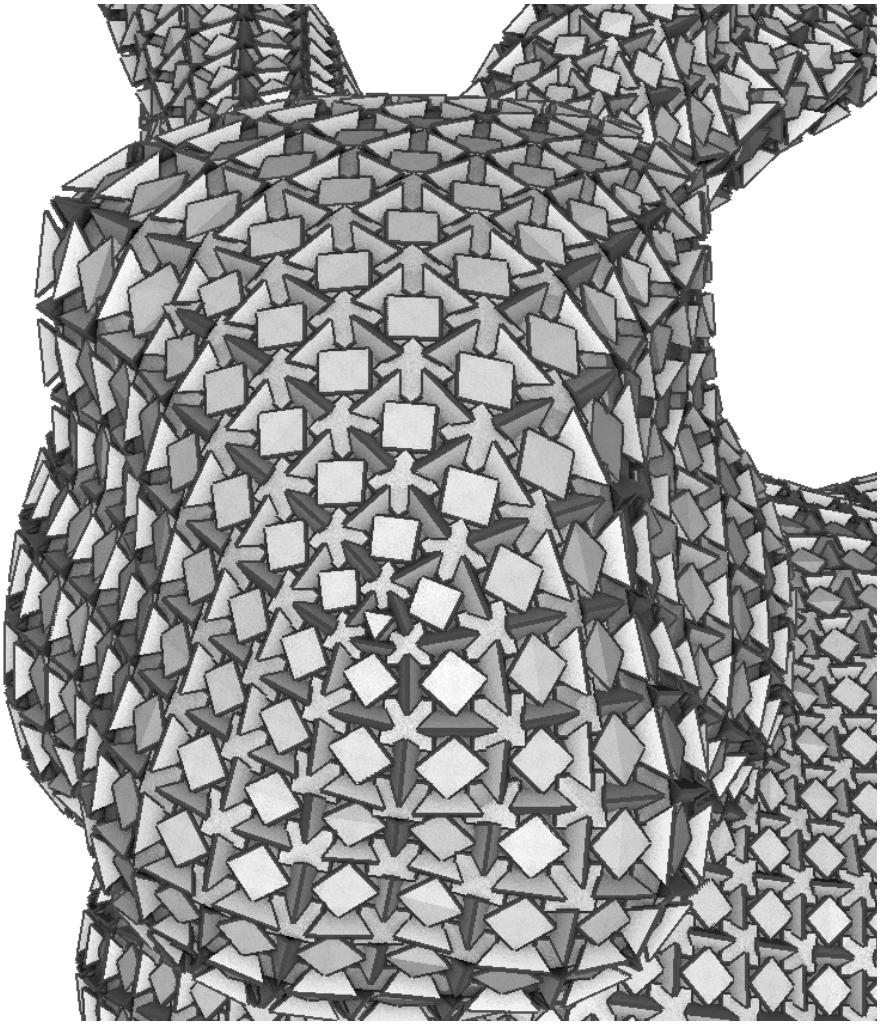
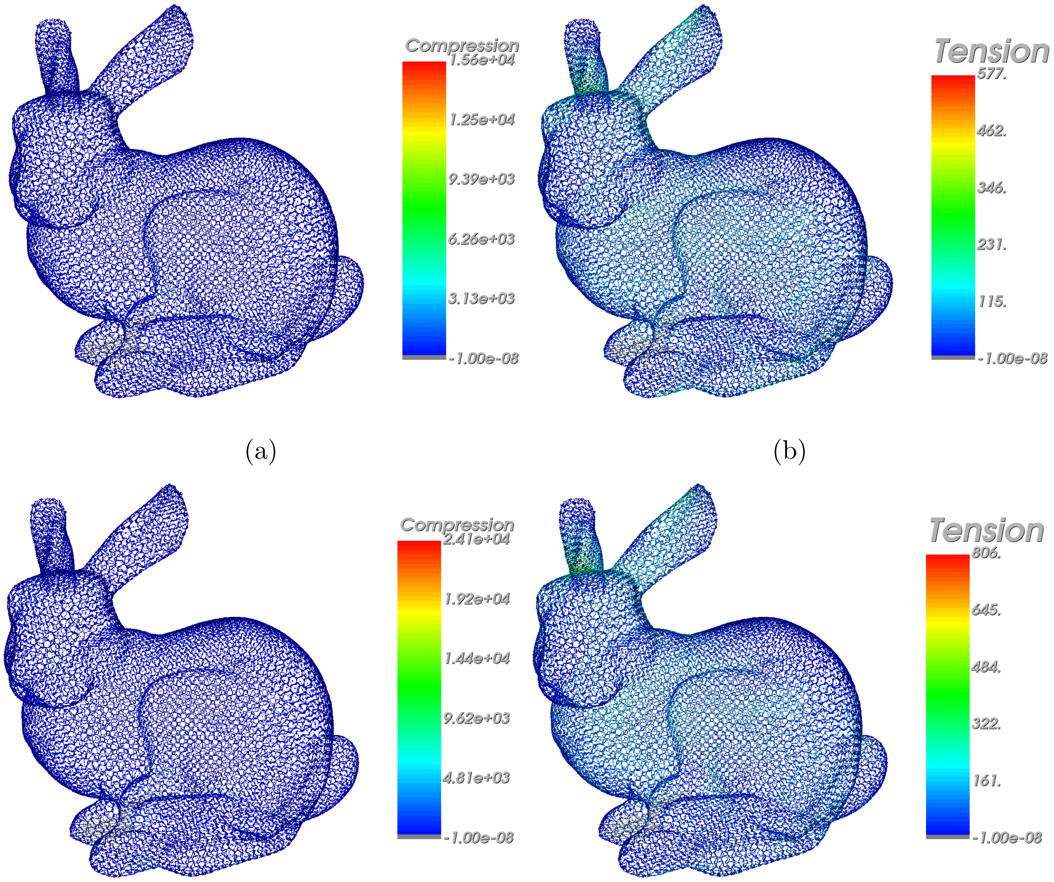


Figure 6.33. Section of the bunny TIC based on clipped tetrahedra and antiprisms.

global interlocking criterion by checking the measure of infeasibility when force loads affect the assembly along with the principal directions $(X, -X, Y, -Y, Z, -Z)$. Some TI assemblies could fulfill such criterion provided a competent support frame that locks the blocks along with the interfaces with larger tension magnitudes.



(a) (b)

(c) (d)

Figure 6.34. Compression and tension force magnitudes on a bunny TIC based on clipped tetrahedra and antiprisms. a) Compression forces (ABS plastic), b) Tension forces (ABS plastic), c) Compression forces (Masonry), d) Tension forces (Masonry)

Table 6.2.
Description of the Static Equilibrium Analysis for the discussed TICs.

| Shape | Tiles | #Blocks | #Intfs. | Material | #Iters. | Dur. (s) |
|---|---|---|---|---|---|---|
| Planar | Squares | 81 | 144 | ABS Plastic | 16 | 0.09 |
| Planar | Squares | 81 | 144 | Masonry | 15 | 0.07 |
| Planar | Archimedean | 145 | 368 | ABS Plastic | 17 | 0.15 |
| Planar | Archimedean | 145 | 368 | Masonry | 17 | 0.14 |
| Saddle | Quads | 81 | 144 | ABS Plastic | 15 | 0.08 |
| Saddle | Quads | 81 | 144 | Masonry | 16 | 0.13 |
| Saddle | Archimedean | 145 | 368 | ABS Plastic | 17 | 0.20 |
| Saddle | Archimedean | 145 | 368 | Masonry | 17 | 0.19 |
| Cyl. (V) | Squares | 400 | 780 | ABS Plastic | 57 | 1.49 |
| Cyl. (V) | Squares | 400 | 780 | Masonry | 42 | 0.87 |
| Cyl. (H) | Squares | 400 | 780 | ABS Plastic | 54 | 1.44 |
| Cyl. (H) | Squares | 400 | 780 | Masonry | 45 | 1.00 |
| Sphere | Quads | 1536 | 3072 | ABS Plastic | 48 | 5.72 |
| Sphere | Quads | 1536 | 3071 | Masonry | 52 | 11.30 |
| Hand | Quads | 2368 | 4715 | ABS Plastic | 114 | 22.86 |
| Hand | Quads | 2368 | 4715 | Masonry | 90 | 18.63 |
| Hand | Archimedean | 4717 | 14068 | ABS Plastic | 124 | 50.62 |
| Hand | Archimedean | 4717 | 14068 | Masonry | 98 | 41.67 |
| Bunny | Archimedean | 13478 | 40420 | ABS Plastic | 85 | 129.05 |
| Bunny | Archimedean | 13478 | 40420 | Masonry | 85 | 126.26 |

# 7 CONCLUSION AND FUTURE WORK

## 7.1 Contributions

This thesis discusses multiple approaches to generate valid TICs based on different types of surface tessellations. Our work introduces three primary contributions to the generation of TI assemblies based on different types of surface tessellation:

1. **Introducing heights as a TIC generation parameters:** We presented the Height–Bisection method to generate TI assemblies, the parameters are the distances from each tile in the tessellation toward its respective top and bottom evolution sections. This method guarantees block alignment and avoids the calculation of the angle values set at the edges of the tiles. Still, we can calculate such angles using the information of the rotated normal vectors at each edge. The Shape–Fit method is a specialization of the HBM on cylindrical tessellations. This method generates two types of blocks: regular tetrahedra and quasi–tetrahedra.

2. **Generalized Mid–Section Evolution:** We introduced a generalized evolution framework that generates a polyhedron as the evolution of its mid-section along with both directions of its normal vector. The evolution is a sweeping procedure that defines the vertices and faces of each block in the assembly. We describe the framework in terms of evolution steps and evolution parameters (angles and distances), the sequence of evolution steps produces the shape of the respective polyhedron. This method provides an algorithmic approach that generates the Platonic solids along with infinitely many TI shapes.

3. **Finding Optimal Generation and Validation Parameters:** We discussed the existence of multiple feasible parameter sets to generate TI assemblies. A

feasible parameter set is one that minimizes the overlapping between blocks in an assembly. The assembly is valid if there is no overlapping. We describe two TIC generation pipelines to find optimal generation parameters. The first pipeline includes feasible parameters along with clipping parameters. The second pipeline includes parameters to run a static equilibrium analysis on the assembly to find the minimum tension force magnitudes.

## 7.2 Future Work

The contributions in this thesis are geometric and algorithmic approaches that deal with the generation of TICs. There are still open questions related to the design of TI assemblies and the resultant physical dynamics of the assembly:

### 7.2.1 Pre–Processing Surface Tessellation for Optimal TICs

Our approaches calculate the shape of the blocks without adjusting the information from the surface tessellations. Meanwhile, some unreinforced masonry structure analysis readjusts the tessellation to generate optimal blocks in terms of specific parameters. It would be valuable to integrate the analysis of the tessellation to the generation pipeline to generate functional assemblies. There are several approaches to deal with the tessellation processing:

- The dihedral angle between two adjacent tiles contributes to the resultant shape of the respective blocks. The smaller the dihedral angle is, the more elongated the resultant blocks become. Including a mesh smoothing processing to the tessellation allows readjusting such angles at the expense of surface details. Such a smoothing method must consider the assembly generation parameters and the shape of the resultant blocks as components of the process.

- Each tile of the tessellation must have an even number of sides for the mentioned generation methods to work. The free–form meshes used to illustrate the resultant assemblies were quadrangulated using the Morse–Parameterization Hybridization method by Fang *et al.* [55]. However, remeshing methods do not return a unique answer. Furthermore, there exist infinitely many tessellations based on even–sided tiles to cover different topologies.

### 7.2.2 Self–Supporting TICs during Assembly Process

The shape of the blocks in a TIC requires using support frames to hold the assembly to prevent lateral strain and reach an equilibrium state. During the assembly process, the support structure must adapt to the current shape of the assembly to keep the blocks in place until the support frame gets closed. Weizmann *et al.* [17] described three different support frames to assist the assembly process of a planar TI assembly (shown in Figure 7.1). Assembling non–planar TICs is a more complicated process that requires more sophisticated solutions.

A potential solution, inspired after the work of Deuss *et al.* [61], would require chains that hold specific blocks during the assembling process. Since the support frame of a TIC could have traversal sections along with the assembly, finding the support structure requires discretizing the configuration into block subsets and consider their assembly requirements.

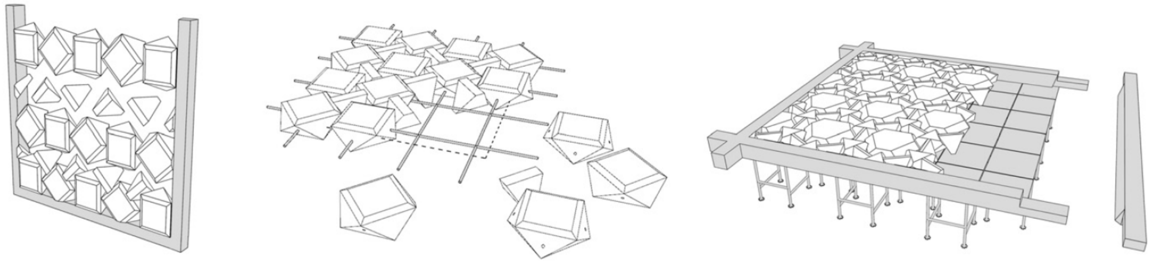


Figure 7.1. TIC assembly methods. Source: Weizmann *et al.* [17]

### 7.2.3 TICs made of Concave Blocks

In Chapter 2, we mentioned TICs made of concave blocks. The usual concave block shapes are the Osteomorphic Block [22] [23] and Tetrahedron–Like shapes [26] [27]. A concave block can lock the motion of more than one piece at a time in a single direction. However, the design of assemblies using such shapes is manual to the best of our knowledge. A design possibility is to design the block shapes using the top and bottom section points (see Section 3.4.1) by connecting them with the vertices of the respective tile. The resultant block aligns with the local curvature described by the neighborhood of a tile.

### 7.2.4 Load Response Design

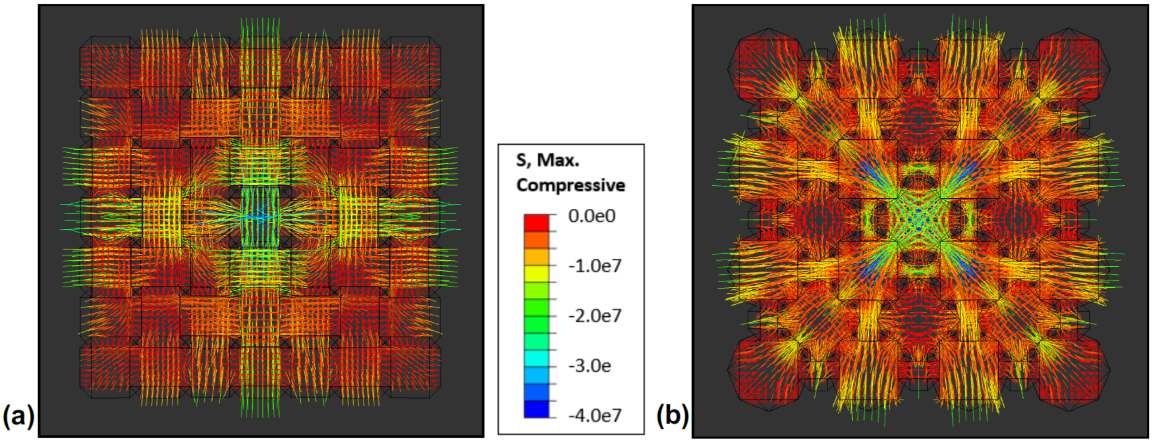


Figure 7.2. Load paths in TIMs. a) Assembly made of truncated tetrahedra, b) Assembly made of cubes and square antiprisms. Source: Williams and Siegmund [68]

An active research topic on TIMs is their load response. Williams and Siegmund in [68] [69] discuss how different planar tessellations correspond to TI assemblies with different responses under loads. Both the shape of the blocks and the boundary conditions generate load paths that flow through the assembly. For example, Figure

7.2 shows load paths flowing through two different TIMs. An important direction for future work is the design of planar, curvilinear, and free–form TI assemblies to manipulate such load paths as a function of the geometry of the blocks.

### 7.2.5 Multi–Layered TI Assemblies

The TI assemblies discussed in this thesis are single–layered. The dimples formed between blocks provide space for infilling elements, which could also be another TIC. Estrin *et al.* in [70] introduced multi–layered TICs as hybrid structures where individual blocks or layers are made of different materials. The geometry of the Osteomorphic Block makes this kind of assemblies feasible (example in Figure 7.3). However, it is not clear how multi–layered TICs made of convex blocks will work. A possible direction could be using the geometry of the dimples to perform a reverse design for additional layers using the General Mid–Section Evolution method discussed in Chapter 5.

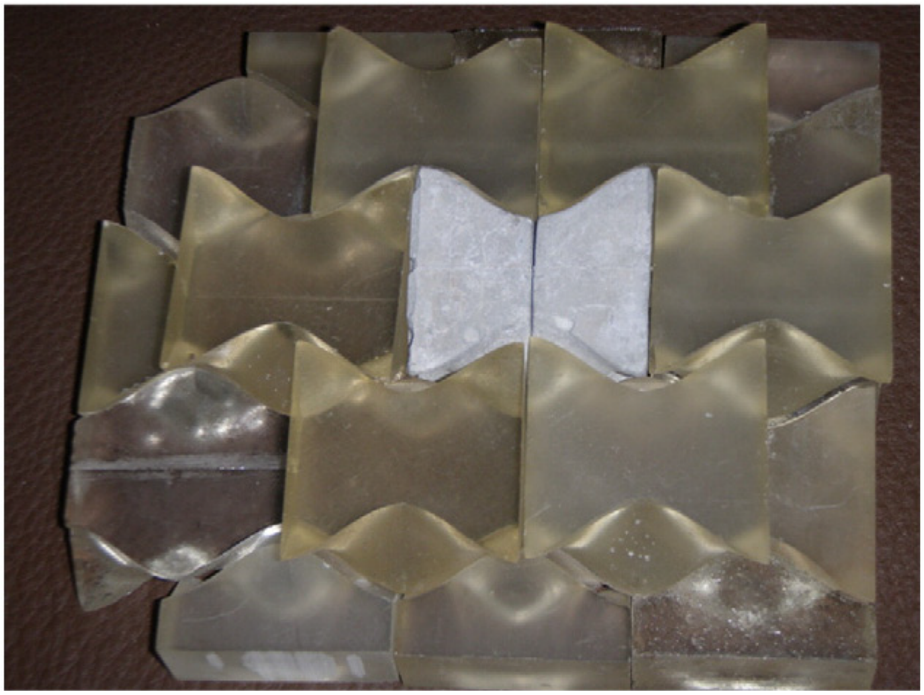


Figure 7.3. Multi–layered TIC based on Osteomorphic Blocks. Source: Estrin *et al.* [70]

# REFERENCES

[1] Giuseppe Fallacara. Digital Stereotomy and topological transformations: reasoning about shape building. In *Proceedings of the 2nd International congress on construction history*, volume 1, pages 1075–1092, 2006.

[2] A. J. Kanel-Belov, A. V. Dyskin, Y. Estrin, E. Pasternak, and I. A. Ivanov-Pogodaev. Interlocking of convex polyhedra: towards a geometric theory of fragmented solids. *arXiv:0812.5089 [math]*, December 2008. arXiv: 0812.5089.

[3] A. V. Dyskin, Y. Estrin, A. J. Kanel-Belov, and E. Pasternak. Topological interlocking of platonic solids: A way to new materials and structures. *Philosophical Magazine Letters*, 83(3):197–203, January 2003.

[4] A. V. Dyskin, Y. Estrin, A. J. Kanel-Belov, and E. Pasternak. Toughening by fragmentation–how topology helps. *Advanced Engineering Materials*, 3(11):885, November 2001.

[5] Emily Whiting, John Ochsendorf, Frdo Durand, Emily Whiting, John Ochsendorf, and Frdo Durand. Procedural modeling of structurally-sound masonry buildings. In *ACM Transactions on Graphics (TOG)*, volume 28, page 112. ACM, December 2009.

[6] A.V Dyskin, Y Estrin, A.J Kanel-Belov, and E Pasternak. A new concept in design of materials and structures: assemblies of interlocked tetrahedron-shaped elements. *Scripta Materialia*, 44(12):2689–2694, June 2001.

[7] Maurizio Brocato. A continuum model of interlocking structural systems. *Rendiconti Lincei - Matematica e Applicazioni*, 29(1):63–83, 2018.

[8] Giuseppe Fallacara. Toward a stereotomic design: experimental constructions and didactic experiences. In *Proceedings of the Third International Congress on Construction History*, pages 553–560, Cottbus, Germany, 2009.

[9] Maurizio Brocato and Lucia Mondardini. Geometric methods and computational mechanics for the design of stone domes based on abeilles bond. In Cristiano Ceccato, Lars Hesselgren, Mark Pauly, Helmut Pottmann, and Johannes Wallner, editors, *Advances in Architectural Geometry 2010*, pages 149–162. Springer Vienna, Vienna, 2010.

[10] M. Brocato and L. Mondardini. A new type of stone dome based on Abeilles bond. *International Journal of Solids and Structures*, 49(13):1786–1801, June 2012.

[11] M. Brocato, W. Deleporte, L. Mondardini, and J.-E. Tanguy. A Proposal for a New Type of Prefabricated Stone Wall. *International Journal of Space Structures*, 29(2):97–112, June 2014.

[12] Michael Glickman. The G-block system of vertically interlocking paving. In *Proceedings of the 2nd International Conference on Concrete Block Paving*, page 4, Delft, Netherlands, April 1984.

[13] A. V. Dyskin, Y. Estrin, A. J. Kanel-Belov, and E. Pasternak. A new class of composite materials based on topological interlocking. In *Applied Mechanics*, pages 485–490, Sydney, Australia, February 2002. WORLD SCIENTIFIC.

[14] A.V. Dyskin, Y. Estrin, A.J. Kanel-Belov, and E. Pasternak. A new principle in design of composite materials: reinforcement by interlocked elements. *Composites Science and Technology*, 63(3-4):483–491, February 2003.

[15] A.V. Dyskin, Y. Estrin, A.J. Kanel-Belov, and E. Pasternak. Interlocking properties of buckyballs. *Physics Letters A*, 319(3-4):373–378, December 2003.

[16] Michael Weizmann, Oded Amir, and Yasha Jacob Grobman. Topological Interlocking in Architectural Design. In *Emerging Experience in Past, Present and Future of Digital Architecture, Proceedings of the 20th International Conference of the Association for Computer-Aided Architectural Design Research in Asia (CAADRIA 2015) / Daegu 20-22 May 2015, pp. 107-116*. CUMINCAD, 2015.

[17] Michael Weizmann, Oded Amir, and Yasha Jacob Grobman. Topological interlocking in buildings: A case for the design and construction of floors. *Automation in Construction*, 72:18–25, December 2016.

[18] Michael Weizmann, Oded Amir, and Yasha Jacob Grobman. Topological interlocking in architecture: A new design method and computational tool for designing building floors. *International Journal of Architectural Computing*, 15(2):107–118, June 2017.

[19] Irina Miodragovic Vella and Toni Kotnik. Geometric Versatility of Abeille Vault - A Stereotomic Topological Interlocking Assembly. In *Herneoja, Aulikki; Toni sterlund and Piia Markkanen (eds.), Complexity & Simplicity - Proceedings of the 34th eCAADe Conference - Volume 2, University of Oulu, Oulu, Finland, 22-26 August 2016, pp. 391-397*. CUMINCAD, 2016.

[20] Oliver Tessmann and Andrea Rossi. Parametric and Combinatorial Topological Interlocking Assemblies. In Thomas Siegmund and Francois Barthelat, editors, *Proceedings of the IUTAM Symposium Architectured Materials Mechanics*, Gleacher Center, Chicago, September 2018. Purdue University Libraries Scholarly Publishing Services, 2018.

[21] Oliver Tessmann and Andrea Rossi. Geometry as Interface: Parametric and Combinatorial Topological Interlocking Assemblies. *Journal of Applied Mechanics*, 86(11):111002, November 2019.

[22] Han Chuan Khor, Arcady Dyskin, Elena Pasternak, Yuri Estrin, and A. J. Kanel-Belov. Integrity and fracture of plate-like assemblies of topologically interlocked elements. *Structural Integrity and Fracture*, pages 449–456, 2002.

[23] Y. Estrin, A. V. Dyskin, A. J. Kanel-Belov, and E. Pasternak. Materials with Novel Architectonics: Assemblies of Interlocked Elements. In G. M. L. Gladwell and B. L. Karihaloo, editors, *IUTAM Symposium on Analytical and Computational Fracture Mechanics of Non-Homogeneous Materials*, volume 97, pages 51–55. Springer Netherlands, Dordrecht, 2002.

[24] A.V. Dyskin, Y. Estrin, E. Pasternak, H.C. Khor, and A.J. Kanel-Belov. Fracture Resistant Structures Based on Topological Interlocking with Non-planar Contacts. *Advanced Engineering Materials*, 5(3):116–119, March 2003.

[25] Oliver Tessmann. Interlocking Manifold Kinematically Constrained Multi-material Systems. In Lars Hesselgren, Shrikant Sharma, Johannes Wallner, Niccolo Baldassini, Philippe Bompas, and Jacques Raynaud, editors, *Advances in Architectural Geometry 2012*, pages 269–278. Springer Vienna, Vienna, 2013.

[26] Oliver Tessmann. Topological Interlocking Assemblies. In *Physical Digitality: Proceedings of the 30th eCAADe Conference*, volume 2, pages 211–219, Prague, Czech Republic: Czech Technical University in Prague, Faculty of Architecture, 2012.

[27] Oliver Tessmann and Mirco Becker. Extremely heavy and incredibly light: performative assemblies in dynamic environments. In *18th International Conference on Computer-Aided Architectural Design Research in Asia: Open Systems, CAADRIA 2013*, pages 469–478, The Association for Computer-Aided Architectural Design Research in Asia (CAADRIA), Hong Kong, and Center for Advanced Studies in Architecture (CASA), Department of Architecture, School of Design and Environment, National University of Singapore, Singapore, May 2013.

[28] Thomas Siegmund, Francois Barthelat, Raymond Cipra, Ed Habtour, and Jaret Riddick. Manufacture and Mechanics of Topologically Interlocked Material Assemblies. *Applied Mechanics Reviews*, 68(4):040803, July 2016.

[29] A. V. Dyskin, Yuri Estrin, and E. Pasternak. Topological Interlocking Materials. In Yuri Estrin, Yves Brchet, John Dunlop, and Peter Fratzl, editors, *Architectured Materials in Nature and Engineering*, volume 282, pages 23–49. Springer International Publishing, Cham, 2019.

[30] Adam Mather, Raymond Cipra, and Thomas Siegmund. Structural integrity during remanufacture of a topologically interlocked material. *International Journal of Structural Integrity*, 3(1):61–78, March 2012.

[31] S. Khandelwal, T. Siegmund, R.J. Cipra, and J.S. Bolton. Transverse loading of cellular topologically interlocked materials. *International Journal of Solids and Structures*, 49(18):2394–2403, September 2012.

[32] S Khandelwal, T Siegmund, R J Cipra, and J S Bolton. Adaptive mechanical properties of topologically interlocking material systems. *Smart Materials and Structures*, 24(4):045037, April 2015.

[33] Yuezhong Feng, Thomas Siegmund, Ed Habtour, and Jaret Riddick. Impact mechanics of topologically interlocked material assemblies. *International Journal of Impact Engineering*, 75:140–149, January 2015.

[34] Mohammad Mirkhalaf, Tao Zhou, and Francois Barthelat. Simultaneous improvements of strength and toughness in topologically interlocked ceramics. *Proceedings of the National Academy of Sciences*, 115(37):9128–9133, September 2018.

[35] Mohammad Mirkhalaf, Tao Zhou, Florent Hannard, and Francois Barthelat. Strong and Tough Ceramics Using Architecture and Topological Interlocking. In Thomas Siegmund and Francois Barthelat, editors, *Proceedings of the IUTAM Symposium Architectured Materials Mechanics*, Gleacher Center, Chicago, September 2018. Purdue University Libraries Scholarly Publishing Services, 2018.

[36] Michael Weizmann, Amir Oded, and Yasha Jacob Grobman. Structural Performance of Semi-regular Topological Interlocking Assemblies. In *Symposium on Simulation for Architecture and Urban Design SimAUD 2019*, pages 217–223, School of Architecture, College of Design, Georgia Institute of Technology, Atlanta, Georgia, April 2019. Society for Modeling & Simulation International (SCS).

[37] C.G. Armstrong, T.K.H. Tam, D.J. Robinson, R.M. McKeag, and M.A. Price. Automatic Generation of Well Structured Meshes Using Medial Axis and Surface Subdivision. In *Proc. 17th ASME Design Automation Conf.*, volume 2, pages 139–146, Miami. Fla., September 1991. New York: ASME.

[38] Aristides Requicha and Herbert Voelcker. Constructive Solid Geometry, November 1977.

[39] Ming Lin and Stefan Gottschalk. Collision detection between geometric models: A survey. *IMA Conference on Mathematics of Surfaces*, 8:37–56, 1998.

[40] P. Jimnez, F. Thomas, and C. Torras. 3d collision detection: a survey. *Computers & Graphics*, 25(2):269–285, 2001.

[41] Christer Ericson. *Real-time collision detection*. Morgan Kaufmann series in interactive 3D technology. Elsevier, Amsterdam ; Boston, 2005.

[42] Daniel N. Wilke, N. Govender, Patrick Pizette, and N. E. Abriak. Computing with non-convex polyhedra on the gpu. In Xikui Li, Yuntian Feng, and Graham Mustoe, editors, *Proceedings of the 7th International Conference on Discrete Element Methods*, volume 188, pages 1371–1377. Springer Singapore, Singapore, 2017.

[43] Ren Weller, Nicole Debowski, and Gabriel Zachmann. kDet: Parallel Constant Time Collision Detection for Polygonal Objects. *Computer Graphics Forum*, 36(2):131–141, May 2017.

[44] Nicolin Govender, Daniel N. Wilke, and Schalk Kok. Collision detection of convex polyhedra on the NVIDIA GPU architecture for the discrete element method. *Applied Mathematics and Computation*, 267:810–829, September 2015.

[45] Robert Nurnberg. Calculating the volume and centroid of a polyhedron in 3d, 2013.

[46] Paul Bourke. Polygons and meshes.

[47] Sébastien Loriot, Jane Tournois, and Ilker O. Yaz. Polygon mesh processing. In *CGAL User and Reference Manual*. CGAL Editorial Board, 4.14.1 edition, 2019.

[48] David Eberly. Clipping a Mesh Against a Plane, March 2008.

[49] Aaron Scherzinger, Tobias Brix, and Klaus H. Hinrichs. An Efficient Geometric Algorithm for Clipping and Capping Solid Triangle Meshes:. In *Proceedings of the 12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, pages 187–194, Porto, Portugal, 2017. SCITEPRESS - Science and Technology Publications.

[50] Mario Botsch, editor. *Polygon mesh processing.* A K Peters, Natick, Mass, 2010. OCLC: ocn423214772.

[51] David Bommes, Henrik Zimmer, and Leif Kobbelt. Mixed-integer quadrangulation. *ACM Transactions on Graphics*, 28(3):1, July 2009.

[52] Wenzel Jakob, Marco Tarini, Daniele Panozzo, and Olga Sorkine-Hornung. Instant field-aligned meshes. *ACM Transactions on Graphics*, 34(6):1–15, October 2015.

[53] Omri Azencot, Etienne Corman, Mirela Ben-Chen, and Maks Ovsjanikov. Consistent functional cross field design for mesh quadrangulation. *ACM Transactions on Graphics*, 36(4):1–13, July 2017.

[54] Jingwei Huang, Yichao Zhou, Matthias Niessner, Jonathan Richard Shewchuk, and Leonidas J. Guibas. QuadriFlow: A Scalable and Robust Method for Quadrangulation. *Computer Graphics Forum*, 37(5):147–160, August 2018.

[55] Xianzhong Fang, Hujun Bao, Yiying Tong, Mathieu Desbrun, and Jin Huang. Quadrangulation through morse-parameterization hybridization. *ACM Transactions on Graphics*, 37(4):1–15, July 2018.

[56] Mario Botsch, Mark Pauly, Leif Kobbelt, Pierre Alliez, Bruno Lvy, Stephan Bischoff, and Christian Rssl. Geometric modeling based on polygonal meshes. In *ACM SIGGRAPH 2007 courses on - SIGGRAPH '07*, page 1, San Diego, California, 2007. ACM Press.

[57] David Bommes, Bruno Lvy, Nico Pietroni, Enrico Puppo, Claudio Silva, Marco Tarini, and Denis Zorin. Quad-Mesh Generation and Processing: A Survey: Quad-Mesh Generation and Processing. *Computer Graphics Forum*, 32(6):51–76, September 2013.

[58] Christoph M. Hoffmann. Geometric Approaches to Mesh Generation. In Avner Friedman, Willard Miller, Ivo Babuska, William D. Henshaw, Joseph E. Oliger, Joseph E. Flaherty, John E. Hopcroft, and Tayfun Tezduyar, editors, *Modeling, Mesh Generation, and Adaptive Numerical Methods for Partial Differential Equations*, volume 75, pages 31–51. Springer New York, New York, NY, 1995.

[59] R. K. Livesley. Limit analysis of structures formed from rigid blocks. *International Journal for Numerical Methods in Engineering*, 12(12):1853–1871, 1978.

[60] Emily Whiting, Hijung Shin, Robert Wang, John Ochsendorf, and Frdo Durand. Structural Optimization of 3d Masonry Buildings. *ACM Trans. Graph.*, 31(6):159:1–159:11, November 2012.

[61] Mario Deuss, Daniele Panozzo, Emily Whiting, Yang Liu, Philippe Block, Olga Sorkine-Hornung, and Mark Pauly. Assembling Self-supporting Structures. *ACM Trans. Graph.*, 33(6):214:1–214:10, November 2014.

[62] Ursula Frick, Tom Van Mele, and Philippe Block. Decomposing Three-Dimensional Shapes into Self-supporting, Discrete-Element Assemblies. In Mette Ramsgaard Thomsen, Martin Tamke, Christoph Gengnagel, Billie Faircloth, and Fabian Scheurer, editors, *Modelling Behaviour*, pages 187–201. Springer International Publishing, Cham, 2015.

[63] Pedro Gmez-Glvez, Pablo Vicente-Munuera, Antonio Tagua, Cristina Forja, Ana M. Castro, Marta Letrn, Andrea Valencia-Expsito, Clara Grima, Marina Bermdez-Gallardo, scar Serrano-Prez-Higueras, Florencia Cavodeassi, Sol Sotillos, Mara D. Martn-Bermudo, Alberto Mrquez, Javier Buceta, and Luis M. Escudero. Scutoids are a geometrical solution to three-dimensional packing of epithelia. *Nature Communications*, 9(1):2960, December 2018.

[64] Sai Ganesh Subramanian, Mathew Eng, Vinayak R. Krishnamurthy, and Ergun Akleman. Delaunay Lofts: A biologically inspired approach for modeling space filling modular structures. *Computers & Graphics*, 82:73–83, August 2019.

[65] Yifang Liu, Maira Saboia, Vivek Thangavelu, and Nils Napp. Approximate Stability Analysis for Drystacked Structures. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8819–8824, Montreal, QC, Canada, May 2019. IEEE.

[66] Gurobi Optimization LLC. Gurobi optimizer reference manual, 2018.

[67] Xianzhong Fang, Hujun Bao, Yiying Tong, Mathieu Desbrun, and Jin Huang. Quadrangulation through Morse-Parameterization Hybridization Supplemental Material, 2018.

[68] Andrew Williams and Thomas Siegmund. Tesselations and Percolations in Topologically Interlocked Stereotomic Material Systems. In Thomas Siegmund and Francois Barthelat, editors, *Proceedings of the IUTAM Symposium Architectured Materials Mechanics*, Gleacher Center, Chicago, September 2018. Purdue University Libraries Scholarly Publishing Services, 2018.

[69] A Williams and T Siegmund. Tiling patterns and the mechanical properties of topologically interlocked materials. In *Proceedings of the 56th Annual Technical Meeting of the Society of Engineering Science*, 2019.

[70] Y. Estrin, A.V. Dyskin, and E. Pasternak. Topological interlocking as a material design concept. *Materials Science and Engineering: C*, 31(6):1189–1194, August 2011.

# A APPENDIX

## Evolution Step Parameters for the Platonic Solids

We detail the evolution step parameters to generate the Platonic Solids. In all cases we refer to the respective polygon with side length $l$ to have centroid located at point $O$ and located in a reference frame defined by normalized orthogonal vectors $U, V, N$ with $V = U \times N$.

### Tetrahedron

The tetrahedron results from the evolution of a square with side length $l$ into a line segment through one evolution step using the double direction evolution method.

Let $A, B, C, D$ be the vertices of a tetrahedron defined as:

$$\begin{aligned} A &= O + \lambda N + lU \\ B &= O + \lambda N - lU \\ C &= O - \lambda N + lV \\ D &= O - \lambda N - lV \end{aligned} \tag{A.1}$$

The length between any two vertices of the tetrahedron is $2l$.

The evolution length parameter $\lambda$ is:

$$\begin{aligned}
||C-A|| &= 2l \\
||O-\lambda N+lV-O-\lambda N-lU|| &= 2l \\
||-2\lambda N+l(V-U)|| &= 2l \\
\sqrt{(-2\lambda N_x+l(V_x-U_x))^2+(-2\lambda N_y+l(V_y-U_y))^2+(-2\lambda N_z+l(V_z-U_z))^2} &= 2l \\
\sqrt{4\lambda^2 N_x^2-4\lambda l N_x(V_x-U_x)+l^2(V_x-U_x)^2+\dots} &= 2l \\
\sqrt{4\lambda^2 N\cdot N-4\lambda l N\cdot(V-U)+l^2(V-U)\cdot(V-U)} &= 2l \\
\sqrt{4\lambda^2+2l^2} &= 2l \\
4\lambda^2+2l^2 &= 4l^2 \\
\lambda^2 &= \frac{2l^2}{4} \\
\lambda &= \frac{l\sqrt{2}}{2} \\
\lambda &= \frac{l}{\sqrt{2}}
\end{aligned} \tag{A.2}$$

where $N\cdot N=N_x^2+N_y^2+N_z^2$, $N\cdot(V-U)=0$, and $(V-U)\cdot(V-U)=2$.

The radius of the sphere containing the tetrahedron is:

$$\begin{aligned}
r^2 &= \lambda^2+l^2=\frac{l^2}{2}+l^2=\frac{3l^2}{2} \\
r &= l\sqrt{\frac{3}{2}}
\end{aligned} \tag{A.3}$$

The tilting angle value $\theta$ is:

$$\begin{aligned}
\theta &= \frac{\pi}{2}-\arctan\left(\frac{\frac{l}{2}}{\lambda}\right) \\
\theta &= \frac{\pi}{2}-\arctan\left(\frac{\frac{l}{2}}{\frac{l}{\sqrt{2}}}\right) \\
\theta &= \frac{\pi}{2}-\arctan\left(\frac{1}{\sqrt{2}}\right) \\
\theta &\approx 0.9553166 \text{ rads} \\
\theta &\approx 54.73561^\circ
\end{aligned} \tag{A.4}$$

Cube

The cube results from the evolution of a regular hexagon with side length $l$ into a point through two evolution steps using the double direction evolution method.

The edge length of the cube is:

$$\begin{aligned} L^2 &= \frac{4l^2}{2} = 2l \\ L &= l\sqrt{2} \end{aligned} \tag{A.5}$$

The radius of the sphere containing the cube is:

$$\begin{aligned} r^2 &= l^2 + \frac{L^2}{4} = l^2 + \frac{2l^2}{4} = l^2 + \frac{l^2}{2} = \frac{3l^2}{2} \\ r &= l\sqrt{\frac{3}{2}} \end{aligned} \tag{A.6}$$

The side length $t$ of the evolved triangles (first evolution step along both directions) is:

$$\begin{aligned} t^2 &= 2L^2 = 4l^2 \\ t &= 2l \end{aligned} \tag{A.7}$$

The height of the evolved triangles $h$ is:

$$\begin{aligned} h^2 &= t^2 - \frac{t^2}{4} = 4l^2 - \frac{4l^2}{4} = 4l^2 - l^2 = 3l^2 \\ h &= l\sqrt{3} \end{aligned} \tag{A.8}$$

The first evolution length $\lambda_0$ is:

$$\begin{aligned} \lambda_0^2 &= r^2 - \left(\frac{2h}{3}\right)^2 = r^2 - \frac{12l^2}{9} = \frac{3l}{2} - \frac{4l^2}{3} = \frac{l^2}{6} \\ \lambda_0 &= \frac{l}{\sqrt{6}} \end{aligned} \tag{A.9}$$

The second evolution length $\lambda_1$ is:

$$\begin{aligned} \lambda_1 &= r - \lambda_0 = \frac{l\sqrt{3}}{\sqrt{2}} - \frac{l}{\sqrt{6}} \\ \lambda_1 &= \frac{2l}{3} \end{aligned} \tag{A.10}$$

The tilting angle value $\theta$ is:

$$\begin{aligned}
\theta &= \frac{\pi}{2} - \arctan\left(\frac{\frac{h}{2}}{r}\right) \\
\theta &= \frac{\pi}{2} - \arctan\left(\frac{\frac{l\sqrt{3}}{2}}{\frac{l\sqrt{3}}{\sqrt{2}}}\right) \\
\theta &= \frac{\pi}{2} - \arctan\left(\frac{1}{\sqrt{2}}\right) \\
\theta &\approx 0.9553166181 \text{ rads} \\
\theta &\approx 54.73561032^\circ
\end{aligned} \tag{A.11}$$

Octahedron

The octahedron results from the evolution of a regular hexagon with side length $l$ into triangle through one evolution steps using the double direction evolution method.

Let $L$ be the edge length of the regular octahedron. Each face of the octahedron is an equilateral triangle with side length $L$. Then $\triangle ABD \sim \triangle ACE$. The edge length of the octahedron is:

$$\begin{aligned}
\frac{AB}{AC} &= \frac{BD}{CE} \\
\frac{\frac{L}{2}}{\frac{L}{1}} &= \frac{l}{L} \\
\frac{1}{2} &= \frac{l}{L} \\
L &= 2l
\end{aligned} \tag{A.12}$$

Let $r$ be the radius of the octahedron. Then:

$$\begin{aligned}
r^2 &= l^2 + \frac{L^2}{4} \\
r^2 &= l^2 + \frac{4l^2}{4} \\
r^2 &= l^2 + l^2 \\
r^2 &= 2l^2 \\
r &= l\sqrt{2}
\end{aligned} \tag{A.13}$$

The height of a triangle face of the octahedron is $h = l\sqrt{3}$. Then $\frac{2l}{\sqrt{3}}$ is the median of such triangle. Let $\lambda$ the length of the evolution step. Its value is the same for both positive and negative evolution. Then:

$$\begin{aligned} r^2 &= \lambda^2 + \frac{4l^2}{3} \\ \lambda^2 &= r^2 - \frac{4l^2}{3} \\ \lambda^2 &= 2l^2 - \frac{4l^2}{3} \\ \lambda^2 &= \frac{6l^2 - 4l^2}{3} \\ \lambda^2 &= \frac{2l^2}{3} \\ \lambda &= l\sqrt{\frac{2}{3}} \end{aligned} \tag{A.14}$$

The height of a triangle in the hexagon is $h' = \frac{l\sqrt{3}}{2}$. Let $\theta$ be the tilting angle value (without direction) for the evolution step. Its value is the same for both positive and negative evolution. Then $\theta = \frac{\pi}{2} - \theta'$. Therefore:

$$\begin{aligned} \tan(\theta') &= \frac{h' - \frac{h}{3}}{\lambda} \\ \theta' &= \arctan\left(\frac{h' - \frac{h}{3}}{\lambda}\right) \\ \theta' &= \arctan\left(\frac{\frac{l}{2\sqrt{3}}}{\frac{l\sqrt{2}}{\sqrt{3}}}\right) \\ \theta' &= \arctan\left(\frac{1}{2\sqrt{2}}\right) \\ \theta' &\approx 0.3398369095 \text{ rads} \\ \theta' &\approx 19.47122063^\circ \end{aligned} \tag{A.15}$$

Then,

$$\begin{aligned} \theta &= \frac{\pi}{2} - \arctan\left(\frac{1}{2\sqrt{2}}\right) \\ \theta &\approx 1.230959417 \text{ rads} \\ \theta &\approx 70.52877937^\circ \end{aligned} \tag{A.16}$$

Dodecahedron

The dodecahedron results from the evolution of a regular decagon with side length $l$ into a pentagonal face through two evolution steps using the double direction evolution method.

Let $L$ be the edge length of the regular dodecahedron. Each face of the dodecahedron is a regular pentagon with side length $L$. Then $\triangle ABD \sim \triangle ACE$. The edge length of the dodecahedron is:

$$\begin{aligned} \frac{AD}{AE} &= \frac{BD}{CE} \\ \frac{\frac{L}{2}}{\frac{L}{1}} &= \frac{\frac{l}{2}}{\frac{\phi L}{2}} \\ \frac{1}{2} &= \frac{l}{\phi L} \\ L &= \frac{2l}{\phi} \end{aligned} \tag{A.17}$$

where $\phi = \frac{1+\sqrt{5}}{2}$ is the golden ratio.

Let $r'$ the radius of the decagon. In a triangle of the decagon:

$$\begin{aligned} \sin\left(\frac{\frac{\pi}{5}}{\frac{2}{1}}\right) &= \frac{\frac{l}{2}}{\frac{r'}{1}} \\ \sin\left(\frac{\pi}{10}\right) &= \frac{l}{2r'} \\ r' &= \frac{l}{2\sin\left(\frac{\pi}{10}\right)} \end{aligned} \tag{A.18}$$

Having $a = 2\sin\left(\frac{\pi}{10}\right)$ then $r' = \frac{l}{a}$.

Let $a'$ be the apothem of the decagon. Then:

$$\begin{aligned} \tan\left(\frac{\frac{2\pi}{10}}{\frac{2}{1}}\right) &= \frac{\frac{l}{2}}{\frac{a'}{1}} \\ \tan\left(\frac{\pi}{10}\right) &= \frac{l}{2a'} \\ a' &= \frac{l}{2\tan\left(\frac{\pi}{10}\right)} \end{aligned} \tag{A.19}$$

Having $b = 2\tan\left(\frac{\pi}{10}\right)$, then $a' = \frac{l}{b}$.

Let $r''$ be the radius of the pentagon (first evolution polytope). Then:

$$\begin{aligned}\sin\left(\frac{\frac{2\pi}{5}}{\frac{2}{1}}\right) &= \frac{\frac{\phi L}{2}}{\frac{r''}{1}}\\ \sin\left(\frac{\pi}{5}\right) &= \frac{\phi L}{2r''}\\ r'' &= \frac{\phi L}{2\sin\left(\frac{\pi}{5}\right)}\end{aligned}\tag{A.20}$$

Having $c = 2\sin\left(\frac{\pi}{5}\right)$, then:

$$\begin{aligned}r'' &= \frac{\phi L}{c}\\ r'' &= \frac{\frac{2\phi l}{\phi}}{\frac{c}{1}}\\ r'' &= \frac{2l}{c}\end{aligned}\tag{A.21}$$

Let $a''$ be the apothem of the pentagon. Then,

$$\begin{aligned}\tan\left(\frac{\frac{2\pi}{5}}{\frac{2}{1}}\right) &= \frac{\frac{\phi L}{2}}{\frac{a''}{2}}\\ \tan\left(\frac{\pi}{5}\right) &= \frac{\phi L}{2a''}\\ a'' &= \frac{\phi L}{2\tan\left(\frac{\pi}{5}\right)}\end{aligned}\tag{A.22}$$

Having $d = 2\tan\left(\frac{\pi}{5}\right)$, then:

$$\begin{aligned}a'' &= \frac{\phi L}{d}\\ a'' &= \frac{\frac{2\phi l}{\phi}}{\frac{d}{a}}\\ a'' &= \frac{2l}{d}\end{aligned}\tag{A.23}$$

Let $R$ be the radius of the dodecahedron. Then:

$$\begin{aligned}
R^2 &= \frac{L^2}{4} + r'^2 \\
R^2 &= \frac{\frac{4l^2}{\phi^2}}{\frac{4}{1}} + \frac{l^2}{a^2} \\
R^2 &= \frac{l^2}{\phi^2} + \frac{l^2}{a^2} \\
R^2 &= l^2 \left( \frac{a^2 + \phi^2}{a^2 \phi^2} \right) \\
R &= \frac{l}{a\phi} \sqrt{a^2 + \phi^2}
\end{aligned} \tag{A.24}$$

Let $\lambda_0$ be the first evolution length, its value is the same for both positive and negative evolution. Then:

$$\begin{aligned}
R^2 &= \lambda_0^2 + r''^2 \\
\lambda_0^2 &= R^2 - r''^2 \\
\lambda_0^2 &= \frac{l^2 \left(a^2 + \phi^2\right)}{a^2 \phi^2} - \frac{4l^2}{c^2} \\
\lambda_0^2 &= \frac{l^2}{a^2 c^2 \phi^2} \left( c^2 \left( a^2 + \phi^2 \right) - 4a^2 \phi^2 \right) \\
\lambda_0 &= \frac{l}{ac\phi} \sqrt{a^2 c^2 + c^2 \phi^2 - 4a^2 \phi^2}
\end{aligned} \tag{A.25}$$

The radius of the top pentagon (second evolved polygon) is $r''' = \frac{2l}{c\phi}$. Similarly, its apothem is $a''' = \frac{2l}{d\phi}$. Let $\lambda'$ be the height of the dodecahedron. Then:

$$\begin{aligned}
R^2 &= \lambda'^2 + a'''^2 \\
\lambda'^2 &= R^2 - a'''^2 \\
\lambda'^2 &= \frac{l^2 \left(a^2 + \phi^2\right)}{a^2 \phi^2} - \frac{4l^2}{c^2 \phi^2} \\
\lambda'^2 &= \frac{l^2}{\phi^2} \left( \frac{c^2 \left(a^2 + \phi^2\right) - 4a^2}{a^2 c^2} \right) \\
\lambda' &= \frac{l}{ac\phi} \sqrt{a^2 c^2 + c^2 \phi^2 - 4a^2}
\end{aligned} \tag{A.26}$$

Let $\lambda_1$ be the length of the second evolution step. Then:

$$\begin{aligned}
\lambda' &= \lambda_0 + \lambda_1 \\
\lambda_1 &= \lambda' - \lambda_0 \\
\lambda_1 &= \frac{l\sqrt{a^2c^2 + c^2\phi^2 - 4a^2}}{ac\phi} - \frac{l\sqrt{a^2c^2 + c^2\phi^2 - 4a^2\phi^2}}{ac\phi} \\
\lambda_1 &= \frac{l}{ac\phi}\left(\sqrt{a^2c^2 + c^2\phi^2 - 4a^2} - \sqrt{a^2c^2 + c^2\phi^2 - 4a^2\phi^2}\right)
\end{aligned} \tag{A.27}$$

Let $\theta$ be the tilting angle value for both evolution steps. It is without direction for the first step, and with direction for the second step.Then $\theta = \frac{\pi}{2} - \theta'$. Therefore:

$$\begin{aligned}
\tan\theta' &= \frac{a' - a''}{\lambda_0} \\
\theta' &= \arctan\left(\frac{\frac{l(d-2b}{db}}{\frac{l\sqrt{a^2c^2+c^2\phi^2-4a^2\phi^2}}{ac\phi}}\right) \\
\theta' &= \arctan\left(\frac{ac\phi(d-2b)}{db\sqrt{a^2c^2 + c^2\phi^2 - 4a^2\phi^2}}\right) \\
\theta' &\approx 0.463647609 \text{ rads} \\
\theta' &\approx 26.56505118^\circ
\end{aligned} \tag{A.28}$$

Then,

$$\begin{aligned}
\theta &= \frac{\pi}{2} - \arctan\left(\frac{ac\phi(d-2b)}{db\sqrt{a^2c^2 + c^2\phi^2 - 4a^2\phi^2}}\right) \\
\theta &= 1.107148718 \text{ rads} \\
\theta &= 63.43494882^\circ
\end{aligned} \tag{A.29}$$

Icosahedron

The icosahedron results from the evolution of a regular decagon with side length $l$ into a point through two evolution steps using the double direction evolution method.

Let $L$ be the edge length of the regular icosahedron. Each face of the icosahedron is an equilateral triangle with side length $L$. Then $\triangle ABD \sim \triangle ACE$. The edge length of the icosahedron is:

$$\begin{aligned} \frac{AB}{AC} &= \frac{BD}{CE} \\ \frac{\frac{L}{2}}{\frac{L}{1}} &= \frac{\frac{l}{1}}{\frac{L}{1}} \\ \frac{1}{2} &= \frac{l}{L} \\ L &= 2l \end{aligned} \tag{A.30}$$

Let $r'$ be the radius of the regular decagon. In a triangle of the decagon:

$$\begin{aligned} \sin\left(\frac{\frac{2\pi}{10}}{\frac{2}{1}}\right) &= \frac{\frac{l}{2}}{\frac{r'}{1}} \\ \sin\left(\frac{\pi}{10}\right) &= \frac{l}{2r'} \\ r' &= \frac{l}{2\sin\left(\frac{\pi}{10}\right)} \end{aligned} \tag{A.31}$$

Having $a = \sin\left(\frac{\pi}{10}\right)$ then $r' = \frac{l}{2a}$.

Let $a'$ be the apothem of the decagon. Then:

$$\begin{aligned} \tan\left(\frac{\frac{2\pi}{10}}{\frac{2}{1}}\right) &= \frac{\frac{l}{2}}{\frac{a'}{1}} \\ \tan\left(\frac{\pi}{10}\right) &= \frac{l}{2a'} \\ a' &= \frac{l}{2\tan\left(\frac{\pi}{10}\right)} \end{aligned} \tag{A.32}$$

Having $b = \tan\left(\frac{\pi}{10}\right)$, then $a' = \frac{l}{2b}$.

Let $r''$ be the radius of the pentagon (first evolution polytope). Then:

$$\begin{aligned} \sin\left(\frac{\frac{2\pi}{5}}{\frac{2}{1}}\right) &= \frac{\frac{L}{2}}{\frac{r''}{1}} \\ \sin\left(\frac{\pi}{5}\right) &= \frac{2l}{2r''} \\ r'' &= \frac{l}{\sin\left(\frac{\pi}{5}\right)} \end{aligned} \tag{A.33}$$

Having $c = \sin\left(\frac{\pi}{5}\right)$, then $r'' = \frac{l}{c}$;

Let $a''$ be the apothem of the pentagon. Then,

$$\begin{aligned} \tan\left(\frac{\frac{2\pi}{5}}{\frac{2}{1}}\right) &= \frac{\frac{L}{2}}{\frac{a''}{1}} \\ \tan\left(\frac{\pi}{5}\right) &= \frac{2l}{2a''} \\ a'' &= \frac{l}{\tan\left(\frac{\pi}{5}\right)} \end{aligned} \tag{A.34}$$

Having $d = \tan\left(\frac{\pi}{5}\right)$, then $a'' = \frac{l}{d}$.

Let $R$ be the radius of the icosahedron. Then:

$$\begin{aligned} R^2 &= r'^2 + \frac{L^2}{4} \\ R^2 &= \frac{l^2}{4a^2} + \frac{4l^2}{4} \\ R^2 &= l^2\left(\frac{1}{4a^2} + 1\right) \\ R^2 &= l^2\left(\frac{1+4a^2}{4a^2}\right) \\ R &= \frac{l}{2a}\sqrt{1+4a^2} \end{aligned} \tag{A.35}$$

Let $\lambda_0$ be the first evolution length, its value is the same for both positive and negative evolution. Then:

$$\begin{aligned} R^2 &= \lambda_0^2 + r''^2 \\ \lambda_0^2 &= R^2 - r''^2 \\ \lambda_0^2 &= \frac{l^2\left(1+4a^2\right)}{4a^2} - \frac{l^2}{c^2} \\ \lambda_0^2 &= l^2\left(\frac{1+4a^2}{4a^2} - \frac{1}{c^2}\right) \\ \lambda_0 &= l\sqrt{\frac{1+4a^2}{4a^2} - \frac{1}{c^2}} \end{aligned} \tag{A.36}$$

Let $\lambda_1$ be the second evolution length, its value is the same for both positive and negative evolution. Then:

$$\begin{aligned}
L^2 &= \lambda_1^2 + r''^2 \\
\lambda_1^2 &= L^2 - r''^2 \\
\lambda_1^2 &= 4l^2 - \frac{l^2}{c^2} \\
\lambda_1^2 &= \frac{l^2}{c^2}\left(4c^2 - 1\right) \\
\lambda_1 &= \frac{l}{c}\sqrt{4c^2 - 1}
\end{aligned} \tag{A.37}$$

Let $\theta_0$ be the tilting angle value (without direction) for the first evolution step. Its value is the same for both positive and negative evolution using the respective edge directions. Then $\theta_0 = \frac{\pi}{2} - \theta'$. Therefore:

$$\begin{aligned}
\theta' &= \arctan\left(\frac{a' - a''}{\lambda_0}\right) \\
\theta' &= \arctan\left(\frac{\frac{l(d-2b}{2bd}}{l\sqrt{\frac{1+4a^2}{4a^2} - \frac{1}{c^2}}}\right) \\
\theta' &= \arctan\left(\frac{d - 2b}{2bd\sqrt{\frac{1+4a^2}{4a^2} - \frac{1}{c^2}}}\right) \\
\theta' &\approx 0.1887105308 \text{ rads} \\
\theta' &\approx 10.81231696^\circ
\end{aligned} \tag{A.38}$$

Then,

$$\begin{aligned}
\theta_0 &= \frac{\pi}{2} - \arctan\left(\frac{d - 2b}{2bd\sqrt{\frac{1+4a^2}{4a^2} - \frac{1}{c^2}}}\right) \\
\theta_0 &= 1.382085796 \text{ rads} \\
\theta_0 &= 79.18768306^\circ
\end{aligned} \tag{A.39}$$

The height of a triangular face of the icosahedron is $h = l\sqrt{3}$. Let $\theta_1$ be the tilting angle value (with direction) for the second evolution step. Its value is the same for both positive and negative evolution. Then:

$$
\begin{aligned}
\theta_1 &= \arccos\left(\frac{a''}{h}\right) \\
\theta_1 &= \arccos\left(\frac{\frac{l}{d}}{\frac{l\sqrt{3}}{1}}\right) \\
\theta_1 &= \arccos\left(\frac{1}{d\sqrt{3}}\right) \\
\theta_1 &\approx 0.6523581398 \text{ rads} \\
\theta_1 &\approx 37.37736814^\circ
\end{aligned}
\tag{A.40}
$$

# VITA

Andres Mauricio Bejarano Posada is a Ph.D. student in the Computer Graphics and Visualization Lab (CGVLAB) at Purdue University, IN. He received a M.Sc. degree in Computer Science from Purdue University, IN., and a M.Sc. and B.Sc. in Systems Engineering and Computation from Universidad del Norte, Colombia. Prior to joining Purdue, he was an adjunct professor in the Systems Engineering and Computation department at Universidad del Norte, Colombia.

His research interest is in Computer Graphics, Computational Geometry, Polygon Geometry Processing, and Scientific Visualization. Specifically, he is interested in the analysis and design of geometric domains to reach functional attributes.