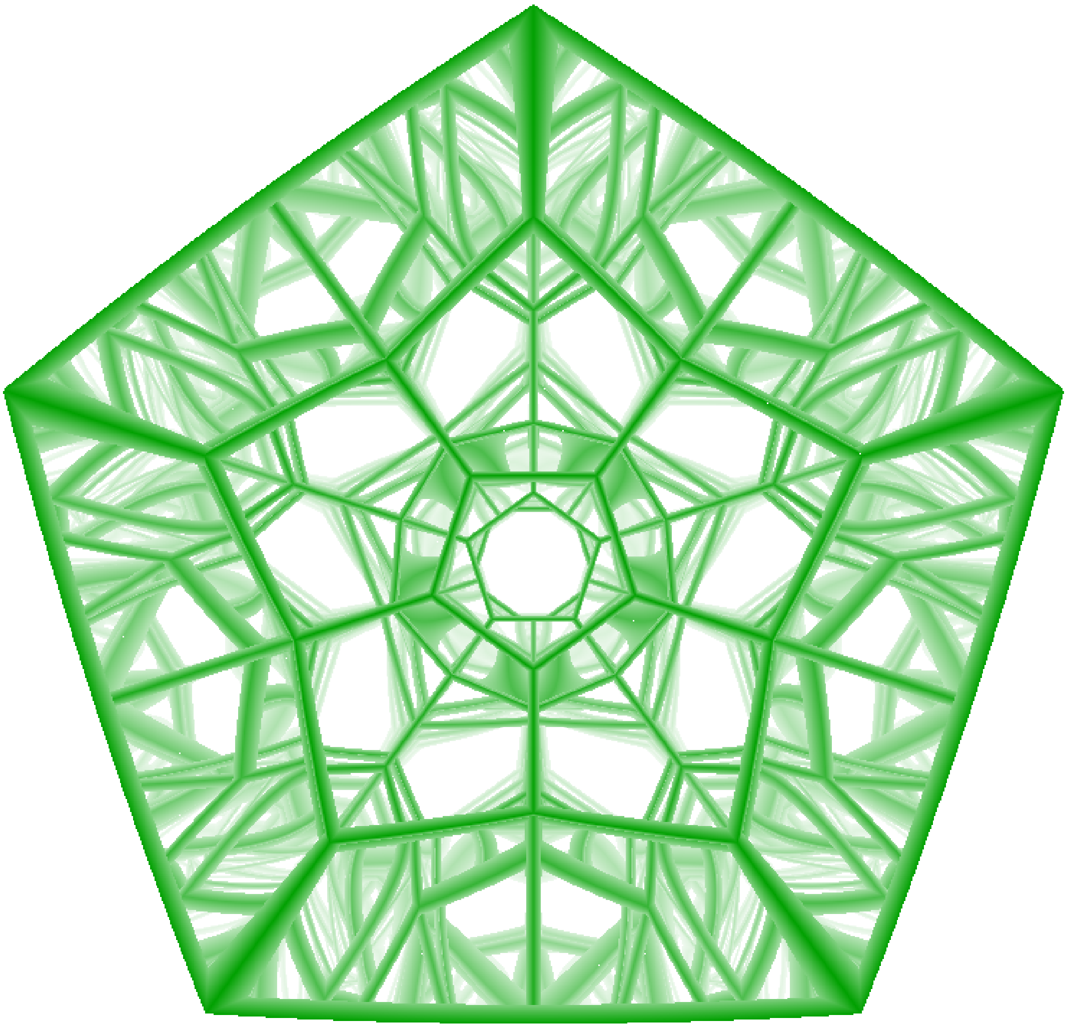
On The Patterns From Curved Infinity-Mirror Boxes

Timothy van der Valk

April 2024



Cover: A dodecahedron infinity box with curved faces.

Abstract

We have defined the infinity boxes in general dimensions, which are mathematical objects that allow rays of light to continuously reflect. When the collision points of these rays are are positioned where they would appear to the viewer (along the inititial ray), then the mirror pattern will be revealed. These mirror patterns can be used to create mirror-tilings of two- and three-dimensional Euclidian space. We have answered the question whether a curved infinity box can create patterns that are nearly identical to exact tilings of Euclidian space. To obtain the answer, we set up the fundamental theory of rays and infinity boxes, and computed the radii of the curved mirror faces that exhibit certain properties. These properties are the alignmnent of internal angles, and the alignment of image points in an infinity box. We have simulated infinity boxes in two- and three-dimensions to visualise the patterns from these infinity boxes curved with these radii. Furthermore, we connected the theory of infinity boxes the theory of dynamical billiards by stating that infinity boxes visualise solutions of dynamical billiards. Lastly, we analysed the connection between the dodecahedron infinity box, and concluded that curving the mirrors of the dodecahedron infinity box results in a slight approximation of the interior of a stereographically projected 120-Cell. All the numerical methods used to visualise infinity boxes have been thoroughly explained, and the computer program that we created called MiRai, is available over the internet.

Contents

1 Introduction

An Infinity Box, also known as an Infinity Mirror Box or Mirror Box, is a box made of mirrors facing inward. When the mirrors are made transparent from the outside, a light can be cast inside. This light will then keep reflecting inside the box until it dissipates or hits a non-reflective surface. Infinity Boxes are commonly built for their visual aesthethic, but beneath their dazzling patterns also lies an intricate mathematical pattern.

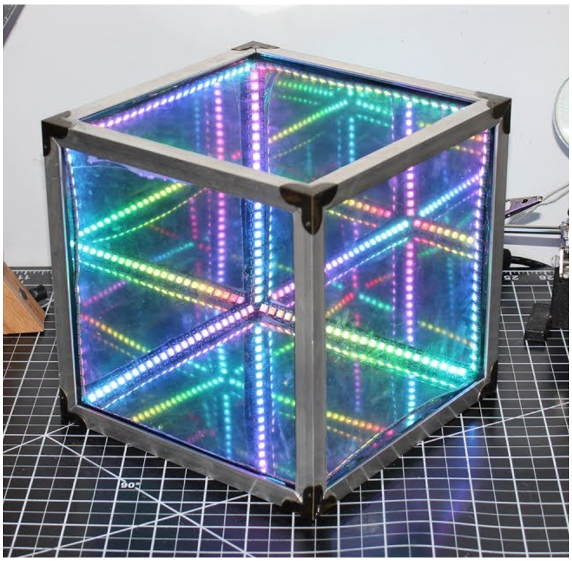


Figure 1.1: A cube infinity box with LED lighting to highlight edges [3].

This pattern that is created by 3D infinity boxes is tightly correlated to tilings of 3D space. However, not all tilings are created from infinity boxes. It turns out that tilings created from infinity boxes, what we call mirror-tilings, are a rare occurence in Euclidian space. A mirror-tiling is a tiling of space where neighboring tiles are mirrored in their common surface, which are edges in 2D and faces in 3D. The cube, as seen in 1.1, is one of the few shapes in 3D Euclidian space that is mirror-tiling. As a result, we can see that the interior of its infinity box is effectively generating this tiling. We are interested in researching the ways in which infinity boxes can generate such mirror-tilings. Questions that we can ask ourselves are

1. Which 3D polyhedra or 2D polygons are mirror-tiling?

2. If an infinity box pattern is not a mirror-tiling, can we find any structure or repetition in this pattern?
3. Can curved infinity boxes give an illusion of a mirror-tiling in their pattern?

It turns out that point (1) is already know in 2D Euclidian space, as shown in a paper by Glenn Harris [6]. For 3D space, a few mirror-tiling polyhedra are given in Figure 1.2. We will concern ourselves with point (2) and (3), in particular. Part of our research is about defining infinity boxes and measuring how well they can tile, the second part of our research continues on point (3). We will analyse the behavior of infinity boxes when the mirrors become slightly curved. The reason for turning to curved mirrors can be explained with pentagon, dodecahedron and 120-Cell.

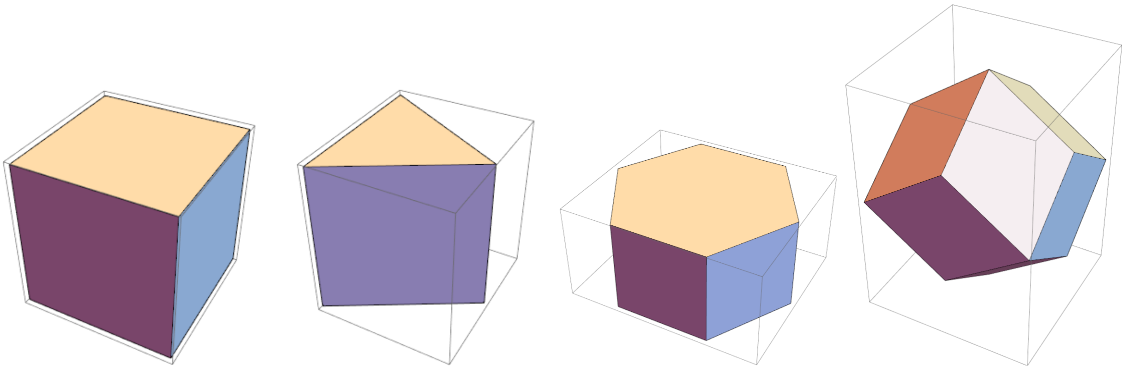


Figure 1.2: The four shapes with connected infinity boxes in Euclidian space (up to scaling). From left to right: Cube, Triangular Prism, Hexagonal Prism and Rhombic Dodecahedron. (Mathematica).

The pentagon, dodecahedron and 120-Cell are 2D, 3D and 4D objects respectively. Starting with the 120 Cell, this is a four-dimensional object that consists of 120 dodecahedra that seamlessly connected to each other. Moving down one dimension, the dodecahedron has 12 regular pentagons as faces. Another dimension lower and we end up with a regular pentagon. We can not tile two-dimensional Euclidian space with pentagons, neither can we tile three-dimensional Euclidian space with dodecahedra. However, we note that the for all three objects, the corresponding vertices lie on a sphere. The vertices of the pentagon lie on the 1-sphere (circle) S^1, similarly, the vertices of the dodecahedron lie on the 2-sphere S^2 and the vertices of the 120-Cell lie on S^3. One can use this fact to tile the 2-sphere with curved pentagons by inflating the dodecahedron onto the 2-sphere. By projecting this tiling of the sphere (spherical space) down to two-dimensional Euclidian space using either a perspective projection or a stereographic projection, we will end up with a deformed tiling of two-dimensional space using pentagons. See Figure 1.3.

Since the edges in Figure 1.3 are curved, we have reasons to believe that we can achieve a similar tiling like 1.3 by considering a pentagonal infinity box with curved mirrors instead. In 2D spherical space, straight lines are replaced by circles. Any point moving in one direction will eventually be back at its same position. What would happen to a ray of light that is cast inside spherical space? The ray would follow the trajectory of a circle. If we we're to make an infinity box in spherical space, then we could generate a tiling of curved pentagons in spherical space. However, we wish to analyse the tilings of Euclidian infinity boxes. So instead of curving the rays, which we can not do in Euclidian space, we curve the mirrors.

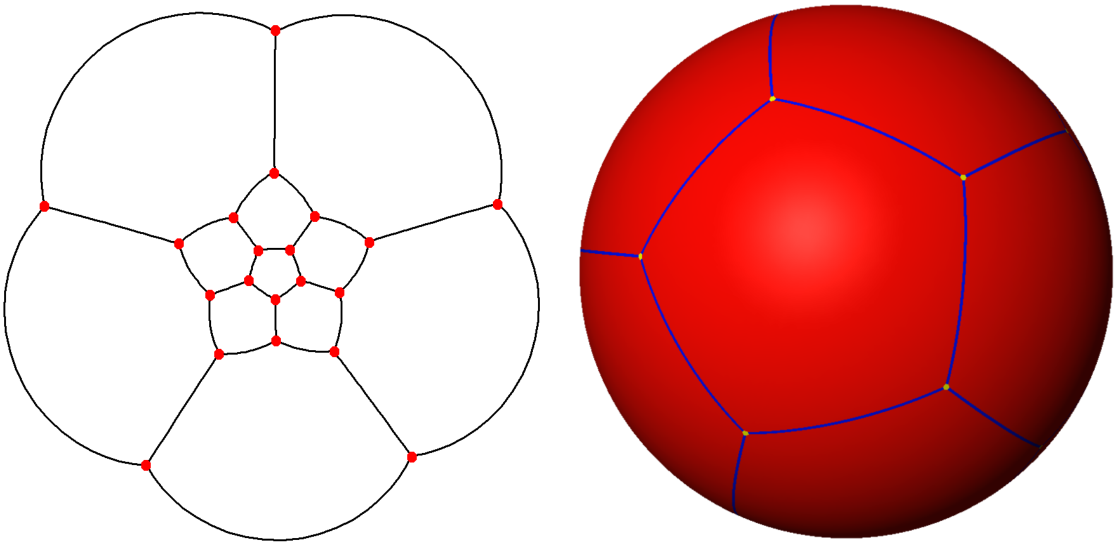


Figure 1.3: A dodecahedron that is stereographically projected face-centered to two-dimensional Euclidian space. This result in a tiling of two-dimensional space with deformed pentagons [14].

The example that we have given here for the pentagon and dodecahedron can be repeated in the same fashion for the dodecahedron and the 120-Cell. The dodecahedron now assumes the role of the pentagon. Again, we can project the 120 dodecahedra from the 120-Cell down to 3D space, and we would end up with a tiling of 3D space with deformed dodecahedra. This is shown in Figure 1.4. We are not restricted to the dodecahedron and 120-Cell either. Other 4D polytopes such as the 8-Cell, 16-Cell, 24-Cell or 600-Cell can also be projected to 3D to create (deformed) tilings.

The primary purpose of this paper will be to define infinity boxes mathematically and subsequently simulate them on a computer. In addition, we relate infinity boxes to earlier literature and mathematics and explain the potential use of infinity boxes to solve problems related to other fields of mathematics. The paper is broadly divided in three sections:

1. A brief overview of related results from other fields of mathematics.
2. The formal definition of infinity boxes and analytical results related curved infinity boxes.
3. The numerical simulation of infinity boxes and explanation of the used rendering techniques.

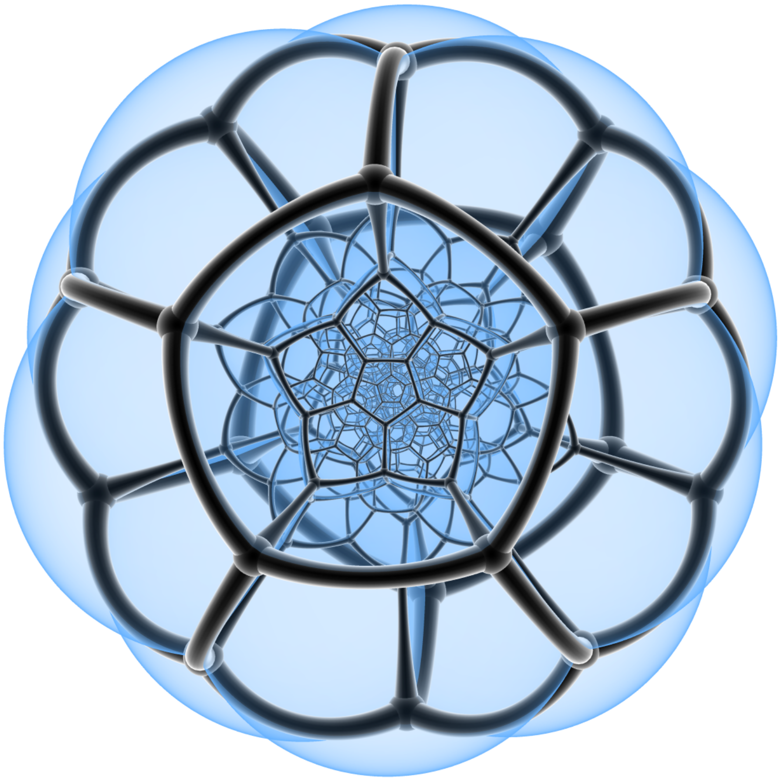


Figure 1.4: Projection from S^3 to $\mathbb{R}^3$ of the 120-cell. The tiling of dodecahedra created by this projection is what motivates us to use dodecahedron infinity boxes with curved mirrors [11].

2 Literature Related To Infinity Boxes

Literature on infinity boxes in particular is scarce, but infinity boxes can be seen as a branch of physical optics problems and dynamic billiard systems. Much research has already been performed in these fields. In this chapter we briefly mention topics that have a strong relation to infinity boxes.

2.1 Dynamical Billiards

Dynamical billiards are dynamical systems in which a particle with no friction is moving and reflecting against walls in all directions, which we will call a table. These systems derive their name for the game of billiards, where a ball must be struck on a rectangular pool table. The trajectory that the ball on this table follows, is an intricate pattern that depends on the shape of the table. For rectangular billiards, the trajectory remains reasonable predictable using that laws of reflection. However, the field of dynamical billiards concerns itself with particles reflecting on tables of varying shapes.

Questions that the theory of dynamical billiards concerns itself with are:

1. Does every polygonal billiard admit a cyclic trajectory?
2. Can every other position be reached when releasing the particle from a specific position?

Question (1) has been frequently approached in literature. A polygonal billiard is a billiard system set on a two-dimensional table shaped like a polygon. For acute triangles, this problem is called *Fagnano's Problem* and has been solved Giovanni Fagnano himself. See the book by Heinrich Dörrie [4] for a proof.

Theorem 1 (Giovanni Fagnano, Fagnano Trajectory). *Given an acute triangle. Then there exists a periodic trajectory inside the triangle.*

The problem for cyclic trajectory has also been studied for other arbitrarily shaped tables. Howard Masur solved the problem in 1986 for rational polygon tables [8]. A rational polygon is a polygon whose edges are all rational numbers.

Theorem 2 (Howard Masur, 1986, Rational Billiard Trajectories). *For any rational billiard table there is a dense set of directions each with a periodic trajectory.*

How can we relate dynamical billiards to infinity boxes? Both systems follow the same principles of a ray reflecting inside a closed surface. During our numerical simulations in Chapter 5, we will encounter patterns called "tunnels" in infinity boxes that repeat to infinity. These tunnels correspond to rays of light that never collide with a vertex (and therefore do not come to a halt). These visual tunnels will be periodic trajectories. This means that for infinity boxes constructed from acute triangles or rational polygons, we will always be able to find a tunnel that repeats to infinity by applying the above theorems.

While the theorems mentioned in this section only apply to two-dimensional billiards we have reasons to believe that similar results can be formulated for higher-dimensional billiards. This is because we will encounter tunnels in three-dimensional infinity boxes as well. To conclude, we can pose that infinity boxes are means to visualise the trajectories of dynamical billiards. This means that infinity box can be applied to gain insight in problems related to dynamical billiards.

Problem (2) brings us to the illumination problem.

2.2 The Illumination Problem

The illumination problem is related to dynamical billiards in that they are both problems set on a "table" with a particle reflecting on the interior. It is Ernst Strauss who first proposed the problem in 1950. The illumination problem is formulated as follows:

Given a billiard system. Is there a pair of points on the table through which no trajectory passes?

This is equivalent to asking if there is any spot on the table that can not be illuminated by a light source at another spot. This problem, like the problem of periodic trajectories, has been studied in the past. In 1958, Roger Penrose used ellipses to create an unilluminable room. This room is shown in Figure 2.1.

The same problem has been solved for polygonal rooms by George Tokarsky in 1995 [12], who also provided an unilluminable room but now without curved mirrors. This solution is shown in Figure 2.2.

How can we relate the illumination problem to infinity boxes? For infinity boxes, we cast rays from a single point in two -or three-dimensional space. We turn infinity boxes to illumination problems by placing a target in the infinity box, and asking ourselves whether there a point in the infinity box from which we can not see this target in the reflections. In this way, we will have a solved the illumination problem with infinity boxes if the target point (pink dot) does not show up in any reflection of the infinity box.

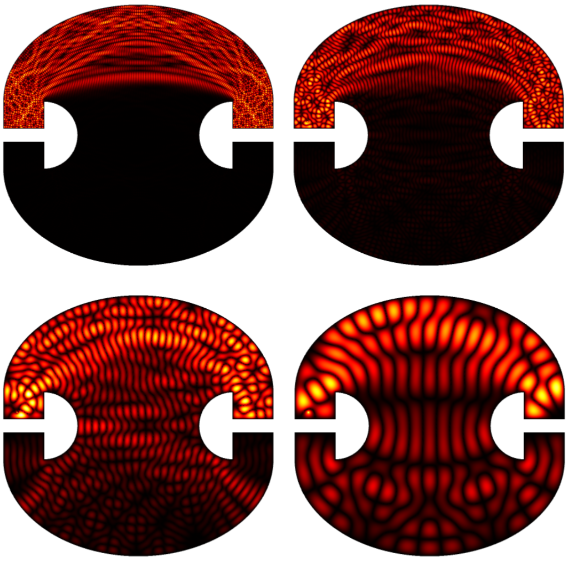


Figure 2.1: The Penrose Unilluminable Room is a shape that where there exists a point from which a light source can not light another part of the room through reflections. The top two images show the illumination when a light source is positioned in one of the pockets near the sides. The bottom two images show the illumination when a light is positioned outside of these pockets. Xinzhong (Tom) Chen [1].

2.3 Mirror-Tiling Polygons

A final note on earlier literature is about the mirror-tiling polygons of two dimensions. In a paper by Glenn Haris [6] has been shown which polygons tile the plane with reflections. We cite the theorem here.

Theorem 3 (Glenn Harris, Mirror-Tiling Polygons)**.** *The class of polygons that tile the plane with reflections is*

- *The regular hexagon (angles* $\frac{2\pi}{6}$*).*
- *The regular rectangle (angles* $\frac{2\pi}{4}$*).*
- *The regular triangle (angles* $\frac{2\pi}{3}$*).*
- *The rhombus with angles* $\frac{2\pi}{3}, \frac{\pi}{6}, \frac{2\pi}{3}, \frac{\pi}{6}$.
- *The kite with angles* $\frac{2\pi}{3}, \frac{\pi}{2}, \frac{\pi}{2}, \frac{\pi}{3}$.
- *The triangle with angles* $\frac{2\pi}{3}, \frac{\pi}{6}, \frac{\pi}{6}$.
- *The triangle with angles* $\frac{\pi}{2}, \frac{\pi}{3}, \frac{\pi}{6}$.
- *The triangle with angles* $\frac{\pi}{2}, \frac{\pi}{4}, \frac{\pi}{4}$.

See Section 5.2.5 for the infinity boxes made with several of these special polygons.

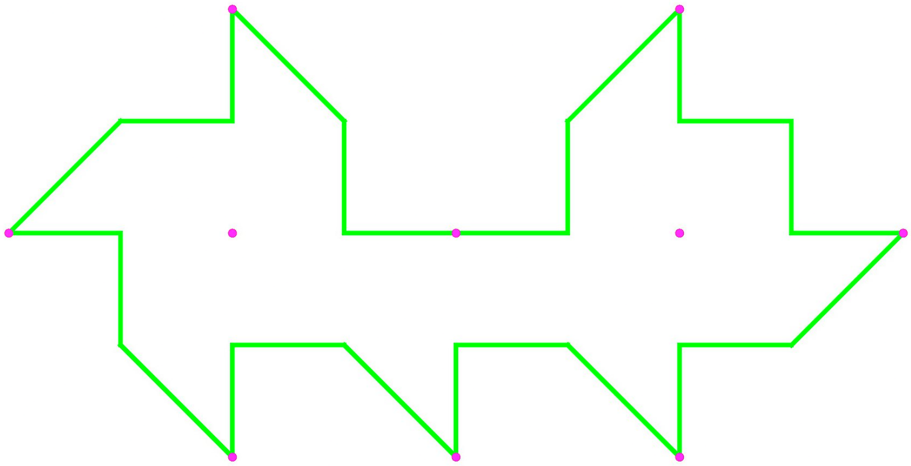


Figure 2.2: The Tokarsky Unilluminable Room is a polygonal shape where there exist pairs of two points that do not illuminate each other [5]. Every pair of points marked in pink can not illuminate each other.

We have now briefly discussed related literature for infinity boxes. We conclude that the topic of dynamical billiards and illumination is very much related to infinity boxes. We will now begin by formally introducing infinity boxes, rays and means to curve infinity boxes.

3 Formal Infinity Boxes

3.1 Rays

In order to analyse the patterns created by infinity boxes, it is necessary to efficiently reflect and collide with mirror surfaces using direct mathematical formulas. This will allow our simulator MiRai to run without delay (real-time) when changing parameters. This chapter provides the foundation for that raytracing theory.

We begin with the definition and notation of a ray and a mirror, which are simply lines and planes respectively.

Definition 3.1.1 (Ray)**.** *A **ray** $(\vec{o}, \vec{d}) \in \mathbb{R}^n \times S^{n-1}$ in Euclidian space is given by the equation*

$$\vec{x}(t) = \vec{o} + t\vec{d}, \quad t \geqslant 0. \tag{3.1}$$

Given a ray, we need to be able to compute its new direction as it gets reflected from a mirror surface. We will define mirrors and vector reflections.

Definition 3.1.2 (Mirror)**.** *A **mirror** $(\vec{c}, \vec{n}) \in \mathbb{R}^n \times S^{n-1}$ is given by all points $\vec{x} \in \mathbb{R}^n$ that satisfy*

$$\langle \vec{x}, \vec{n} \rangle = \langle \vec{c}, \vec{n} \rangle. \tag{3.2}$$

We will now define the mirroring of a vector in a normal.

Definition 3.1.3 (Vector Mirroring)**.** *Given a normal $\vec{n} \in S^{n-1}$. The **vector mirroring** $\vec{x}_*$ of $\vec{x} \in \mathbb{R}^n$ is given by*

$$\vec{x}_* = \vec{x} - 2\vec{n}\langle \vec{x}, \vec{n} \rangle. \tag{3.3}$$

This definition of mirroring only provides the mirrored vector relative to the origin. For mirrors with a position we define the affine mirroring.

Definition 3.1.4 (Affine Mirroring)**.** *Given a mirror $(\vec{c}, \vec{n}) \in \mathbb{R}^n \times S^{n-1}$. The **affine mirroring** $\vec{x}_{*A}$ of the point $\vec{x} \in \mathbb{R}^n$ in the mirror is given by*

$$\vec{x}_{*A} = \vec{c} + (\vec{x} - \vec{c})_* \tag{3.4}$$

Here, a subscript of $*A$ indicates affine mirroring in the mirror $(\vec{c}, \vec{n})$, and a subscript of $*$ indicates vector mirroring in the normal $\vec{n}$.

Be careful to not confuse affine mirroring with vector mirroring. We will use affine mirroring with subscript $*A$ when mirroring a point in Euclidian space in a mirror, and vector mirroring with subscript $*$ when mirroring a direction vector in a normal. Before we move to the Unfolding Theorem, we first show that a mirroring operation (both affine and regular)

Lemma 3.1.1 (Mirror Inverse)**.** *Given a mirror $(\vec{c}, \vec{n}) \in \mathbb{R}^n \times S^{n-1}$. Then for all points $\vec{x} \in \mathbb{R}^n$, $\vec{x}_{**} = \vec{x} = \vec{x}_{*A*A}$.*

Proof. We first prove the result for vector mirroring.

$$\begin{aligned}\vec{x}_{**} &= \vec{x}_{*} - 2\vec{n}\langle \vec{x}_{*}, \vec{n}\rangle \\ &= \vec{x} - 2\vec{n}\langle \vec{x}, \vec{n}\rangle - 2\vec{n}\langle \vec{x} - 2\vec{n}\langle \vec{x}, \vec{n}\rangle, \vec{n}\rangle \\ &= \vec{x} - 4\vec{n}\langle \vec{x}, \vec{n}\rangle + 4\vec{n}\langle \vec{x}, \vec{n}\rangle \\ &= \vec{x}.\end{aligned}$$

Now, for affine mirroring we have

$$\begin{aligned}\vec{x}_{*A*A} &= \vec{c} + (\vec{x}_{*A} - \vec{c})_{*} \\ &= \vec{c} + (\vec{c} + (\vec{x} - \vec{c})_{*} - \vec{c})_{*} \\ &= \vec{c} + (\vec{x} - \vec{c})_{**} \\ &= \vec{x}.\end{aligned}$$

This completes the proof.

□

Note that a reflection of an **incoming** vector (pointing towards the surface) off the surface is equivalent to mirroring that vector in the surface. So Definition 3.1.3 can be used for reflection and mirror. This is also clarified in Figure 3.1.

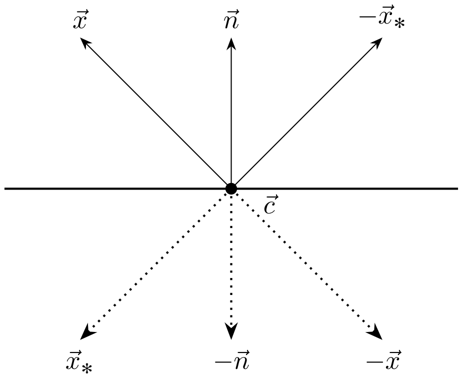


Figure 3.1: Mirroring and reflections of a vector $\vec{x}$ around a normal $\vec{n}$. When a vector points in the same direction as the normal, the result is a mirroring. If the vector points away from the normal (like $-\vec{x}$), then the result is a reflection.

3.2 The Unfolding Theorem

An important result, that is inuitively obvious, is the fact that undoing successive mirrorings on collision points for rays, results in a point that lies on the initial ray $\vec{o}_0 + t\vec{d}_0$. This is similar to "unfolding" a piece of paper. See Figure 3.2. This result justifies the numerical simulations that we use to visualise 2D infinity boxes. We can find the position where a mirror point would appear to a viewer by simply taking the total travelled distance t of the ray until that point and computing $\vec{o} + t\vec{d}$.

Theorem 4 (Unfolding Theorem)**.** *Given a sequence of rays $(\vec{o_i}, \vec{d_i})$ for $i = 0, \dots, N$ where ray $i+1$ is a reflection of ray i in the mirror $(\vec{c_i}, \vec{n_i})$ given by the relation $(\vec{o}_{i+1}, \vec{d}_{i+1}) = (\vec{o_i} + \Phi_i \vec{d_i}, (\vec{d_i})_{*_i}$. Then for all $\lambda \in \mathbb{R}$*

$$(\vec{o}_N + \lambda \vec{d}_N)_{*A_{N-1} \cdots *A_1} = \vec{o}_0 + \left(\lambda + \sum_{j=0}^{N-1} \Phi_j\right) \vec{d}_0 \tag{3.5}$$

Here, $\vec{x}_{*A_{N-1} \cdots *A_1}$ indicates applying successive affine mirroring operations starting with mirror $N-1$.

Proof. Using induction on N, starting from $N = 1$. We perform the affine mirroring of $\vec{o}_1$ in the first mirror $(\vec{o}_1, \vec{d}_0, \vec{n}_0)$. Note that the mirror center is given by $\vec{o}_1 = \vec{o}_0 + \Phi_0 \vec{d}_0$. We get

$$\begin{aligned}
(\vec{o}_1 + \lambda \vec{d}_1)_{*A_0} &= \vec{o}_1 + (\vec{o}_1 + \lambda \vec{d}_1 - \vec{o}_1)_{*_0} \\
&= \vec{o}_1 + \lambda (\vec{d}_1)_{*_0} \\
&= \vec{o}_0 + \Phi_0 \vec{d}_0 + \lambda \vec{d}_0 \\
&= \vec{o}_0 + (\lambda + \Phi_0) \vec{d}_0
\end{aligned}$$

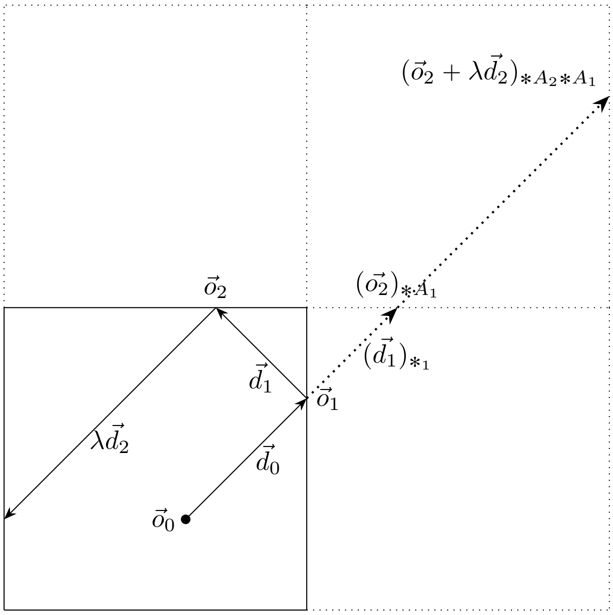


Figure 3.2: The Unfolding Theorem identity asserts that all points $(\vec{o}_1, \vec{o}_2 + \lambda\vec{d}_2)$ along the trajectory of a ray lie on the ray's initial trajectory $\vec{o} + t\vec{d}$. This result is used to visualise image points for infinity boxes in 2D.

Now we assume that the induction hypothesis holds up until N. We consider the collision point $\vec{o}_{N+1}$ and mirror it in all mirrors up until that point to obtain

$$\begin{aligned}
(\vec{o}_{N+1} + \lambda\vec{d}_{N+1})_{*A_N\cdots *A_1} &= ((\vec{o}_{N+1} + (\vec{o}_{N+1} + \lambda\vec{d}_{N+1} - \vec{o}_{N+1}))_{*A_{N-1}\cdots *A_1} \\
&= ((\vec{o}_{N+1} + \lambda(\vec{d}_{N+1})_{*_N}))_{*A_{N-1}\cdots *A_1} \\
&= (\vec{o}_N + \Phi_N\vec{d}_N + \lambda\vec{d}_N)_{*A_{N-1}\cdots *A_1} \\
&= (\vec{o}_N + (\lambda + \Phi_N)\vec{d}_N)_{*A_{N-1}\cdots *A_1} \\
&\overset{I.H}{=} \vec{o}_0 + \left(\lambda + \sum_{j=0}^{N} \Phi j\right)\vec{d}_0
\end{aligned}$$

This completes the proof. □

3.3 The Infinity Box

In order to work with infinity boxes in $\mathbb{R}^2$ and $\mathbb{R}^3$ we need a consistent definition for infinity boxes. This definition is given below.

Definition 3.3.1 (Infinity Box). *An* ***infinity box*** *in $\mathbb{R}^n$ is a set $F \subset \mathbb{R}^n$ that satisfies*

- $F \cong S^{n-1}$ *(F is a topological sphere).*
- $\vec{0} \in$ *Interior*(F).
- *The normal $\vec{n}(\vec{x}) : F \to S^{n-1}$ is a piecewise-continuous function in $\vec{x}$.*

This definition ensures that the infinity box is a closed shape from which a ray can not escape. We require only piece-wise continuity such that infinity boxes with flat faces also fall under this definition.

We provide the general definition to lay a foundation for more specialised definitions of infinity boxes. Throughout this Chapter, we will mostly be working with flat-faced infinity boxes with vertices on the unit sphere. We begin with defining the notion of a "flat" infinity box.

Definition 3.3.2 (Flat Infinity Box). *An infinity box $F \subset \mathbb{R}^n$ is called* ***flat*** *if the normal function $\vec{n}(\vec{x}) : F \to S^{n-1}$ is piecewise-constant in $\vec{x}$ on polytopes P_i in $n-1$ dimensions.*

Every such polytope is called a ***face*** *P_i for $i = 0, \ldots, N$ with $N \in \mathbb{N}$.*

We want each set P_i to be an affine space of dimension $n-1$ such that there exists a single mirror normal that we can use for mirror reflections. See Figure 3.3 for an example of a flat and an arbitrarily shaped infinity box in $\mathbb{R}^2$.

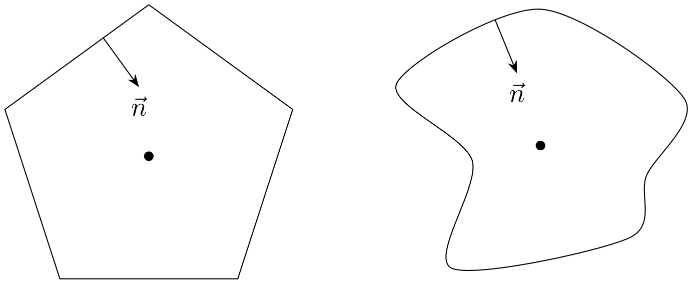


Figure 3.3: A flat pentagon (Left) and smooth curve (Right) infinity box in $\mathbb{R}^2$.

Our next definition further specializes the definition of a flat infinity box to those which have vertices on the unit sphere. These are what we called "spherical flat" infinity boxes.

Definition 3.3.3 (Spherical Flat Infinity Box). *A flat infinity box is called* ***spherical flat*** *if all vertices on the faces $P_i, i = 0, \ldots, N$ lie on the unit sphere.*

Note that a consequence of having all polytope faces P_i on the unit sphere is that the line $\lambda\vec{n}_i$ with $\vec{n}_i \in S^{n-1}$ the normal of the polytope, is equidistant to all vertices of P_i. A sphere centered on a line along the normal passes through all vertices.

In $\mathbb{R}^2$, the definition of the spherical flat infinity box can be simplified. Spherical flat infinity boxes are called "circular flat" in $\mathbb{R}^2$. The vertices lie on the unit circle.

Definition 3.3.4 (Circular Flat Infinity Box)**.** *An infinity box $F \subset \mathbb{R}^2$ is called **circular flat** if for all vertices $\vec{v_i}$ with $i = 1..n$ in the union of P_j we have*

$$\vec{v}_i = \begin{pmatrix} \cos(\alpha_i) \\ \sin(\alpha_i) \end{pmatrix}, \quad i = 1, \ldots, n. \tag{3.6}$$

With $\alpha_i \in [0, 2\pi)$. In addition, the infinity box is said to be N-Regular if $\alpha_i = \frac{2\pi}{N}$ for all $i = 1, \ldots, n$.

We will now turn our attention to our primary object of investigation: curved infinity boxes. Since infinity boxes can be curved in a myriad of different ways, we will consider only curving using spheres and circles. This brings us to the next section.

3.4 Spherical Infinity Boxes

We will define a continuous transformation from a spherical flat infinity box to a curved infinity box by replacing each face P_i with a section of a sphere. The center of the sphere will be varied with a parameter named λ along the normal of the face. and is given by

$$\vec{c}_i(\lambda) = \lambda \vec{n}_i, \quad \lambda \in \mathbb{R}. \tag{3.7}$$

Note we can choose λ per face as well, but our definitions will use the same λ for all faces.

By the definition of spherical flat infinity boxes, we know that this sphere center $\vec{c_i}$ will pass through the center of the face P_i (since its vertices lie on the unit sphere). In addition we know that each vertex of the face has an equal distance to $\vec{c_i}$. Let $\vec{v_i}$ be such a vertex on the face P_i. Then, we set the radius of the sphere equal to

$$r_i(\lambda) = \|\vec{v}_i - \vec{c}_i(\lambda)\|, \tag{3.8}$$

The sphere will pass through all vertices of the face P_i for all λ. In order to now continuously transform points on the face P_i to the sphere defined above, we simply project the points of the face onto the sphere from the origin.

Definition 3.4.1 (Spherical Infinity Boxes)**.** *Given a spherical flat infinity box $F \subset \mathbb{R}^n$. The induced **spherical infinity box** given by $S_\lambda(F) \subset \mathbb{R}^n$ is defined as*

$$S_\lambda(F) = \vec{p}(F) \tag{3.9}$$

Here, $\vec{p}(\vec{x}) = s(\vec{x})\vec{x} : F \to \mathbb{R}^n$ must be a continuous function. The scaling factor $s(\vec{x})$ for a point $\vec{x} \in Hull(P_i)$ on the face P_i with normal $\vec{n}_i$ and a vertex $\vec{v}_i$ must satisfy

$$\|s(\vec{x})\vec{x} - \lambda \vec{n}_i\| = \|\vec{v}_i - \lambda \vec{n}_i\|, \quad \lambda \in \mathbb{R}_\infty. \tag{3.10}$$

Since this condition will have two solutions because we are projecting on a sphere, we take $s(\vec{x}) > 1$ if $\lambda \leqslant 0$ and $s(\vec{x}) < 1$ if $\lambda > 0$. This corresponds to projecting on the outer part and inner part of the sphere respectively.

*λ is called the **curving parameter**. A value λ for which $\vec{p}(\vec{x})$ becomes discontinuous is called a **critical curving parameter**. If $\lambda = \infty$, then $S_\lambda(F)$ is simply the flat-faced infinity box F.*

See Figure 3.4 for a visual explanation of spherical infinity boxes.

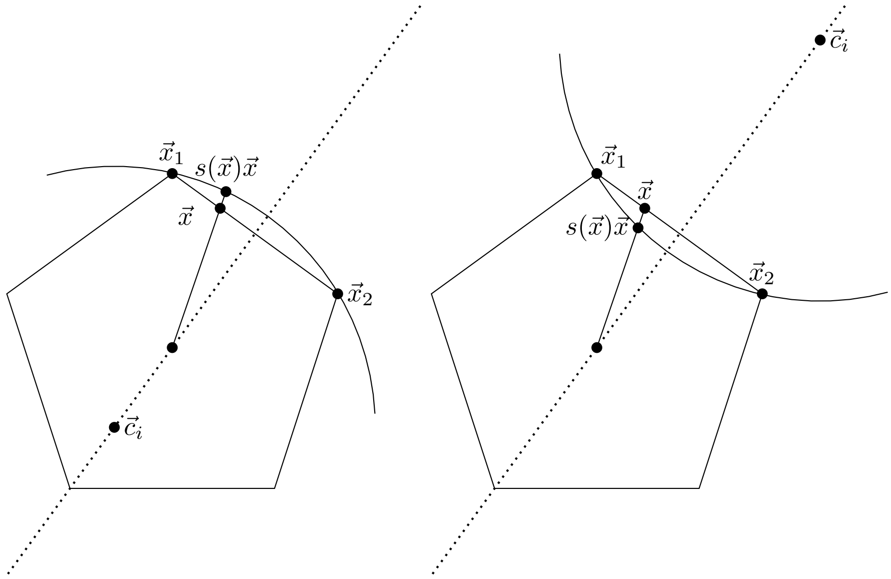


Figure 3.4: Spherical (Left) and Hyperbolic (Right) Infinity Box construction through projection from a flat pentagon infinity box in $\mathbb{R}^2$. A poiint $\vec{x}$ is projected onto the sphere passing through its face by means of scaling with $s\vec{x}$.

3.5 Curving Parameter For Desired Internal Angle

Now that we have defined the spherical infinity boxes, we want to compute values of λ that result in infinity boxes with special properties. In this section we will derive the curving parameter λ for which an infinity box $S_\lambda(F)$ has an internal angle between two faces $\beta \in [0, 2\pi]$ that we can freely choose.

Before we can compute such a formula for λ, we must first derive a formula for the "angle" between two intersecting spheres. It should come intuitively that two intersecting spheres in $\mathbb{R}^3$ have the same angle between their normals anywhere on the intersection surface. We will begin computing this angle in the form of a dot product for an arbitrary pair of hyperspheres.

To bring the concept of "angles" to higher dimensions, we realize that angles between vectors are equivalent to inner products. We will state without proof a common result from linear algebra.

Proposition 1 (Angle Between Two Vectors)**.** *The angle $\alpha \in [0, \pi]$ between two vectors $\vec{a}, \vec{b} \in \mathbb{R}^n$ is given by*

$$\cos(\alpha) = \frac{\langle \vec{a}, \vec{b} \rangle}{||\vec{a}||\,||\vec{b}||} \tag{3.11}$$

We will work with inner products over angles in our results for dimensions higher than $\mathbb{R}^2$. We will now provide the preliminary result for the dot product between normals on the intersection surface of spheres. We will call this dot product the "Sphere Inner Product".

Theorem 5 (Sphere Inner Product)**.** *Given two intersecting spheres* $(\vec{c}_1, r_1), (\vec{c}_2, r_2) \in \mathbb{R}^n \times \mathbb{R}_{>0}$. *The inner product between any two normals* $\vec{n}_1, \vec{n}_2 \in S^{n-1}$ *on the intersection between both spheres is called the* ***Sphere Inner Product*** $S_\bullet$ *and is given by*

$$S_\bullet(\vec{c}_1, r_1; \vec{c}_2, r_2) = \langle \vec{n}_1, \vec{n}_2 \rangle = \frac{r_1^2 + r_2^2 - \|\vec{c}_1 - \vec{c}_2\|^2}{2r_1 r_2} \tag{3.12}$$

Proof. Let $\vec{x}$ lie on both spheres. Starting from the normals we have

$$\begin{aligned}\langle \vec{n}_1, \vec{n}_2 \rangle &= \left\langle \frac{\vec{x} - \vec{c}_1}{r_1}, \frac{\vec{x} - \vec{c}_2}{r_2} \right\rangle \\ &= \frac{1}{r_1 r_2} \Big(\langle \vec{x}, \vec{x} \rangle - \langle \vec{x}, \vec{c}_1 + \vec{c}_2 \rangle + \langle \vec{c}_1, \vec{c}_2 \rangle \Big).\end{aligned}$$

To eliminate the term with $\vec{x}$, we rewrite the equation for $\vec{x}$ to lie on the sphere $\vec{c}$ with radius r,

$$\begin{aligned}\|\vec{x} - \vec{c}\| &= r \\ \langle \vec{x} - \vec{c}, \vec{x} - \vec{c} \rangle &= r^2 \\ \langle \vec{x}, \vec{x} \rangle - 2\langle \vec{x}, \vec{c} \rangle + \langle \vec{c}, \vec{c} \rangle &= r^2.\end{aligned}$$

This results in

$$\langle \vec{x}, \vec{x} \rangle = r^2 - \langle \vec{c}, \vec{c} \rangle + 2\langle \vec{x}, \vec{c} \rangle$$

Since we must use the fact that $\vec{x}$ lies on both spheres, we average the above equality for both spheres. This yields

$$\langle \vec{x}, \vec{x} \rangle = \frac{r_1^2 + r_2^2 - \langle \vec{c}_1, \vec{c}_1 \rangle - \langle \vec{c}_2, \vec{c}_2 \rangle}{2} + \langle \vec{x}, \vec{c}_1 + \vec{c}_2 \rangle.$$

Substituting this back in the inner product of the normals and multiplying by 2/2 results in

$$\frac{1}{2r_1 r_2} \Big(r_1^2 + r_2^2 - \big(\langle \vec{c}_1, \vec{c}_1 \rangle - 2\langle \vec{c}_1, \vec{c}_2 \rangle + \langle \vec{c}_2, \vec{c}_2 \rangle \big) \Big).$$

Now we can recognize the inner term in brackets as simply the inner product $\langle \vec{c}_1 - \vec{c}_2, \vec{c}_1 - \vec{c}_2 \rangle = \|\vec{c}_1 - \vec{c}_2\|^2$, resulting in the formula

$$\frac{r_1^2 + r_2^2 - \|\vec{c}_1 - \vec{c}_2\|^2}{2r_1 r_2}.$$

□

This simple geometric result allows us to directly compute the angle between two intersecting spheres in $\mathbb{R}^n$. If we now take two neighboring faces P_1 and P_2 of a spherical flat infinity box and insert the formulas for the spheres passing through them, $(\vec{c}(\lambda)_1, r_1), (\vec{c}(\lambda)_2, r_2)$, then we can solve for λ such that $S_\bullet = p$ with p our desired inner product between the normals. We present the main result of this section.

Theorem 6 (Curving Parameter For Sphere Inner Product)**.** *Given two faces $P_1, P_2 \subset \mathbb{R}^n$ with normals $\vec{n}_1, \vec{n}_2 \in S^{n-1}$ and a common vertex $\vec{N} \in P_1 \cap P_2$ satisfying $\langle \vec{N}, \vec{n}_1 \rangle = \langle \vec{N}, \vec{n}_2 \rangle > 0$ for all $\lambda \in \mathbb{R}$.*

Given two spheres $(\lambda\vec{n}_1, r), (\lambda\vec{n}_2, r) \in \mathbb{R}^n \times \mathbb{R}_{>0}$ with common radius $r = \|\vec{N} - \lambda\vec{n}_1\|$. Then, the curving parameter λ for which $S_\bullet(\lambda\vec{n}_1, r; \lambda\vec{n}_2, r) = p$ is given by

$$\lambda_{S\pm}(\vec{N}, \vec{n}_1, \vec{n}_2; p) = \frac{q\langle \vec{N}, \vec{n}_1 \rangle \pm \sqrt{q}\sqrt{p - \langle \vec{n}_1, \vec{n}_2 \rangle + q\langle \vec{N}, \vec{n}_1 \rangle^2}}{\langle \vec{n}_1, \vec{n}_2 \rangle - p} \in \mathbb{R}. \tag{3.13}$$

Here $q = 1 - p$ with $|p| \leqslant 1$. A lower bound for p is given by the ***critical angle*** *p^**

$$p^* = \frac{\langle \vec{n}_1, \vec{n}_2 \rangle - \langle \vec{N}, \vec{n}_1 \rangle^2}{1 - \langle \vec{N}, \vec{n}_1 \rangle^2} \leqslant p. \tag{3.14}$$

The value $\lambda_S^* = \lambda_{S\pm}(\vec{N}, \vec{n}_1, \vec{n}_2; p^*)$ is the ***critical curving parameter****. It is given by*

$$\lambda_S^* = \frac{1}{\langle \vec{N}, \vec{n}_1 \rangle} = 2\lambda_0. \tag{3.15}$$

Here, λ_0 is an additional solution (apart from $\lambda = \infty$) that satisfies $p = \langle \vec{n}_1, \vec{n}_2 \rangle$. Finally, we have that $\lambda_{S-} \in [0, \lambda_S^*]$ and $\lambda_{S+} \in (-\infty, 0] \cup [\lambda_S^*, \infty)$.

Proof. We will begin by inserting the spheres in the expression for $S_\bullet$.

$$\begin{aligned} S_\bullet(\vec{c}_1, r; \vec{c}_2, r) &= \frac{r^2 + r^2 - \|\vec{c}_1 - \vec{c}_2\|^2}{2rr} \\ &= 1 - \frac{\|\vec{c}_1 - \vec{c}_2\|^2}{2r^2}. \end{aligned} \tag{3.16}$$

Then we first simplify the numerator. This results in

$$\begin{aligned} \|\vec{c}_1 - \vec{c}_2\|^2 &= \|\lambda\vec{n}_1 - \lambda\vec{n}_2\|^2 \\ &= \lambda^2\|\vec{n}_1 - \vec{n}_2\|^2 \\ &= \lambda^2(\langle \vec{n}_1, \vec{n}_1 \rangle - 2\langle \vec{n}_1, \vec{n}_2 \rangle + \langle \vec{n}_2, \vec{n}_2 \rangle) \\ &= 2\lambda^2(1 - \langle \vec{n}_1, \vec{n}_2 \rangle). \end{aligned} \tag{3.17}$$

In a similar fashion we can simplify the denominator,

$$\begin{aligned} r^2 &= \|\vec{N} - \lambda\vec{n}_1\|^2 \\ &= \langle \vec{N} - \lambda\vec{n}_1, \vec{N} - \lambda\vec{n}_1 \rangle \\ &= \langle \vec{N}, \vec{N} \rangle - 2\lambda\langle \vec{N}, \vec{n}_1 \rangle + \lambda^2\langle \vec{n}_1, \vec{n}_1 \rangle \\ &= 1 + \lambda^2 - 2\lambda\langle \vec{N}, \vec{n}_1 \rangle. \end{aligned} \tag{3.18}$$

Note that using $\vec{n}_1$ or $\vec{n}_2$ in Equation 3.18 does not matter since $\langle \vec{N}, \vec{n}_1 \rangle = \langle \vec{N}, \vec{n}_2 \rangle$. We substitute these two equations back in Equation 3.16, which results in

$$S_\bullet = 1 - \frac{\lambda^2(1 - \langle \vec{n}_1, \vec{n}_2 \rangle)}{1 + \lambda^2 - 2\lambda\langle \vec{N}, \vec{n}_1 \rangle}. \tag{3.19}$$

Rewriting this equation with $p = S_\bullet$ and $q = 1 - p$ yields

$$\begin{aligned} \frac{\lambda^2(1 - \langle \vec{n}_1, \vec{n}_2 \rangle)}{1 + \lambda^2 - 2\lambda\langle \vec{N}, \vec{n}_1 \rangle} &= q \\ \lambda^2(\langle \vec{n}_1, \vec{n}_2 \rangle - p) - 2q\langle \vec{N}, \vec{n}_1 \rangle\lambda + q &= 0 \end{aligned} \tag{3.20}$$

In the case that $\langle \vec{n}_1, \vec{n}_2 \rangle = p$, we obtain the special solution λ_0.

$$\lambda_0 = \frac{1}{2\langle \vec{N}, \vec{n}_1 \rangle}. \tag{3.21}$$

This solution is related to the critical curving parameter as we shall see.

Now we recognize this as a quadratic equation in λ. If $\langle \vec{n}_1, \vec{n}_2 \rangle = p$, then we have $\lambda = \infty$ We first simplify the discriminant.

$$\begin{aligned} D &= (-2q\langle \vec{N}, \vec{n}_1 \rangle)^2 - 4(\langle \vec{n}_1, \vec{n}_2 \rangle - p)q \\ D &= 4q(p - \langle \vec{n}_1, \vec{n}_2 \rangle + q\langle \vec{N}, \vec{n}_1 \rangle^2). \end{aligned} \tag{3.22}$$

Finally, solving the quadratic formula from Equation 3.20 with this discriminant results in

$$\begin{aligned} \lambda_\pm &= \frac{2q\langle \vec{N}, \vec{n}_1 \rangle \pm \sqrt{4q(p - \langle \vec{n}_1, \vec{n}_2 \rangle + q\langle \vec{N}, \vec{n}_1 \rangle^2)}}{2(\langle \vec{n}_1, \vec{n}_2 \rangle - p)} \\ \lambda_\pm &= \frac{q\langle \vec{N}, \vec{n}_1 \rangle \pm \sqrt{q}\sqrt{p - \langle \vec{n}_1, \vec{n}_2 \rangle + q\langle \vec{N}, \vec{n}_1 \rangle^2}}{\langle \vec{n}_1, \vec{n}_2 \rangle - p}. \end{aligned} \tag{3.23}$$

Which is exactly what we wanted to prove. For the condition on p, we rewrite what it means for the terms under the square root to be positive. We already have $|p| \leqslant 1$ so $\sqrt{q}$ is real. The second square root results in

$$\begin{aligned} 0 &\leqslant p - \langle \vec{n}_1, \vec{n}_2 \rangle + q\langle \vec{N}, \vec{n}_1 \rangle^2 \\ 0 &\leqslant p - \langle \vec{n}_1, \vec{n}_2 \rangle + (1 - p)\langle \vec{N}, \vec{n}_1 \rangle^2 \\ 0 &\leqslant p(1 - \langle \vec{N}, \vec{n}_1 \rangle^2) - \langle \vec{n}_1, \vec{n}_2 \rangle + \langle \vec{N}, \vec{n}_1 \rangle^2 \\ p &\geqslant \frac{\langle \vec{n}_1, \vec{n}_2 \rangle - \langle \vec{N}, \vec{n}_1 \rangle^2}{1 - \langle \vec{N}, \vec{n}_1 \rangle^2} = p^*. \end{aligned} \tag{3.24}$$

If we compute the value of λ_S^* for p^*, then we end up with twice λ_0.

$$\begin{aligned}\lambda_S^* &= \frac{q\langle \vec{N}, \vec{n}_1\rangle}{\langle \vec{n}_1, \vec{n}_2\rangle - p} \\ &= \frac{q\langle \vec{N}, \vec{n}_1\rangle}{q\langle \vec{N}, \vec{n}_1\rangle^2} \\ &= \frac{1}{\langle \vec{N}, \vec{n}_1\rangle} = 2\lambda_0.\end{aligned} \tag{3.25}$$

The last thing we must prove is the parameter ranges of λ_{S+} and λ_{S-}. We start by showing under which condition we have that

$$q\langle \vec{N}, \vec{n}_1\rangle \pm \sqrt{q}\sqrt{p - \langle \vec{n}_1, \vec{n}_2\rangle + q\langle \vec{N}, \vec{n}_1\rangle^2} \geqslant 0 \tag{3.26}$$

Clearly, since $q \geqslant 0$ and $\langle \vec{N}, \vec{n}_1\rangle > 0$, for the $\oplus$ term we have that this will be true. Now we consider the circumstances under which it holds for $\ominus$.

$$\begin{aligned} q^2\langle \vec{N}, \vec{n}_1\rangle^2 &\geqslant q(p - \langle \vec{n}_1, \vec{n}_2\rangle + q\langle \vec{N}, \vec{n}_1\rangle^2) \\ q\langle \vec{N}, \vec{n}_1\rangle^2 &\geqslant p - \langle \vec{n}_1, \vec{n}_2\rangle + q\langle \vec{N}, \vec{n}_1\rangle^2 \\ \langle \vec{n}_1, \vec{n}_2\rangle &\geqslant p \end{aligned} \tag{3.27}$$

Comparing this with the expression for λ_{S-}, we have the sign of the numerator and denominator both dependent on $\langle \vec{n}_1, \vec{n}_2\rangle > p$. This means that they are positive and negative at the same time, hence $\lambda_{S-} \geqslant 0$.

For the bounds $\lambda_{S-} \leqslant \lambda_S^* \leqslant \lambda_{S+}$, we analyse the behavior of λ near p^* by setting $p = p^* + \varepsilon \leqslant 1$ with $\varepsilon > 0$. We have already shown equality in Equation 3.25. Now inserting this perturbed p we obtain

$$\begin{aligned}\lambda_\pm &= \frac{(q^* - \varepsilon)\langle \vec{N}, \vec{n}_1\rangle \pm \sqrt{q^* - \varepsilon}\sqrt{p^* + \varepsilon - \langle \vec{n}_1, \vec{n}_2\rangle + (q^* - \varepsilon)\langle \vec{N}, \vec{n}_1\rangle^2}}{\langle \vec{n}_1, \vec{n}_2\rangle - p} \\ &= \frac{(q^* - \varepsilon)\langle \vec{N}, \vec{n}_1\rangle \pm \sqrt{q^* - \varepsilon}\sqrt{\varepsilon(1 - \langle \vec{N}, \vec{n}_1\rangle^2)}}{(q^* - \varepsilon)\langle \vec{N}, \vec{n}_1\rangle^2} \\ &= \frac{\oplus \pm \sqrt{\varepsilon}\cdot \oplus}{\oplus}, \quad \oplus > 0.\end{aligned} \tag{3.28}$$

This means that the if we increase ε to move away from the critical curvature p^*, we will have λ_{S+} increasing and λ_{S-} decreasing. Combining this with $\lambda_{S-} \geqslant 0$ we have that $\lambda_{S-} \in [0, \lambda_S^*]$. Furthermore, we know $\lambda_{S+} \geqslant \lambda_S^*$ if $\langle \vec{n}_1, \vec{n}_2\rangle > p$. This means that $\lambda_{S+} \in (-\infty, 0] \cup [\lambda_S^*, \infty)$. This finishes the proof of the theorem. □

See Figure 3.5 for a visual overview of the bounds on $\lambda_{S\pm}$ and the explanation why we use λ_{S+} for infinity boxes.

Note that $p^* = -1$ when $\vec{n}_2 = \vec{n}_{1*_A}$ (the mirroring of $\vec{n}_1$ about $\vec{N}$), which is equivalent to $\vec{N} \in Span\{\vec{n}_1, \vec{n}_2\}$. In $\mathbb{R}^2$, this will always hold. This means that all possible values for p, and hence all possible angles between two circles of infinity boxes are possible in $\mathbb{R}^2$.

However, in general for $\mathbb{R}^3$ and beyond, we do not have that $\vec{N} \in Span\{\vec{n}_1, \vec{n}_2\}$. This indicates that making higher-dimensional spheres meet at any angle is not possible in general. This is relevant to the curving of infinity boxes in $\mathbb{R}^3$.

Another observation and validation to make on this formula is the value of $\lambda_{S_\pm}$ when we set the vectors $\vec{N} = \vec{n}_1 = \vec{n}_2$ for the geometry of a sphere. A simple computation will reveal that $\lambda_{S\pm} = 1$ regardless of p. In addition, we have $p^* = 1$ after taking a limit such that the only possible value p for a sphere is $p = 1$ as expected.

The λ Curving Parameter Range

One very important thing we wish to note for $\lambda_{\pm S}$ is the fact that there are two solutions provided by Theorem 6. However, only one of these two λ will be the one that we intended to find. We will from now on set $\lambda_{S+} = \lambda_S$ as the curving parameter for spherical infinity boxes. This is because λ_{S+} is the only curving parameter which converges to $\pm\infty$ as $p \to \langle \vec{n}_1, \vec{n}_2 \rangle$ (we want flat infinity boxes as solutions).

This also means that we discard λ_0 for $p = \langle \vec{n}_1, \vec{n}_2 \rangle$ because it falls in the range of λ_{S-}. An overview of the parameter ranges for $\lambda_{S\pm}$ is given below along with their meaning.

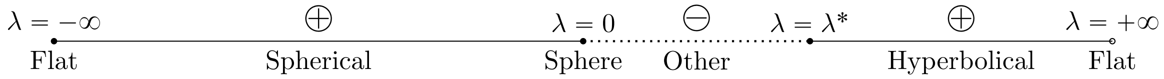


Figure 3.5: Parameter ranges for $\lambda_{S+} = \oplus$ and $\lambda_{S-} = \ominus$. Since we want to include the flat infinity boxes in our parameter range, we pick $\lambda_S = \lambda_{S+}$. The values of λ in the $\ominus$ range are those where the spheres of the spherical infinity box $S_\lambda(F)$ no longer meet at the vertices of faces, making them invalid (the transformation $\vec{p}(\vec{x})$ is no longer continuous). λ^* is called the critical curving parameter.

Specialisation For $\mathbb{R}^2$

In $\mathbb{R}^2$ we can specialise the result from Theorem 6 for polygons with an angle of α between two points and a desired inner angle β. For illustration, we will derive this equation by a geometrical argument rather than to substitute cosines in Theorem 6. The below theorem gives the curving parameter λ for which the internal angle of a circular N-Regular infinity box is equal to β.

Theorem 7 (Curving Parameter For Internal Angle)**.** *Let F be an N-Regular infinity box with angle $\alpha = \frac{2\pi}{N}$ between vertices. Then, the curving parameter $\lambda \in \mathbb{R}$ for which the internal angle between two meeting spheres of $S_\lambda(F)$ is equal to $\beta \in [0, 2\pi]$ is given by*

$$\lambda_S(\alpha, \beta) = \frac{\cos\left(\dfrac{\beta}{2}\right)}{\cos\left(\dfrac{\alpha + \beta}{2}\right)}, \quad \alpha, \beta < 2\pi. \tag{3.29}$$

In addition, the associated circle radius is given by

$$r^2 = 1 + \lambda_A^2 - 2\lambda_A \cos\left(\frac{\alpha}{2}\right). \tag{3.30}$$

Proof. Let the spacing of the points be given by α. Then we can define the circle center $\vec{c}$ as

$$\vec{c} = \lambda \begin{pmatrix} \cos(\frac{\alpha}{2}) \\ \sin(\frac{\alpha}{2}) \end{pmatrix}, \quad \lambda \in \mathbb{R} \tag{3.31}$$

The vector pointing from $\vec{c}(\lambda)$ to $\vec{N} = (1, 0)$ is given by

$$\vec{N} - \vec{c}(\lambda) = \begin{pmatrix} 1 - \lambda\cos\left(\frac{\alpha}{2}\right) \\ -\lambda\sin\left(\frac{\alpha}{2}\right) \end{pmatrix} \tag{3.32}$$

Then, from a geometrical argument, we see that this relates our desired internal angle β in the following way. (See Figure 3.6 for justification to this argument.)

$$\tan\left(\frac{\pi}{2} - \frac{\beta}{2}\right) = \frac{\lambda\sin(\frac{\alpha}{2})}{\lambda\cos(\frac{\alpha}{2}) - 1} \tag{3.33}$$

If we solve for λ, then we end up with

$$\lambda = \frac{\tan\left(\frac{\pi}{2} - \frac{\beta}{2}\right)}{\tan\left(\frac{\pi}{2} - \frac{\beta}{2}\right)\cos\left(\frac{\alpha}{2}\right) - \sin\left(\frac{\alpha}{2}\right)} \tag{3.34}$$

.

We start by simplify the tangent. The tangent can be reduced to

$$\tan\left(\frac{\pi}{2} - \frac{\beta}{2}\right) = \frac{\sin\left(\frac{\pi}{2} - \frac{\beta}{2}\right)}{\cos\left(\frac{\pi}{2} - \frac{\beta}{2}\right)} = \frac{\cos\left(\frac{\beta}{2}\right)}{\sin\left(\frac{\beta}{2}\right)}. \tag{3.35}$$

Now we can simplify the denumerator by means of the cosine sum formula. The cosine sum formula states $\cos(\alpha + \beta) = \cos(\alpha)\cos(\beta) - \sin(\alpha)\sin(\beta)$. We find that

$$\begin{aligned} \tan\left(\frac{\pi}{2} - \frac{\beta}{2}\right)\cos\left(\frac{\alpha}{2}\right) - \sin\left(\frac{\alpha}{2}\right) &= \frac{\cos\left(\frac{\beta}{2}\right)}{\sin\left(\frac{\beta}{2}\right)}\cos\left(\frac{\alpha}{2}\right) - \sin\left(\frac{\alpha}{2}\right) \\ &= \frac{1}{\sin\left(\frac{\beta}{2}\right)}\Big(\cos\left(\frac{\alpha}{2}\right)\cos\left(\frac{\beta}{2}\right) - \sin\left(\frac{\alpha}{2}\right)\sin\left(\frac{\beta}{2}\right)\Big) \\ &= \frac{\cos\left(\frac{\alpha+\beta}{2}\right)}{\sin\left(\frac{\beta}{2}\right)}. \end{aligned} \tag{3.36}$$

Now substituting this in Equation 3.33 results in

$$\lambda = \frac{\cos\left(\dfrac{\beta}{2}\right)}{\cos\left(\dfrac{\alpha+\beta}{2}\right)}. \tag{3.37}$$

Which is the primary result that we wanted to prove. To find the associated arc radius, we simply take the length of the vector $\vec{N} - \vec{c}$. This finishes the proof.

□

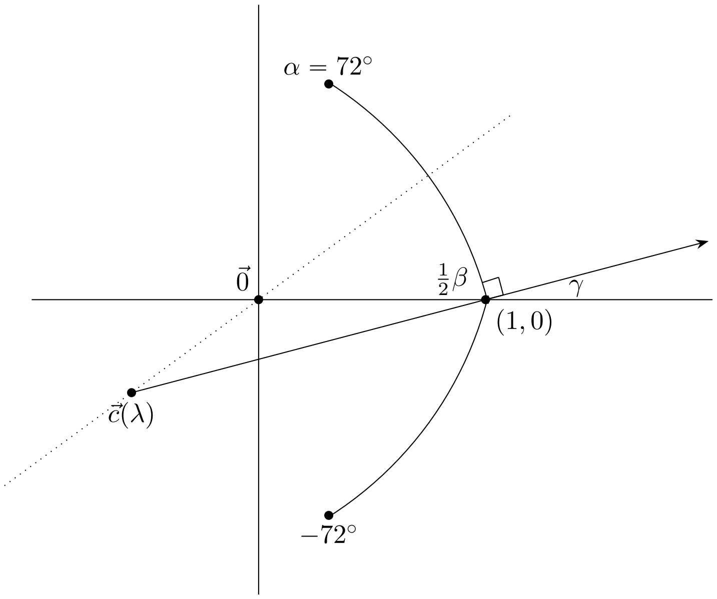


Figure 3.6: Derivation of the internal angle alignment. A tangent is constructed from the line passing though $\vec{c}(\lambda)$ and $(1,0)$.

Note that Theorem 7 only provides one solution. This solution is equal to λ_{S+} from Theorem 6, due to the nature of the derivation. We do not need to worry about a second solution like λ_{S-} in this argument. The critical curving parameter in this case can also be simply found as

$$\lambda_S(\alpha, 2\pi) = \frac{1}{\cos\left(\dfrac{\alpha}{2}\right)}. \tag{3.38}$$

We can immediately note that as $\alpha \to \infty$, then $\lambda_S \to 1$. In other words, the critical curving parameter of a unit circle is 1 (where $\lambda\vec{n}$ lies exactly on the circle itself). We will now turn our attention to the second curving parameter of interest: the image aligning curvature parameter for which the first-order image points in an infinity box come together.

3.6 Curving Parameter For First-Order Image Point Alignment

A second result for spheres that we will later use with infinity boxes is the formula for the curvature λ for which the sphere normal forms is orthogonal specified vector.

Theorem 8 (Curvature For Normal Alignment)**.** *Given an incident vector and outgoing vector, $\vec{d}, \vec{r} \in S^{n-1}$ respectively. Let $\vec{N} \in \mathbb{R}^n$ be a point on the sphere with $(\lambda\vec{v}, r) \in \mathbb{R}^n \times \mathbb{R}_{>0}$ with $\lambda \in \mathbb{R}$. The value of λ for which the sphere normal $\vec{n}$ at $\vec{N}$ satisfies $\langle \vec{d}, \vec{n}\rangle = \langle \vec{-r}, \vec{n}\rangle$ is given by*

$$\lambda_N(\vec{N}, \vec{v}; \vec{d}, \vec{r}) = \frac{\langle \vec{d} + \vec{r}, \vec{N}\rangle}{\langle \vec{d} + \vec{r}, \vec{v}\rangle}. \tag{3.39}$$

That is, this is the curving parameter for which an incident vector $\vec{d}$ and outgoing vector $\vec{r}$ have an equal angle to $\vec{n}$. (In general, this does not imply that $\vec{r}$ is the reflection of $\vec{d}$).

Proof. The inner product between the sphere normal at $\vec{N}$ and $\vec{p} = \vec{d} + \vec{r}$ is given by

$$\begin{aligned} \frac{1}{r}\left\langle \vec{p}, \vec{N} - \lambda\vec{v}\right\rangle &= \\ \frac{1}{r}\left(\langle \vec{p}, \vec{N}\rangle - \lambda\langle \vec{p}, \vec{v}\rangle\right) &= \\ \frac{1}{r}\left(\langle \vec{p}, \vec{N}\rangle - \langle \vec{p}, \vec{N}\rangle\right) &= 0. \end{aligned} \tag{3.40}$$

Where $r = \|\vec{N} - \lambda\vec{n}\| > 0$ is the radius of the sphere. □

The curvature required for aligning two image points in an infinity box is important in a first attempt at aligning infinity boxes. The sphere normal alignment formula from Theorem 8 provides exactly what is needed. Given an incident vector $\vec{d}$ and a desired reflected vector $\vec{r}$, Theorem 8 gives the curvature for which reflection of $\vec{d}$ at the point $\vec{N}$ is equal to $\vec{r}$.

Note that Theorem 8 only grants that $\vec{d}_* = \vec{r}\ \vec{d}_*$ given that $\vec{N}, \vec{v} \in Span\{\vec{d}, \vec{r}\}$ (the four vectors span a two-dimensional subspace). For $\mathbb{R}^3$ and higher, care must be taken to check if this condition is true. Otherwise λ_N does not ensure that the ray with direction $\vec{d}$ gets reflected to $\vec{r}$ (which is what we seek).

In $\mathbb{R}^2$ we can again achieve a nicer result for N-Regular infinity boxes. This formula is provided here. We also have from the reasoning above that $\vec{d}_* = \vec{r}$.

Theorem 9 (Curving Parameter For First-Order Image Point Alignment)**.** *Given an N-Regular infinity box F with point spacing $\alpha = \frac{2\pi}{N}$. The curving parameter λ for which the reflection of the incident vector $\vec{d} = (1, 0)$ at the point $\vec{N} = (1, 0)$ of the circular infinity box $S_\lambda(F)$ points towards the vertex with angle α_n is given by*

$$\lambda_{I_1}(\alpha, n) = \frac{1 - \sin\left(n\frac{\alpha}{2}\right)}{\cos\left(\frac{\alpha}{2}\right) - \sin\left((n+1)\frac{\alpha}{2}\right)} \leqslant 0, \quad 1 < n < \frac{N}{2}. \tag{3.41}$$

If $\lambda_{I_1} > 0$, then the resulting infinity box S_λ is no longer convex, making λ_{I_1} invalid in general. This is because the ray can get obstructed when it gets reflected.

Proof. Using Theorem 8, set $\vec{d}$ and $\vec{N}$ as given, and $\vec{r}$ equal to

$$\vec{r}_n = \begin{pmatrix} -\sin\left(n\frac{\alpha}{2}\right) \\ \cos\left(n\frac{\alpha}{2}\right). \end{pmatrix}$$

Here $\vec{r}_i$ is the ray pointing from $\vec{N}$ to the vertex with angle α_i. This expression is derived by observing that the angle γ of $\vec{r}$ is given by $n\alpha + 2(\pi - \gamma) = \pi$, since the vertex with angle α_n and $\vec{N}$ form a symmetric triangle. Now, applying Theorem 8 results in

$$\frac{\langle \vec{d} + \vec{r}, \vec{N} \rangle}{\langle \vec{d} + \vec{r}, \vec{n} \rangle} = \frac{1 - \sin\left(n\frac{\alpha}{2}\right)}{\cos\left(\frac{\alpha}{2}\right) - \sin\left((n+1)\frac{\alpha}{2}\right)}.$$

By using the summation formula for sines. □

Theorem 9 is used to connect N-Regular infinity boxes at their first image point. Due to the symmetry of N-Regular infinity boxes at the right point $(1, 0)$, rays that reflect with either mirror normal will both end up in opposing corner vertices. This results in the world image points of these rays to coincide.

Formulas to align higher-order image points are beyond the scope of this paper, as the formulas for computing them will require the collision formula for spheres due to the second collision point no longer being fixed like $\vec{N}$.

3.7 Tables With Common Curving Parameters

In the following sections we will provide tables of curving parameters λ for commonly used infinity boxes. These curving parameters are later used in numerical simulations. To compute the tables, we will use the main results from Section 3.5 and Section 3.6. **Note**: The sphere and circle radii $r(\lambda)$ for each value of λ can be computed from $r(\lambda) = \|\vec{N} - \lambda\vec{n}\|$ with $\vec{N}$ a vertex of the polygon/polyhedron and $\vec{n}$ the face normal of the face connected to $\vec{N}$. All results are subject to numerical imprecisions, and rounding errors of approximately ± 0.01. Exact expressions can be computed for all values.

For reference in computing radii, the radius of the image-aligned dodecahedron infinity box computed from the curving parameter $\lambda = -11.4686$ (Dodecahedron λ_N Table 3.8) is given by $r = 12.2782$. The radius of the 120° internal angle dodecahedron infinity box with $\lambda = -15.6483$ (Dodecahedron $\lambda_S(\cos(60^\circ)))$ is given by $r = 16.4541$.

A value $\lambda < 0$ indicates a convex spherical infinity box. A value of $\lambda = 0$ indicates a fully spherical infinity box. A value of $\lambda > 0$ indicates a concave spherical infinity box. A value of $\lambda = \pm\infty$ indicates a flat infinity box. The closer λ is to zero, the more curvature the infinity box will have. For an overview of the meaning of the λ parameter, see Figure 3.5 (λ Parameter Range).

3.7.1 N-Regular Polygons

We have listed the curving parameters for N-Regular polygons in Table 3.7. The critical curving parameters $\lambda_S^* = \lambda_S(\alpha, 360^\circ)$ have been provided as well.

N	α	$180^\circ - \alpha$	$\lambda_{I_1}(\alpha, 1)$	$\lambda_S(\alpha, 120^\circ)$	λ_S^*
3	120°	60°	-0.3660	-1	2
4	90°	90°	-1	-1.9318	1.4142
5	72°	108°	-2.9021	-4.7833	1.2360
6	60°	120°	∞	∞	1.1547
7	51.42°	128.58°	Concave	6.6907	1.1098
8	45°	135°	Concave	3.8306	1.0823
9	40°	140°	Concave	2.8793	1.0641
10	36°	144°	Concave	2.4048	1.0514

Figure 3.7: The curving parameters λ_{I_1} (first-order image alignment) and λ_S (120° internal angle alignment) for the N-Regular polygons. We omit $\lambda_{I_1} > 0$ because then the resulting infinity box is convex, making Theorem 9 invalid.

3.7.2 Platonic Solids

In this section we will look at the curving parameters of interest for the tetrahedron, cube, dodecahedron, octahedron and icosahedron. The process of computing the curving parameters $\lambda_S(\vec{N}, \vec{n}_1, \vec{n}_2; p)$ and $\lambda_{I_1} = \lambda_N(\vec{N}, \vec{v}; \vec{d}, \vec{r})$ requires us to find the vertices, face normals and incoming- and outcoming ray directions for each of them. For the sphere alignment angle p required for λ_S, we set $p = \cos(180^\circ - 120^\circ) = \cos(60^\circ)$. This is because the relation between the internal angle β and the angle between the normals γ is given by

$$\beta + \gamma + \pi = 2\pi. \tag{3.42}$$

Name	β	λ_N	$\lambda_S(\cos(60°))$	λ_S^*	p^*
Tetrahedron	70.52°	−0.3797	−1.0000	3.0001	−0.4999
Cube	90°	−1	−1.7320	1.7320	−0.5
Octahedron	109.47°	Non-planar	−4.1815	1.7320	0
Dodecahedron	116.56°	−11.4686	−15.6483	1.2584	−0.4998
Icosahedron	138.19°	−3.3993	2.3849	1.2581	0.3090

Figure 3.8: Curving parameters for the platonic solids. The curving for image alignment is performed by curving the mirrors until a ray cast at one vertex ends up in another. The curving for internal angles is performed by applying Theorem 6 to two faces of the polyhedron. The octahedron is the only shape for which it is not possible to align image points using Theorem 8.

Taking a look at Table 3.8, we can see that the icosahedron requires a positive curving parameter $\lambda_S > 0$ in order to obtain a internal angle of 120°. This makes sense because its internal angle is greater than 120°. This table also clearly illustrates duality between the cube and the octahedron, and the dodecahedron and icosahedron, by looking at the value of λ_S^*. It appears to be identical for polyhedra that are dual to each other. However, the corresponding angle lower bound p^* is not correlated at all.

The reason why we could not provide a value λ_N for the octahedron is because a ray cast from the center at one vertex can never reflect into another vertex. See Figure 3.9

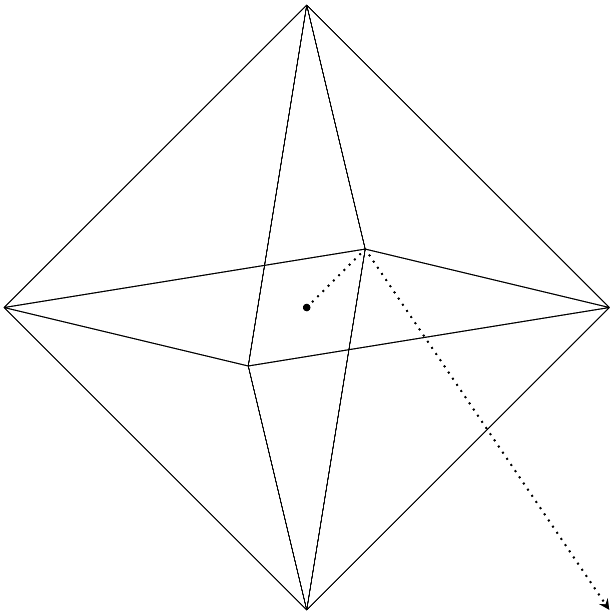


Figure 3.9: The octahedron can not be aligned to image points. A ray cast at one of the vertices from the origin, can never strike another vertex by curving the mirrors.

4 Ray Collision Testing

In this Chapter we will elaborate on finding the collision step Φ for use in numerical simulations and ray tracing. A collision step $\Phi(\vec{o}, \vec{d}) \in \mathbb{R}_\infty$ is a function on a ray that gives the distance that the ray can travel until it collides with a surface. These surfaces can be anything from planes to spheres. The collision step is needed to simulate how a ray moves through the infinity box.

We will begin with the primitive collision formulas of rays with planes and spheres in any dimension.

4.1 Plane Collisions

The simplest form of collision detection of a ray is with an infinite plane prescribed by a single normal. This result is used in simulating flat infinity box mirrors.

Theorem 10 (Ray Collision Step For Planes)**.** *Let all vectors lie in $\mathbb{R}^n$. Given a ray $(\vec{o}, \vec{d})$ and a plane with normal $\vec{n}$. The collision step $\Phi_P(\vec{n}; \vec{o}, \vec{d})$ for which the ray $\vec{o} + t\vec{d}$ lies on the plane is given by*

$$\Phi_P(\vec{c}, \vec{n}; \vec{o}, \vec{d}) = \frac{\langle \vec{c} - \vec{o}, \vec{n} \rangle}{\langle \vec{d}, \vec{n} \rangle} \tag{4.1}$$

Proof. We insert the condition on Φ in the equation for the plane to obtain

$$\begin{aligned} \langle \vec{o} + \Phi\vec{d}, \vec{n} \rangle = \langle \vec{o}, \vec{n} \rangle + \Phi\langle \vec{d}, \vec{n} \rangle &= \langle \vec{c}, \vec{n} \rangle \\ \Phi &= \frac{\langle \vec{c} - \vec{o}, \vec{n} \rangle}{\langle \vec{d}, \vec{n} \rangle}. \end{aligned}$$

This yields the desired result. □

Note that the solution is ∞ if $\vec{d}$ and $\vec{n}$ are orthogonal. We now turn our attention to sphere collisions.

4.2 Sphere Collisions

For simulating of curved mirrors, we will need to collide with spheres. We first provide an intermediate result that simply computes the intersections of a ray and a sphere without taking into account which intersection is the first the ray will hit.

Theorem 11 (Ray-Sphere Intersection)**.** *Let all vectors lie in $\mathbb{R}^n$. Given a ray $(\vec{o}, \vec{d})$ and a sphere with center $\vec{c}$ and radius r. The values $t_\pm$ for which the ray $\vec{o} + t_\pm\vec{d}$ lies on the sphere are given*

$$t_\pm = \langle\vec{c} - \vec{o}, \vec{d}\rangle \pm \sqrt{\langle\vec{c} - \vec{o}, \vec{d}\rangle^2 + r^2 - \|\vec{c} - \vec{o}\|^2}, \tag{4.2}$$

under the condition that the discriminant is not negative,

$$D = \langle\vec{c} - \vec{o}, \vec{d}\rangle^2 + r^2 - \|\vec{c} - \vec{o}\|^2 \geqslant 0. \tag{4.3}$$

Proof. We insert the equation for the ray $\vec{o} + t\vec{d}$ in the equation for the sphere to obtain

$$\begin{aligned} \langle\vec{o} + t\vec{d} - \vec{c}, \vec{o} + t\vec{d} - \vec{c}\rangle &= r^2 \\ t^2\langle\vec{d}, \vec{d}\rangle - 2t\langle\vec{c} - \vec{o}, \vec{d}\rangle + \|\vec{c} - \vec{o}\|\|^2 - r^2 &= 0. \end{aligned}$$

Now let A be $\langle\vec{d}, \vec{d}\rangle = 1$, B be $2\langle\vec{o} - \vec{c}, \vec{d}\rangle$ and C be $\|\vec{c} - \vec{o}\|^2 - r^2$ then we have

$$\begin{aligned} t_\pm &= \frac{-B \pm \sqrt{B^2 - 4AC}}{2A} \\ &= \frac{-2\langle\vec{o} - \vec{c}, \vec{d}\rangle \pm \sqrt{4\langle\vec{o} - \vec{c}, \vec{d}\rangle^2 - 4(\|\vec{c} - \vec{o}\|^2 - r^2)}}{2} \\ &= \langle\vec{c} - \vec{o}, \vec{d}\rangle \pm \sqrt{\langle\vec{c} - \vec{o}, \vec{d}\rangle^2 + r^2 - \|\vec{c} - \vec{o}\|^2)} \end{aligned} \tag{4.4}$$

Recognizing the discriminant under the root yields the desired result. □

The discriminant under the square root determines the number of intersections the ray has with the sphere along its trajectory. This number can be zero if the ray misses ($D < 0$), one if the ray "touches" ($D = 0$) and two if the ray enters and leaves ($D > 0$). If we ignore "touch" collisions with $D = 0$, it is clear that if $D \leqslant 0$, then the ray has no collisions with the sphere. Using this information, we will create two collision step functions, one for colliding with the interior and one for colliding with the exterior of a sphere.

Corollary 11.1 (Ray Collision Step For Spheres)**.** *Let all vectors lie in $\mathbb{R}^n$. Given a ray $(\vec{o}, \vec{d})$ and a sphere with center $\vec{c}$ and radius r Let D. Then, given the discriminant D and intersections points $t_\pm$ of the ray $\vec{o} + t\vec{d}$, the collision steps $\Phi_{S\pm}(\vec{c}, r; \vec{o}, \vec{d})$ are given by*

$$\Phi_{S\pm}(\vec{c}, r; \vec{o}, \vec{d}) = \begin{cases} \max\{t_\pm\} & D > 0, \quad S_+ \\ \min\{t_\pm\} & D > 0, \quad S_- \\ \infty & D \leqslant 0. \end{cases} \tag{4.5}$$

Here, Φ_{S+} and Φ_{S_-} are for colliding with the interior and exterior of the sphere respectively.

Two cases for ray-sphere collisions are illustrated in Figure 4.1 and Figure 4.2. These correspond to exterior and interior collisions. In our numerical simulations, we discard collisions with $\Phi \leqslant 0$ or $\Phi = \infty$.

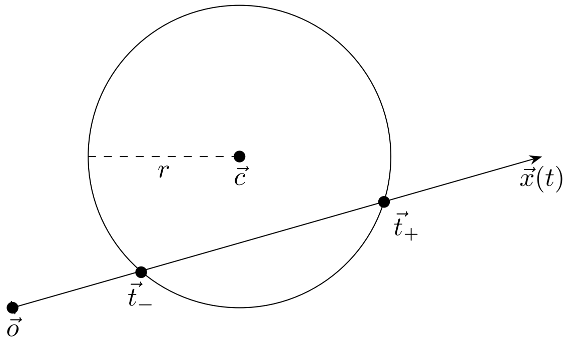


Figure 4.1: Exterior collision of a ray with a sphere.

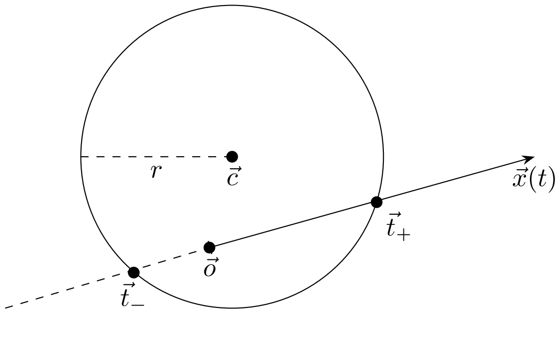


Figure 4.2: Interior collision of a ray with a sphere, with $t_- < 0$ ignored.

4.3 Infinity Box Collisions

When colliding with multiple spheres at once, we must take care to collide only with the domains of the sphere that we are interested in. For spherical infinity boxes, we want that the edges of face act as the boundary of the sphere that is passing through that face. We do not want to collide with the parts of a sphere that do not originate from the projecting corresponding face onto the sphere. This is important when compositing multiple sphere collisions together in a spherical infinity box. If we do not properly test the collisions with the spheres, then we can end up with situations where a ray collides with the part of a sphere that passes through the interior of the infinity box. See Figure 4.3.

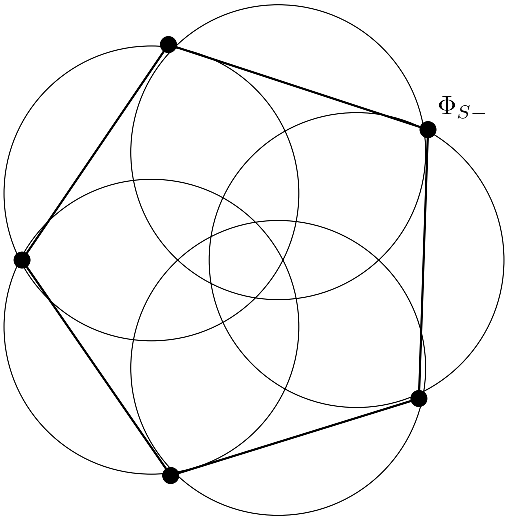


Figure 4.3: All circle arcs inside the pentagon are not part of the infinity box. When performing collision tests on multiple spheres, it is necessary to check if the collision is part of the infinity box.

Because of this collision issue, we must define a rule that checks if a collision point is part infinity box. We will state that rule here.

Theorem 12 (Spherical Infinity Box Point Condition)**.** *Given a spherical flat infinity box $F \subset \mathbb{R}^n$ and one of its faces $P \subset \mathbb{R}^n$ with normal $\vec{n} \in S^{n-1}$ and a vertex $\vec{v} \in P$. Let $\vec{x} \in \mathbb{R}^n$ be a collision point with the sphere with center $\lambda\vec{n}$ and radius r that passes through the face P. If there exists a $p > 0$ such that $p^{-1}\vec{x} \in P$, then $\vec{x} \in S_\lambda(F)$. Furthermore, if p exists, then it is given by*

$$p = \frac{\langle \vec{x}, \vec{n} \rangle}{\langle \vec{v}, \vec{n} \rangle}. \tag{4.6}$$

Proof. Suppose that there exists a $p > 0$ such that $p^{-1}\vec{x} \in P$. Then by definition of the spherical infinity box, there exists an $s > 0$ such that

$$\|sp^{-1}\vec{x} - \lambda\vec{n}\| = r. \tag{4.7}$$

However, since $\vec{x}$ is a collision point with that same sphere, we must have that $cp^{-1} = 1$. Hence $c = p$. Therefore $\vec{x} = p \cdot p^{-1}\vec{x}$ with $p^{-1}\vec{x} \in P$, and by definition of the spherical infinity box $\vec{x} \in S_\lambda(F)$.

Furthermore, since $p^{-1}\vec{x} \in P$, we have

$$\begin{aligned} \langle p^{-1}\vec{x}, \vec{n}\rangle &= \langle \vec{v}, \vec{n}\rangle \\ p &= \frac{\langle \vec{x}, \vec{n}\rangle}{\langle \vec{v}, \vec{n}\rangle}. \end{aligned} \tag{4.8}$$

Which is what we wanted to prove. □

We can use this result to derive a condition for sphere collisions in an infinity box. Suppose we found a collision point $\vec{o} + \Phi\vec{d}, \Phi > 0$ with one of the spheres passing through the face P of the infinity box. Then we know that if $p^{-1}(\vec{o} + \Phi\vec{d}) \in P$ with p given by Equation 4.6, that $\vec{o} + \Phi\vec{d} \in S_\lambda(F)$, making the collision valid.

In $\mathbb{R}^2$, the face P will be a line segment. So the test if $p^{-1}\vec{x}$ reduces to checking if $p^{-1}\vec{x}$ lies on the line P. In $\mathbb{R}^3$, the face P will be a convex polygon, in which case we must check if $p^{-1}\vec{x}$ lies on the convex polygon. This theorem is used during numerical simulations to validate the collisions with curved mirrors.

5 Numerical Simulations Of Infinity Boxes

5.1 Introduction

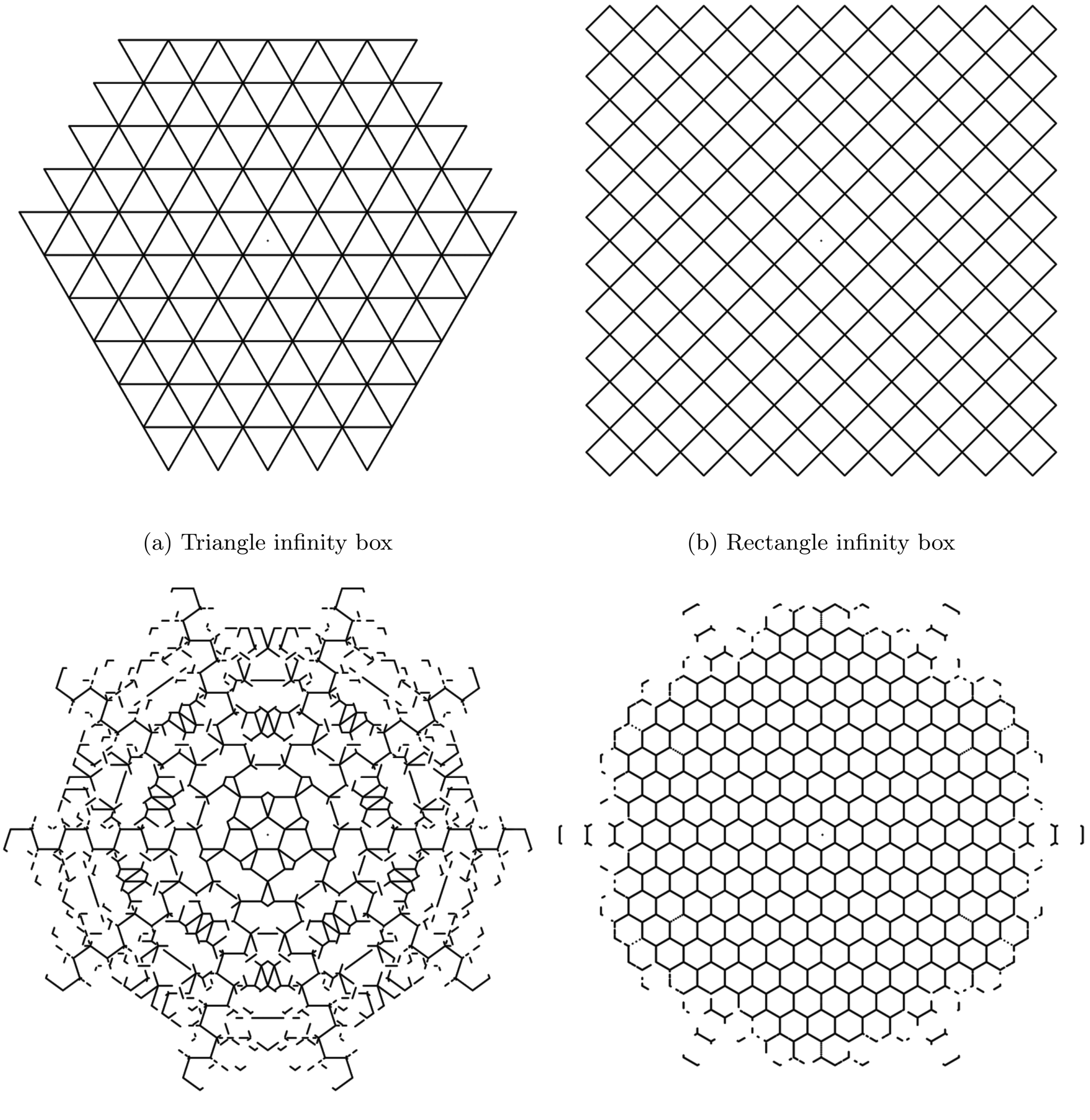
We have laid the fundamental theory behind infinity boxes in the preceding chapters, now we wish to apply this theory to better understand the patterns that emerge from infinity boxes in two and three dimensions. To simulate infinity boxes, we use a computer program named "Mirai" that can draw and export infinity boxes in 2D and 3D. This program is split in two components called Mirai2D and Mirai3D. We will go over 2D and 3D infinity boxes created by MiRai in detail in the coming sections. For an explanation of the computer implementation of MiRai, see Section 6.

5.2 Two-Dimensional Infinity Boxes

The next sections will show results from simulating two-dimensional infinity boxes. What is visualised in the figures that follow are the world-image points of all rays that start from the origin with an angle from 0 to 2π. Every time that the ray collides with a mirror surface, a point is drawn from the origin in the initial direction, with step equal to the total distance travelled up until that collision. This way we get a "birds-eye" view of all the reflections of the infinity box from above, which is not possible in 3D since we would need to view from the fourth dimension to see all of 3D at once.

This allows us to get a very strong intuition for the patterns created by infinity boxes, which makes us better prepared for the patterns from 3D infinity boxes, since they can look more chaotic at first glance. We will begin with N-Regular infinity boxes whose patterns we are already familiar with for the rectangle, hexagon and triangle.

5.2.1 N-Regular Polygons



(a) Triangle infinity box

(b) Rectangle infinity box

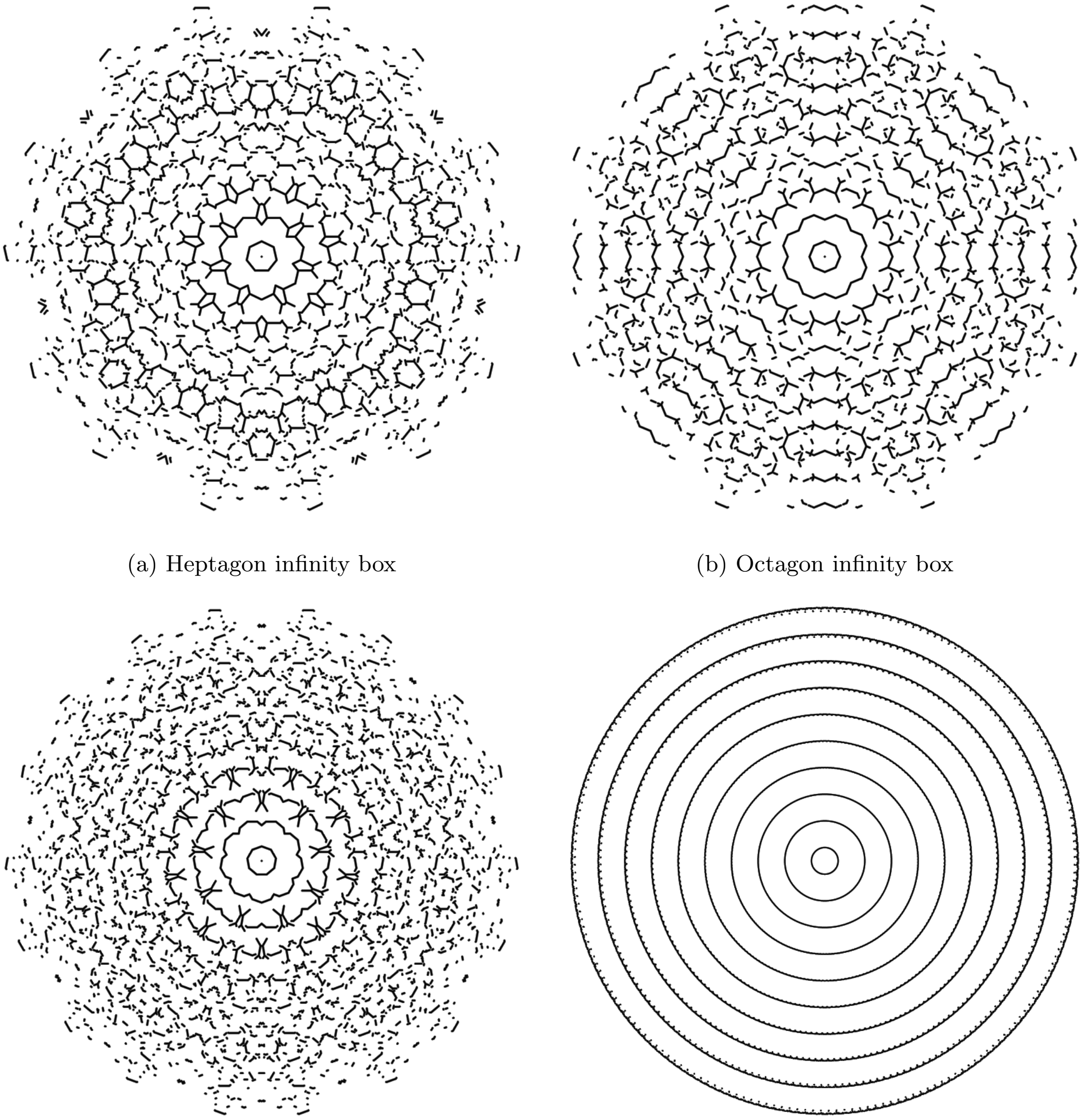
(c) Pentagon infinity box

(d) Hexagon infinity box

Figure 5.1: N-Regular infinity box simulations for $N = 3$ (triangle) to $N = 6$ (hexagon) with depth 10. All shapes except for the pentagon create a mirror-tiling pattern.

While the pentagon might look chaotic from the outset, there is still a high degree of symmetry between the "spokes" of the pattern. The spokes are the repeating patterns between the chaotic parts of the infinity box, they can be seen at poking out at the end of the pentagon infinity box. We will see that, in general, every N-Regular infinity box has N spokes if N is even and $2N$ spokes if N is odd. This is not surprising when N is even, since two opposite mirrors facing each other cause the ray to keep reflecting between them.

When considered as dynamical billiard systems, these infinity boxes give a visual solution to the trajectories of a particle released from the origin. The chaotic nature of non-tiling infinity boxes also explains why finding an exact solution to dynamical billiard systems has been subject to much study.



(a) Heptagon infinity box

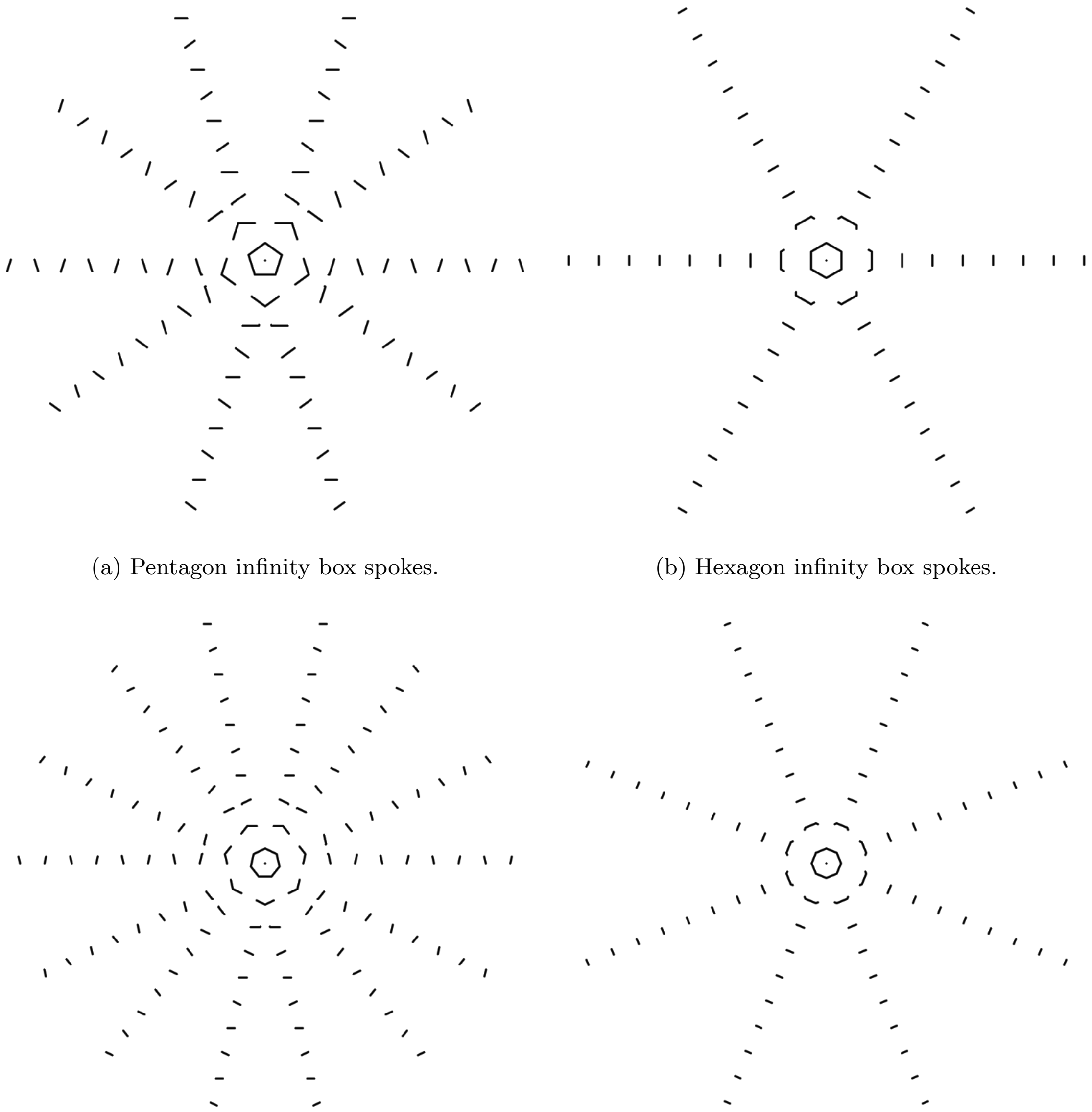
(b) Octagon infinity box

(c) 9-Regular infinity box

(d) 100-Regular (near-circle) infinity box

Figure 5.2: N-Regular infinity box simulations for $N = 7$, $N = 8$, $N = 9$ and $N = 100$ (near-circular) with depth 10. As can be seen with $N = 100$, as the N-Regular polygon transitions closer and closer to a circle, the infinity box will more and more like the circle infinity box as well.

Note that the pattern of spokes continues in the above figure as well, but these are harder to see. If we stop rays in our simulation when they come near a corner, we can filter out the spokes. These results are shown in Figure 5.3.



(a) Pentagon infinity box spokes.

(b) Hexagon infinity box spokes.

(c) Heptagon infinity box spokes.

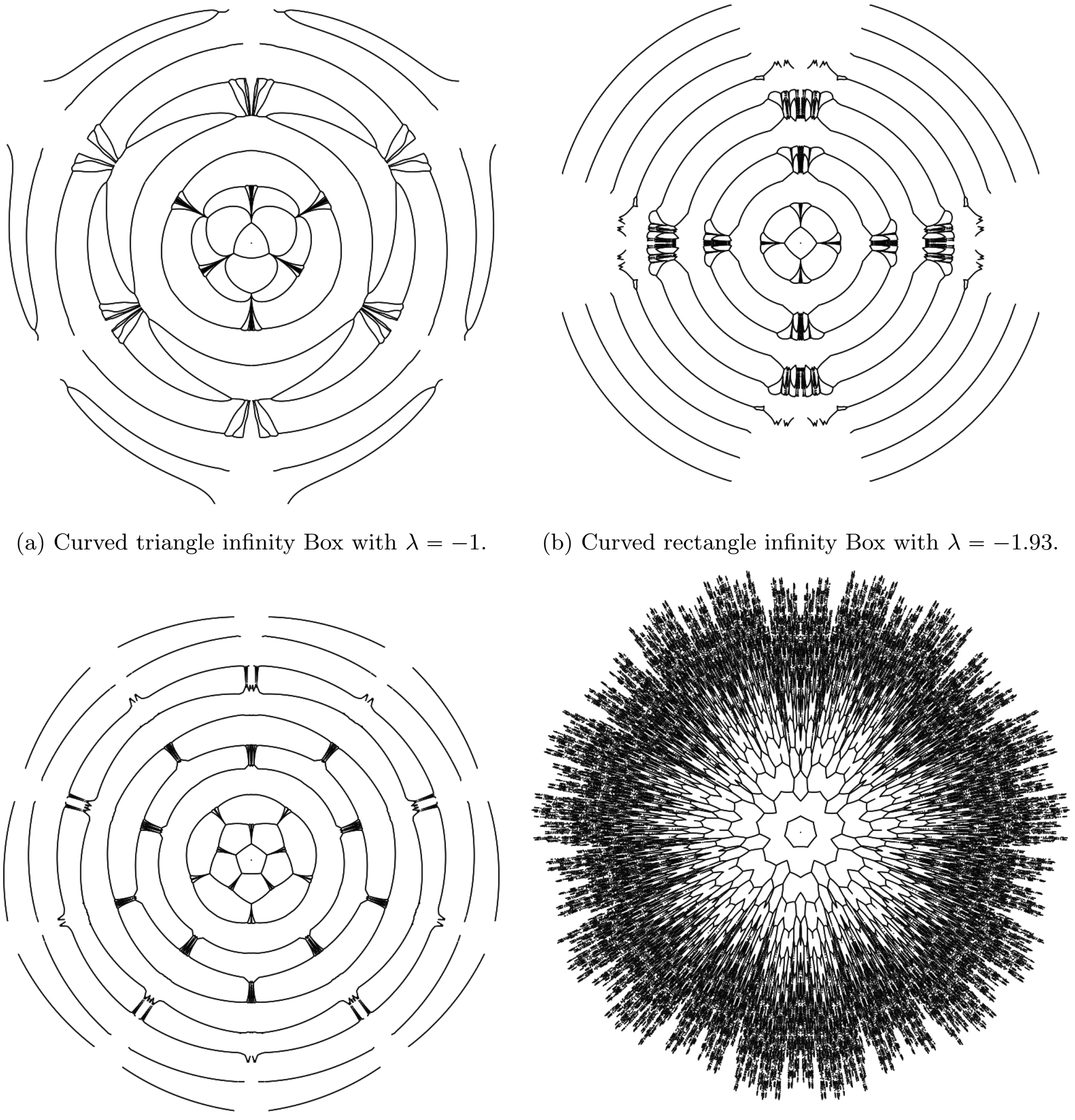
(d) Octagon infinity box spokes.

Figure 5.3: N-Regular infinity box simulations for $N = 5$ to $N = 8$ at depth 10 with spokes visualised by eliminating rays when they collide sufficiently close to a corner. For even N there are N spokes, for odd N there are $2N$ spokes.

When considering infinity boxes as visualisation of dynamical billiard systems (particles reflecting in a closed environment). Then the spokes are exactly the cyclic trajectories of the billiard system. These are not trivial to compute, even for triangles. But with the help of infinity boxes, the solution is immediately visible. Note that all the infinity boxes we visualise have rays cast from the origin, this means that we are visualising all billiard trajectories from the origin. It is possible that such cyclic trajectories do not pass through the origin, which would require the origin to be moved in order to find the correct spot. Note that the Theorems on the existence of cyclic trajectories from Chapter 2 prove the existence of spokes in infinity boxes. Spokes in infinity boxes and cyclic trajectories are equivalent concepts.

In the next Section we will curve the infinity boxes to obtain an internal angle of $2\pi/3$, in the hopes of creating patterns that remind of a tiling of $\mathbb{R}^2$. We will see that when we curve the infinity boxes to become more circular, the spokes expand until they connect.

5.2.2 N-Regular Polygons Curved To Internal Angle 120°



(a) Curved triangle infinity Box with $\lambda = -1$.

(b) Curved rectangle infinity Box with $\lambda = -1.93$.

(c) Curved pentagon infinity Box with $\lambda = -4.78$.

(d) Curved heptagon infinity box with $\lambda = 6.69$ resembling hyperbolic space.

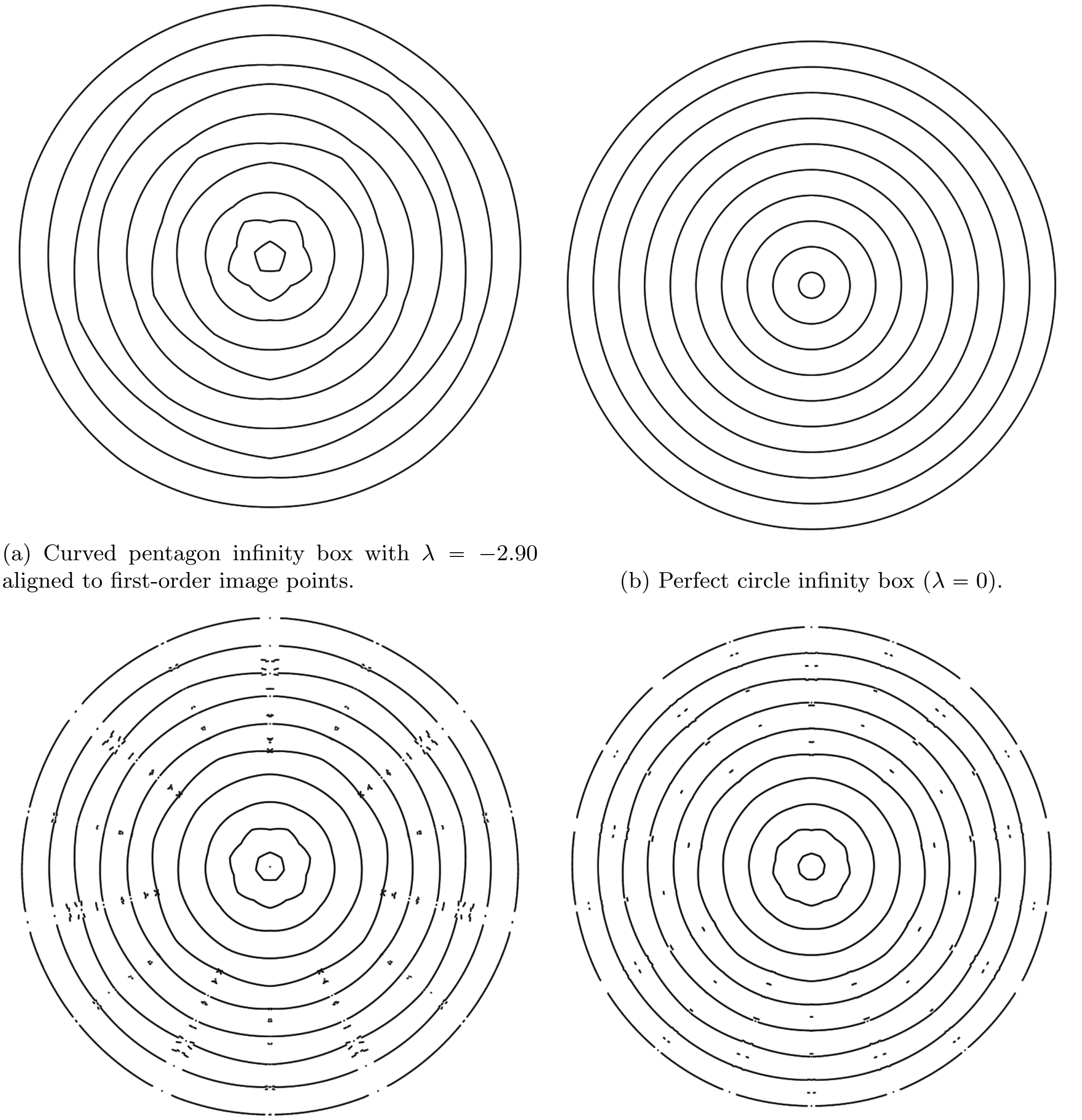
Figure 5.4: N-Regular circular infinity box simulations for $N = 3, 4, 6$ and 7 at depth 10 with the curving parameter $\lambda = \lambda_A(2\pi/N, 2\pi/3)$. The curving parameters λ used in this figure can be found in Table 3.7. Observe that the "spokes" from the previous of the infinity boxes expand for $\lambda \leqslant 0$. This can be seen clearly for the rectangle infinity box, the circular regions on the corners are the expanded spokes.

We can see for the triangle, rectangle and pentagon that the internal angle is $\pi/3$ by looking closely at the first mirrored reflection. Three edges will meet at the corners of each infinity box, though this is not visible for the curved heptagon infinity box.

We can see that the curved pentagon infinity box closely resembles the fully circular infinity box from Figure 5.2 (100-Regular infinity box). If we compare the curved pentagon infinity box to its flat variant from Figure 5.1, we can see that much of the chaotic pattern is eliminated by curving the mirrors. Note that the spokes have expanded for the triangle, rectangle and pentagon, while they have contracted for the heptagon. (Compare with Figure 5.3.)

The highly chaotic pattern near the rim in the curved heptagon infinity box is due the nature of convex mirrors. The trajectories of two rays with a nearly-identical initial direction will quickly diverge, hence resulting in world-image points with highly varying distances.

5.2.3 N-Regular Polygons Curved To Align Image Points



(a) Curved pentagon infinity box with $\lambda = -2.90$ aligned to first-order image points.

(b) Perfect circle infinity box ($\lambda = 0$).

(c) Heptagon infinity box with $\lambda = -2.76$ aligned by hand to second-order image points.

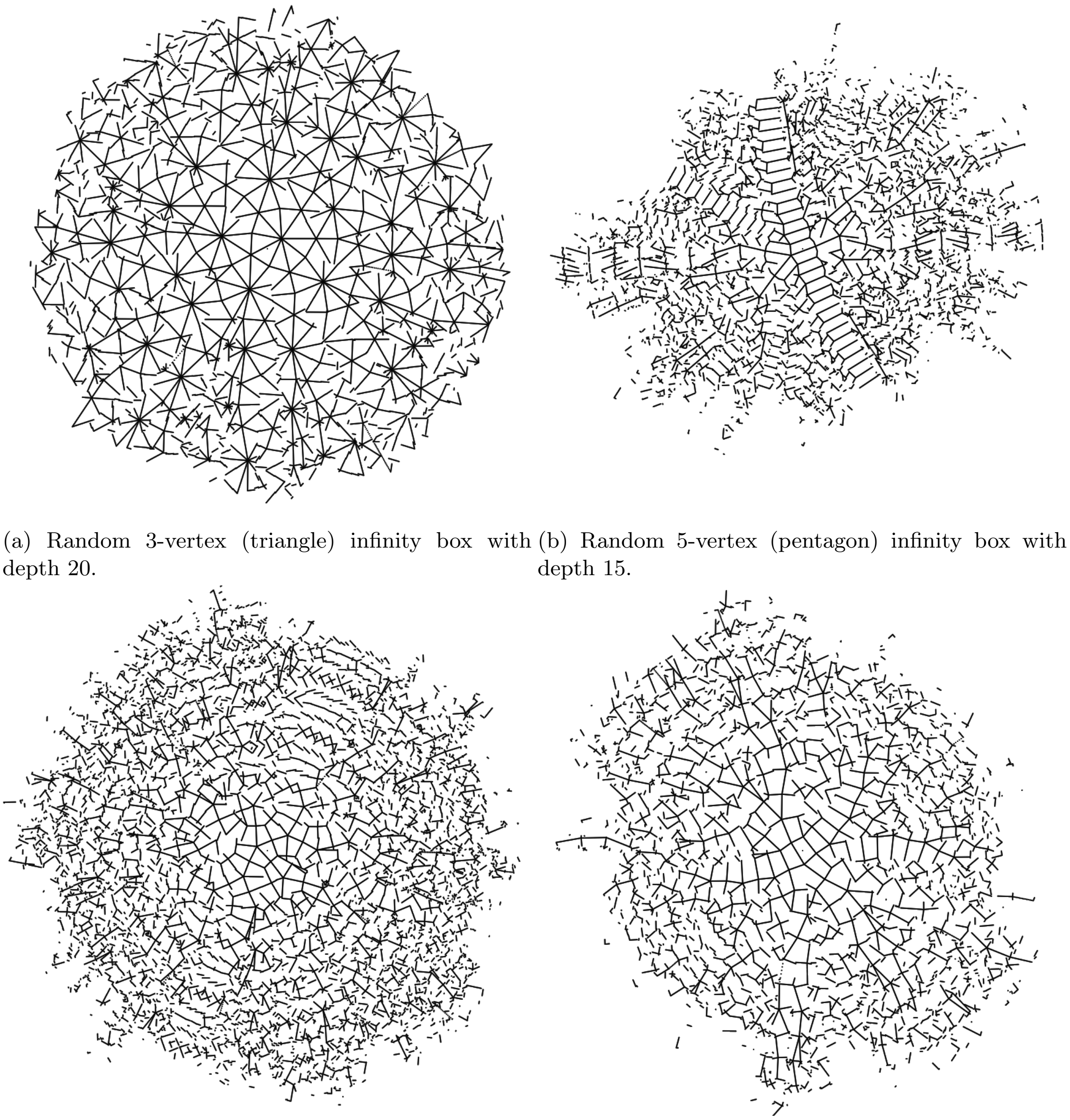
(d) 9-Regular infinity box with $\lambda = -2.76$ aligned by hand to higher-order image points.

Figure 5.5: Image-aligned pentagon, heptagon and 9-regular infinity boxes compared with the perfect circle infinity box.

This figure illustrates how the pentagon infinity box can be made to resemble the perfect circle at $\lambda = 0$ while still have a relatively high curving parameter, $|\lambda| = 2.90$. This property is what makes the Pentagon infinity box special among other N-Regular infinity boxes.

The heptagon and 9-regular infinity boxes have been aligned by tweaking the parameter λ by hand since we do not have a direct formula for the higher-order reflections. Note that their curving parameter is slightly lower than that of the pentagon and that their infinity box features more irregularities than the pentagon.

5.2.4 Random Polygons



(a) Random 3-vertex (triangle) infinity box with depth 20.

(b) Random 5-vertex (pentagon) infinity box with depth 15.

(c) Random 4-vertex (rectangular) infinity box with depth 20.

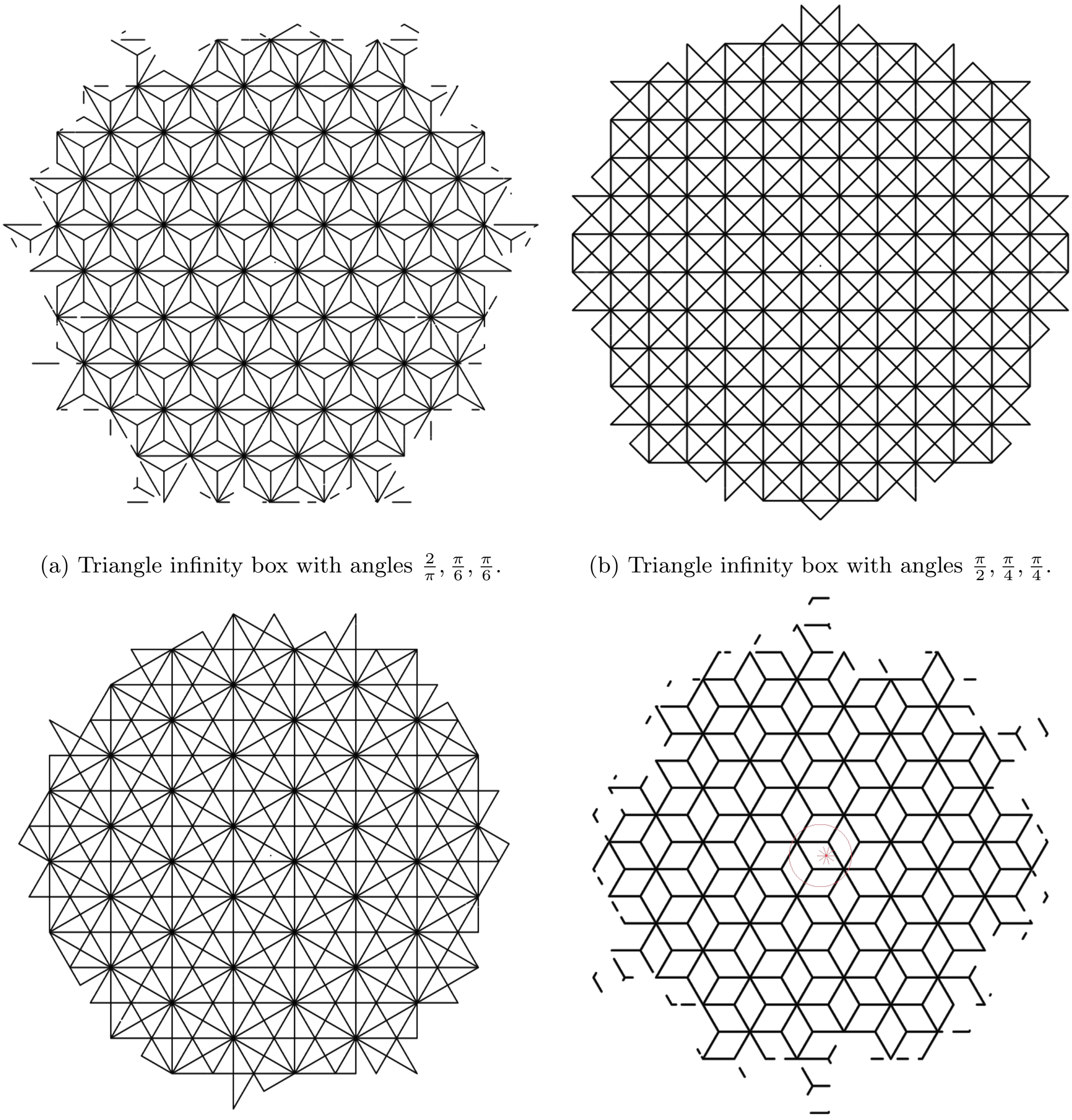
(d) Random 4-vertex (rectangular) infinity box with depth 15.

Figure 5.6: Random N-Regular infinity box simulations with varying depth.

Random polygonal infinity box create the least-regular patterns as expected. However, we can spot certain patterns in parts of each infinity box that part of a regular infinity box. Consider for example the top-right infinity box in the above figure, a repeating pattern runs all the way from the top to the bottom because two mirrors happen to nearly face each other. These infinity boxes are another example to illustrate the chaotic nature of dynamical billiard trajectories. All points that lie on the same ray cast from the origin give the collision points of the infinity box. These points characterize the trajectory of the ray and thus the trajectory in a dynamical billiard system.

The same holds for the other three infinity boxes, regularities in their shape are visible in the infinity box. In the next section we see some more mirror-tiling infinity boxes from which these random infinity boxes derive some of their patterns.

5.2.5 Special Mirror-Tiling Polygons



(a) Triangle infinity box with angles $\frac{2}{\pi}, \frac{\pi}{6}, \frac{\pi}{6}$.

(b) Triangle infinity box with angles $\frac{\pi}{2}, \frac{\pi}{4}, \frac{\pi}{4}$.

(c) Triangle infinity box with angles $\frac{\pi}{2}, \frac{\pi}{3}, \frac{\pi}{6}$.

(d) Rhombus infinity box with angles $\frac{2\pi}{3}, \frac{\pi}{2}, \frac{2\pi}{3}, \frac{\pi}{2}$.

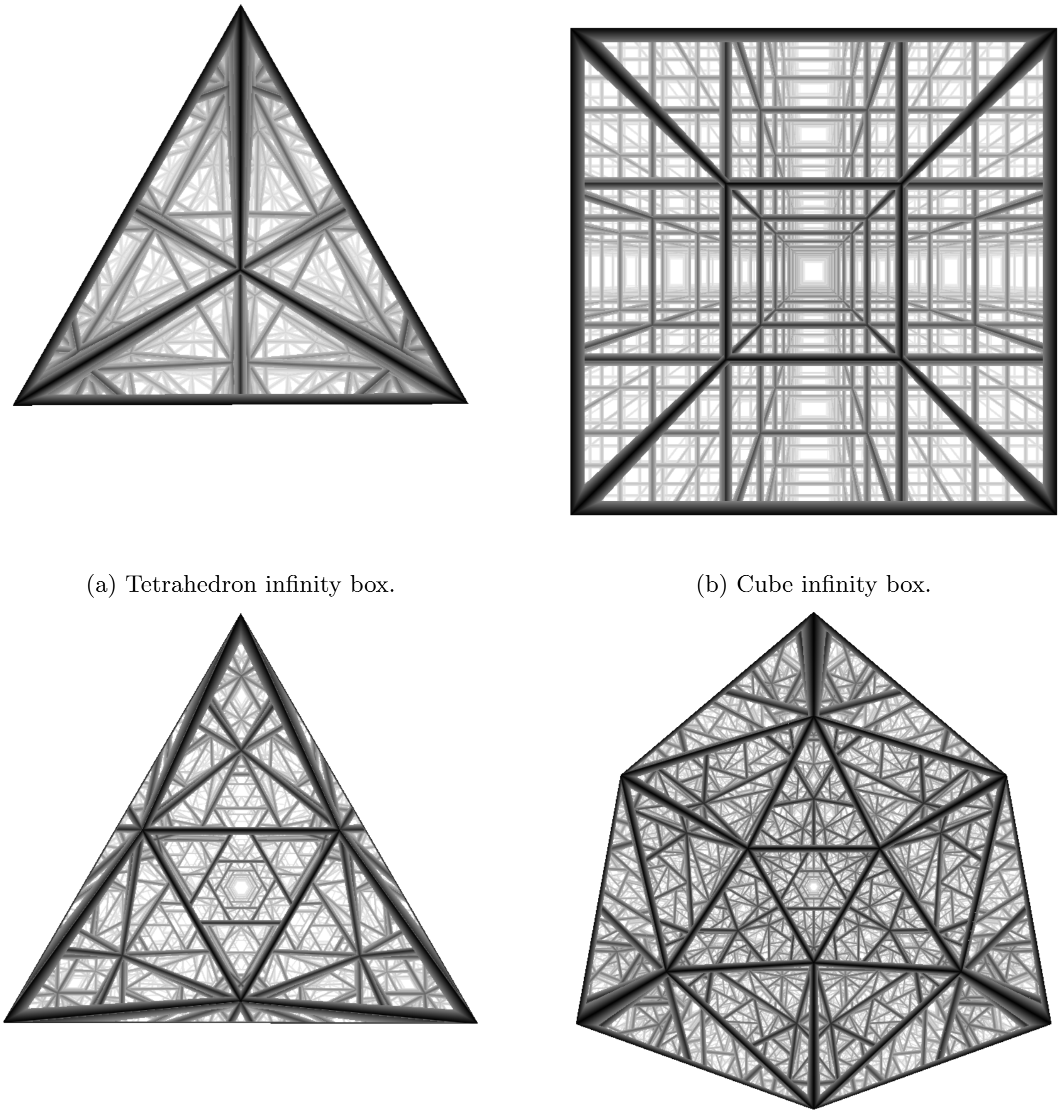
Figure 5.7: Mirror-tiling infinity boxes derived from quads and triangles at varying reflection depths. See Section 2.3 for an overview of all mirror-tiling polygons.

These infinity boxes have the special property of being mirror-tiling. They are all derived from an N-Regular infinity box that we know tiles two-dimensional space (triangle, rectangle and hexagon). In the random infinity boxes from Section 5.2.4, some of their patterns can be found as parts of the four presented here.

5.3 Three-Dimensional Infinity Boxes

For 3D infinity boxes, we will not be able to provide a single picture of the entire pattern as a whole. This is because we will be viewing these 3D infinity boxes with from a single camera perspective using either an orthogonal or perspective projection.

5.3.1 Platonic Solids



(a) Tetrahedron infinity box.

(b) Cube infinity box.

(c) Octahedron infinity box.

(d) Icosahedron infinity box.

Figure 5.8: Platonic solid infinity boxes at depth 15 rendered using a perspective projection with a 90° degrees field-of-view.

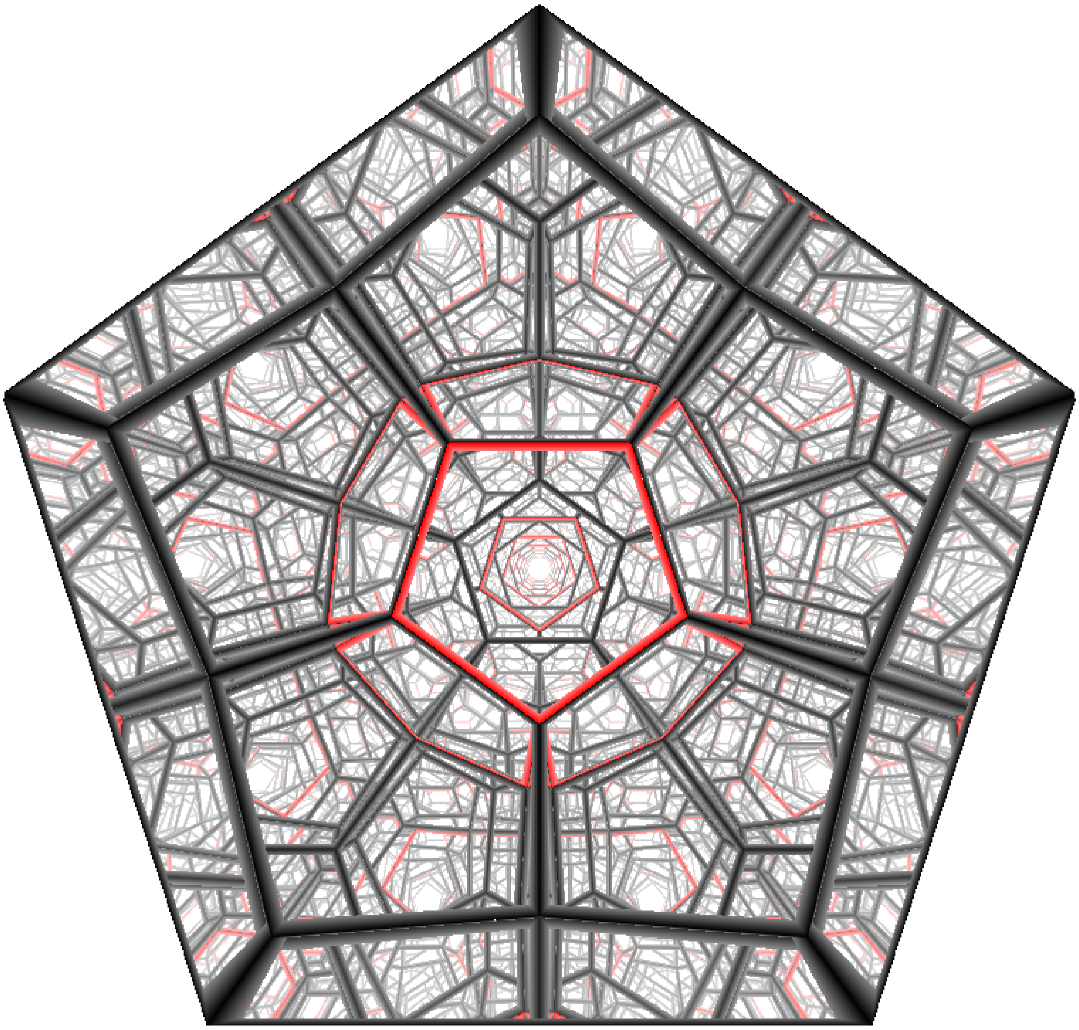


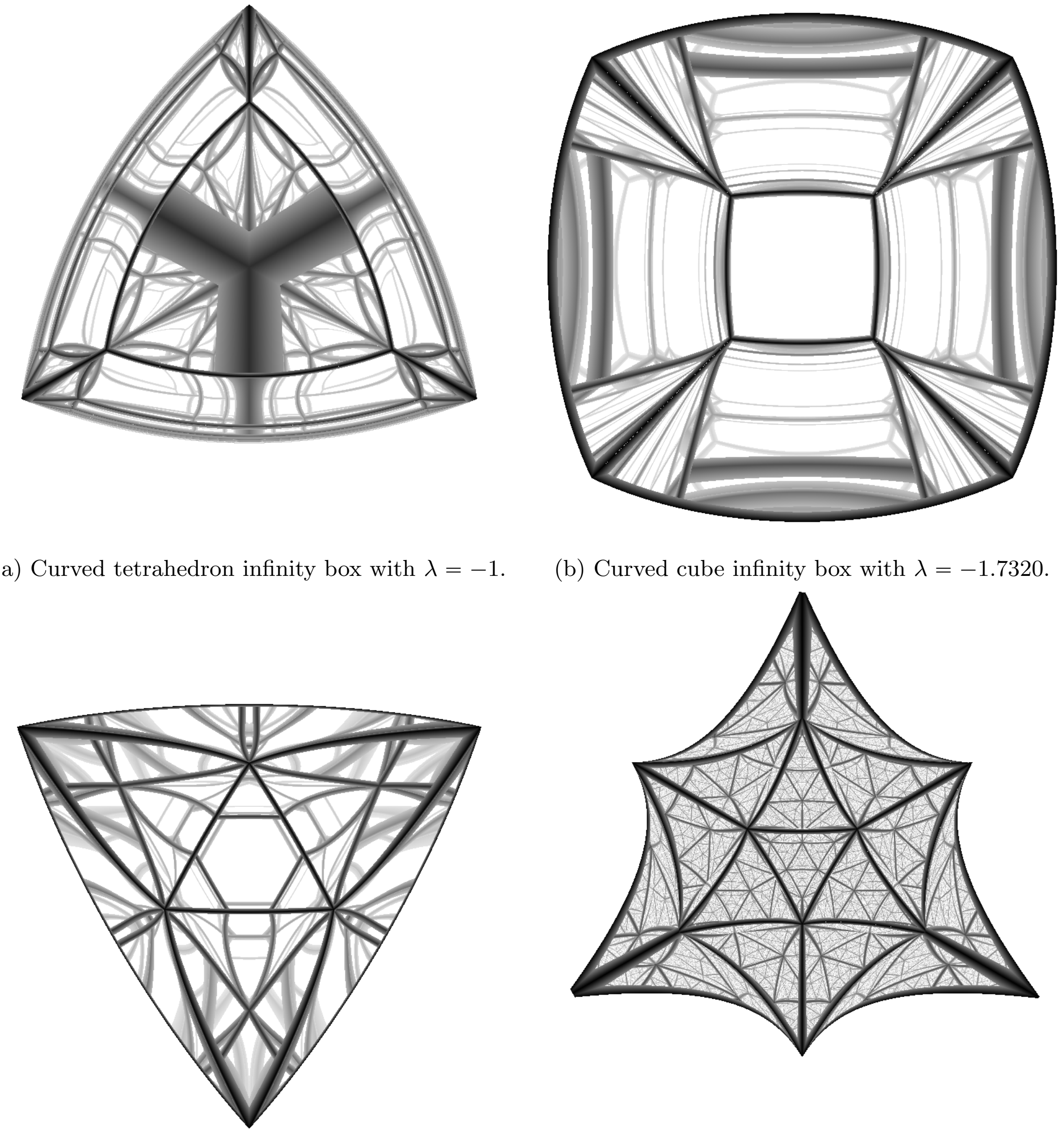
Figure 5.9: Dodecahedron infinity box at depth 15 rendered using a perspective projection with a 90° degrees field-of-view. The edges of a single face are marked in red.

Observe that the only mirror-tiling platonic solid is the cube. All shapes are viewed from a point that highlights their symmetry. If the camera we're rotated, then the resulting pictures would be more chaotic. Understanding these 3D infinity boxes is easier when controls to manipulate the camera are available. Refer to MiRai [13] to view these figures in real-time from various perspectives. We are not showing the outside edges of the infinity box to reduce clutter, these figures look directly into the interior of the infinity box.

If we turn to the dodecahedron infinity box, we can see that the reflections of the red edges nearly connect to from a dodecahedron. This means that the dodecahedron infinity box nearly has a tiling behavior for the first few reflections. We will see in Section 5.3.3 that we can make these red edges come together by curving the faces of the dodecahedron to create a better tiling illusion.

The edges are fading in the distance because the light is simulated to lose energy with each reflection. This results in a smoother picture with less clutter, since the edges at the front are better visible than the edges in the back. The spokes that we observed for 2D infinity boxes become infinitely deep holes in 3D infinity boxes. In the previous two figures, these are the white parts of the image where no edges are in the way.

5.3.2 Platonic Solids Curved To Internal Angle 120°



(a) Curved tetrahedron infinity box with $\lambda = -1$.

(b) Curved cube infinity box with $\lambda = -1.7320$.

(c) Curved octahedron infinity box with $\lambda = -4.1815$.

(d) Curved icosahedron infinity box with $\lambda = 2.3849$.

Figure 5.10: Platonic solid infinity boxes curved to an internal angle of 120° using the values from Table 3.8. The icosahedron is analogous to the heptagon in two dimensions in that it requires convex mirrors (hyperbolic). The other infinity boxes are viewed from the outside, aligned to the normal of a face.

Note that these infinity boxes change when viewed from another position, making it hard to create a single image. In each of these pictures, the infinity box is shown from a point of high symmetry, which results in a symmetric look into the infinity box. However, if the camera is slightly off, then the resulting infinity boxes are much harder to comprehend. See Figure 5.11 for the same curved cube infinity box, but viewed from a different position.

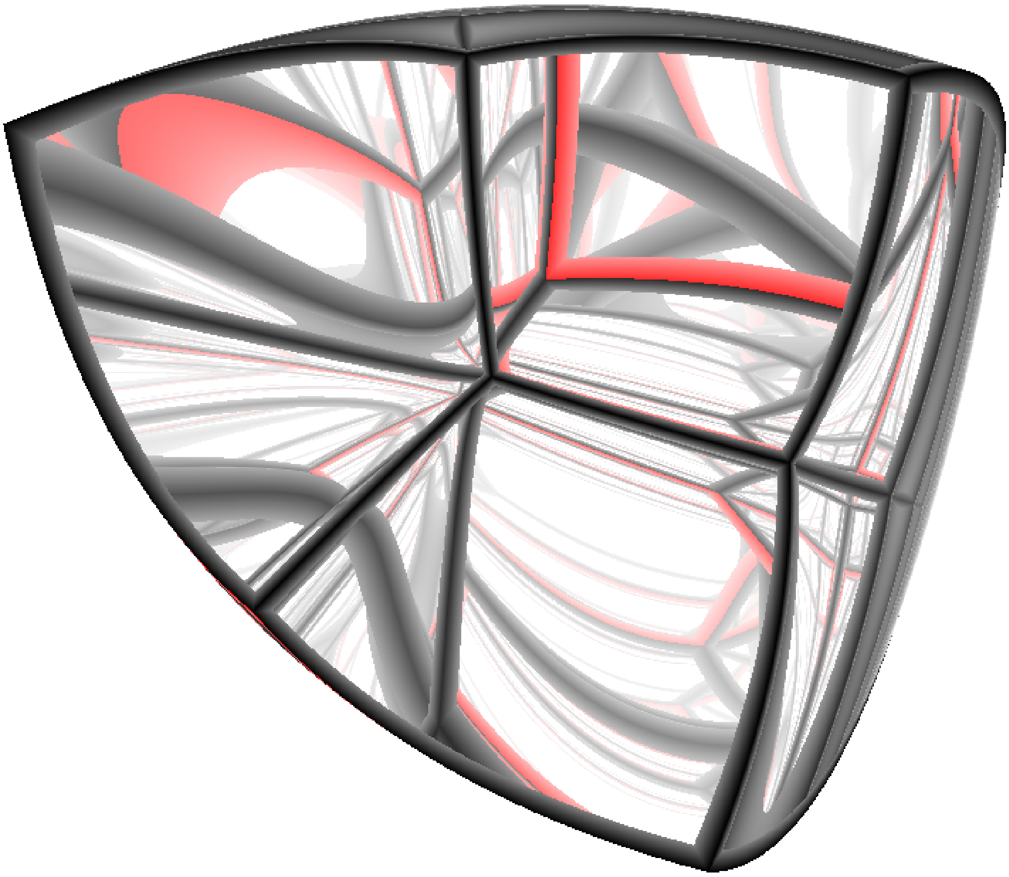


Figure 5.11: The cube infinity box curved to an internal angle of 120° with $\lambda = -1.7320$. Viewed from an arbitrary position that highlights less symmetry.

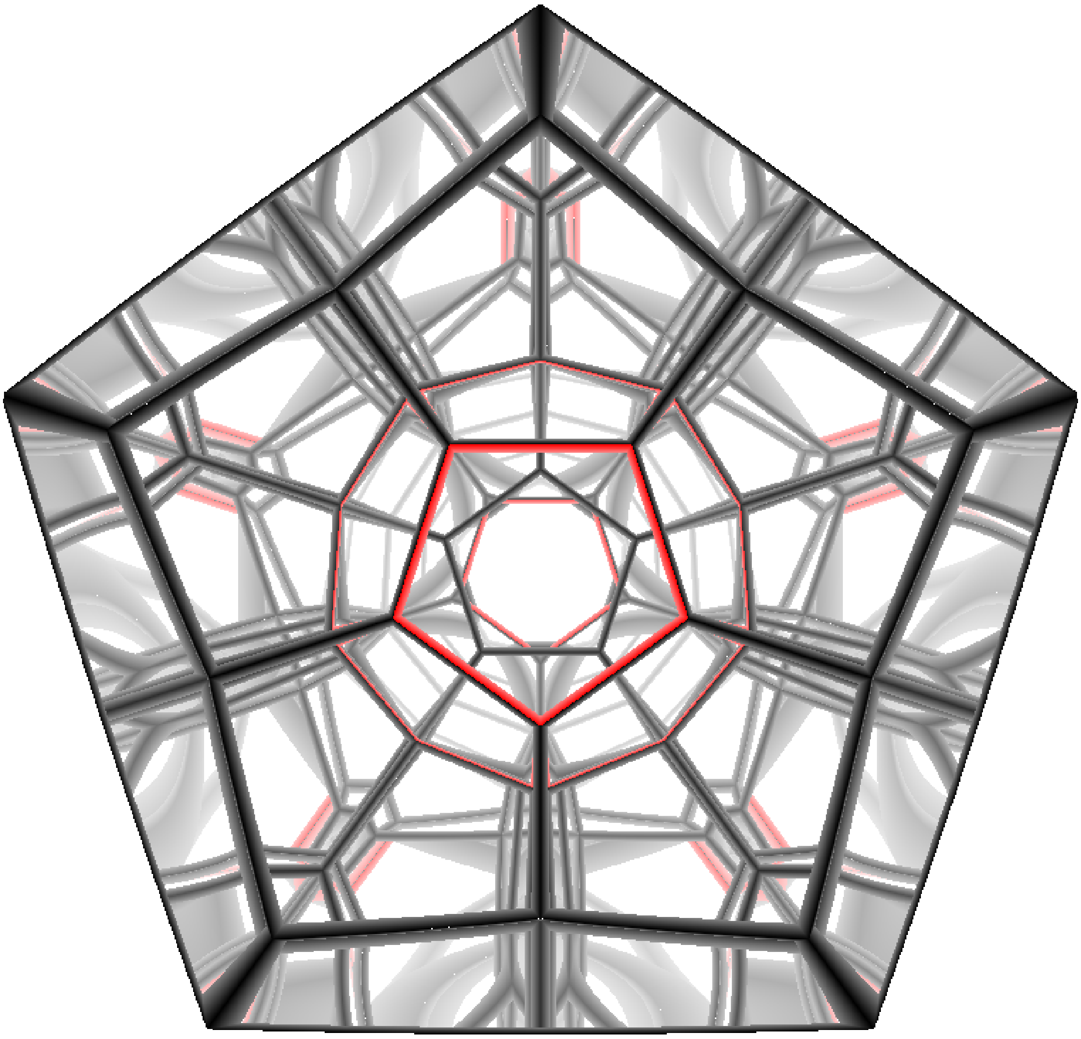
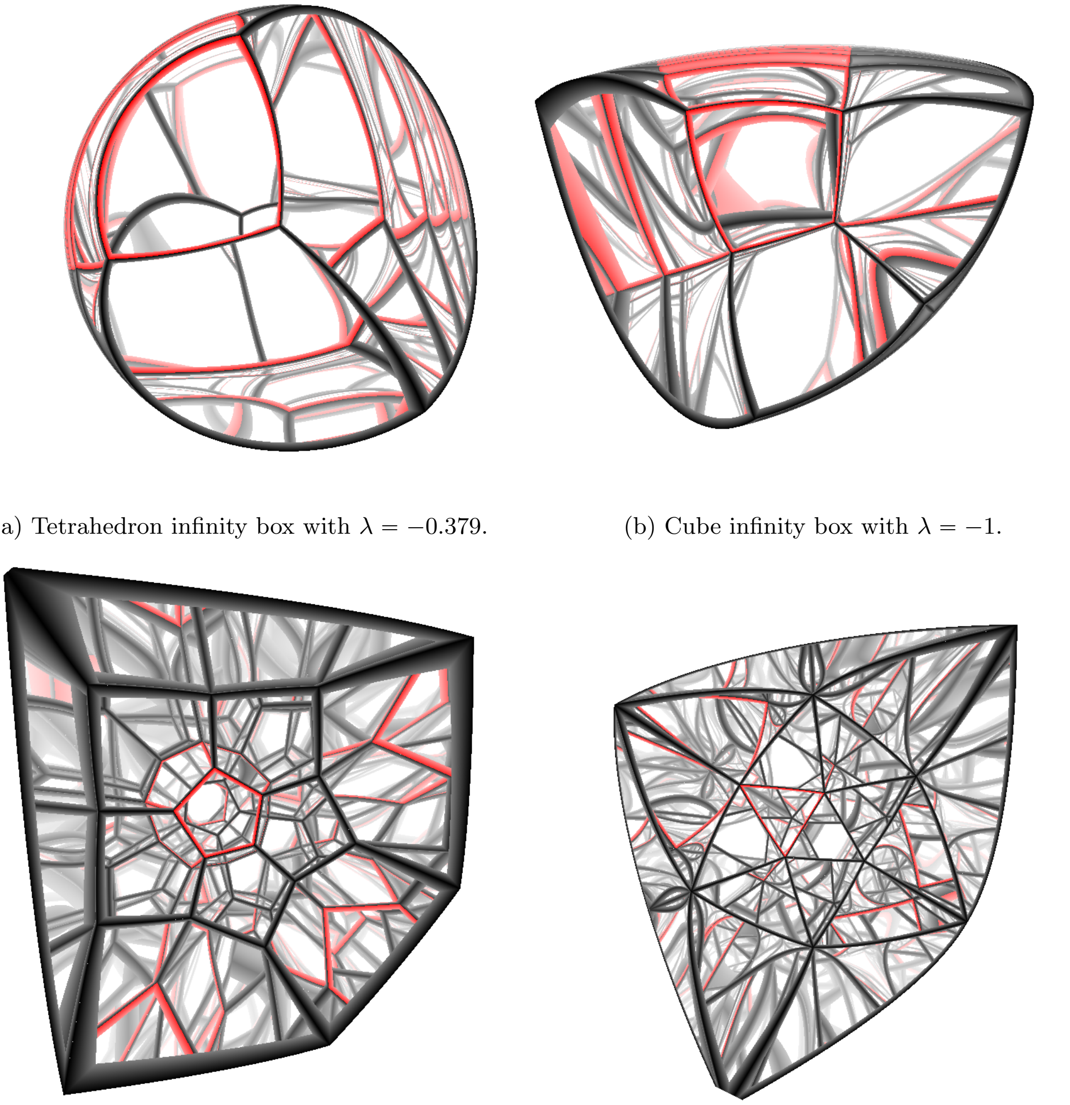


Figure 5.12: The dodecahedron infinity box curved to an internal of 120° with $\lambda = -15.6483$, as given from Table 3.8. When compared to the flat dodecahedron infinity box from Figure 5.9, the regions marked in red are better connected.

In ways that we have expected, curving the mirrors makes the pattern from the dodecahedron infinity box come together. See Figure 5.12. The red regions now nearly appear to form the front region of another dodecahedron, the beginning of a tiling pattern. We can still improve on the pattern from the dodecahedron by aligning to image points instead.

5.3.3 Platonic Solids Curved To Align Image Points



(a) Tetrahedron infinity box with $\lambda = -0.379$.

(b) Cube infinity box with $\lambda = -1$.

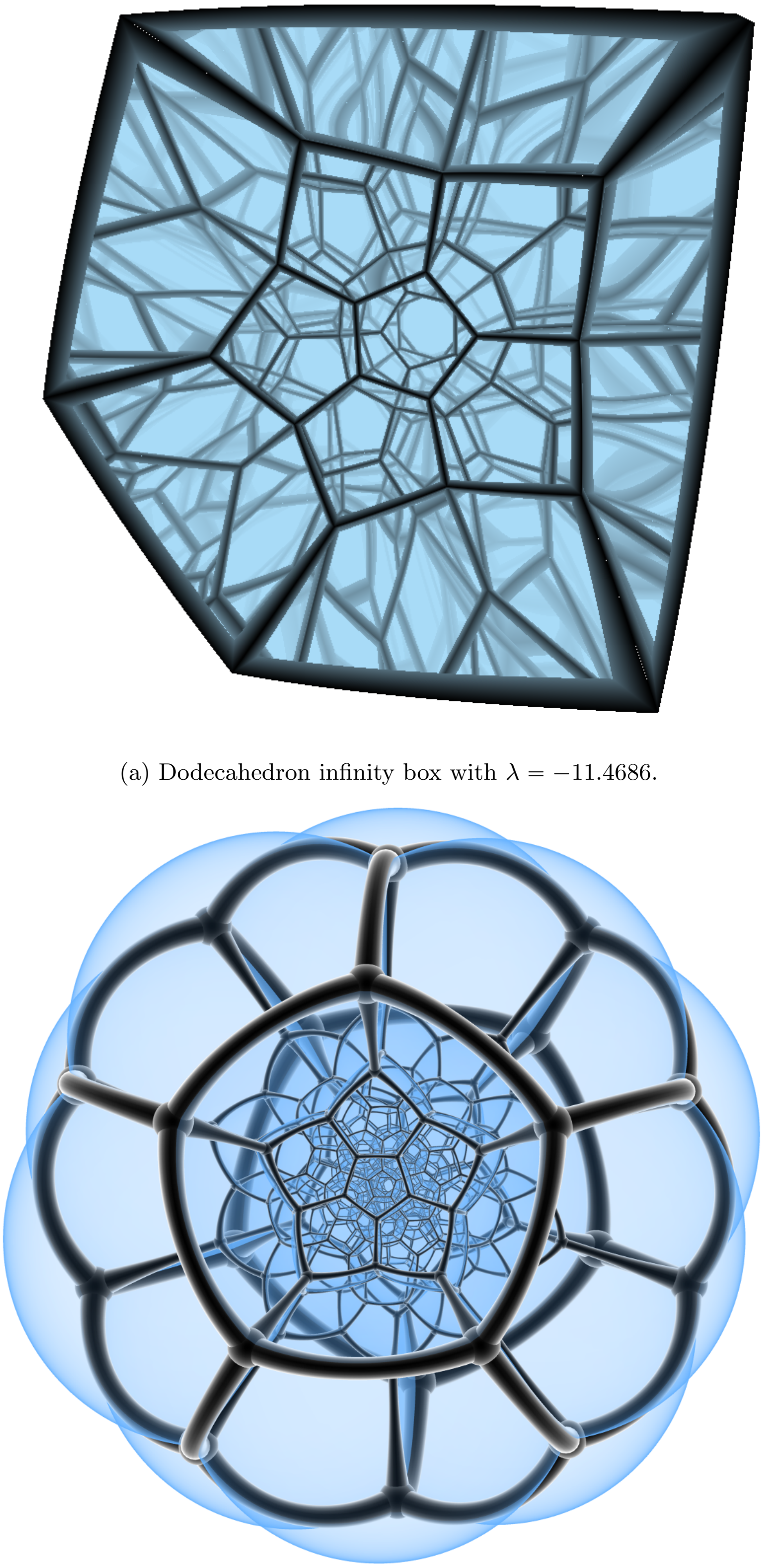
(c) Dodecahedron infinity box with $\lambda = -11.4686$.

(d) Icosahedron infinity box with $\lambda = -3.3993$.

Figure 5.13: Platonic solid infinity boxes curved to align first image points with values from Table 3.8. Viewed from the outside with red edge marks to a single face. It is visible that the dodecahedron admits the most tiling pattern compared to the others when curved to image points.

Another notable difference between the dodecahedron and the other three, is that it requires the least curvature to align its first-order image points. The tetrahedron is nearly a full sphere in this setting, which is not a surprise due to the large angle between the image points. The same goes for the cube, it requires a large curvature because its first-order image points are far apart. However, if we we're to align the cube to second-order image points, then it would perfectly tile as if it we're flat.

See Figure 5.14 for a better view of the inside of the dodecahedron and icosahedron in this setting. The dodecahedron infinity box in this setting can give an impression of the interior of the 120-Cell for the first few reflections. However, this pattern continues infinity far while the 120-Cell only consists of 120 dodecahedra.



(a) Dodecahedron infinity box with $\lambda = -11.4686$.

(b) The stereographically projected 120-Cell

Figure 5.14: The curved dodecahedron infinity box compared to the stereographically-projected 120-Cell. The curved dodecahedron infinity box can be seen as an approximation of the interior of the 120-Cell near the origin.

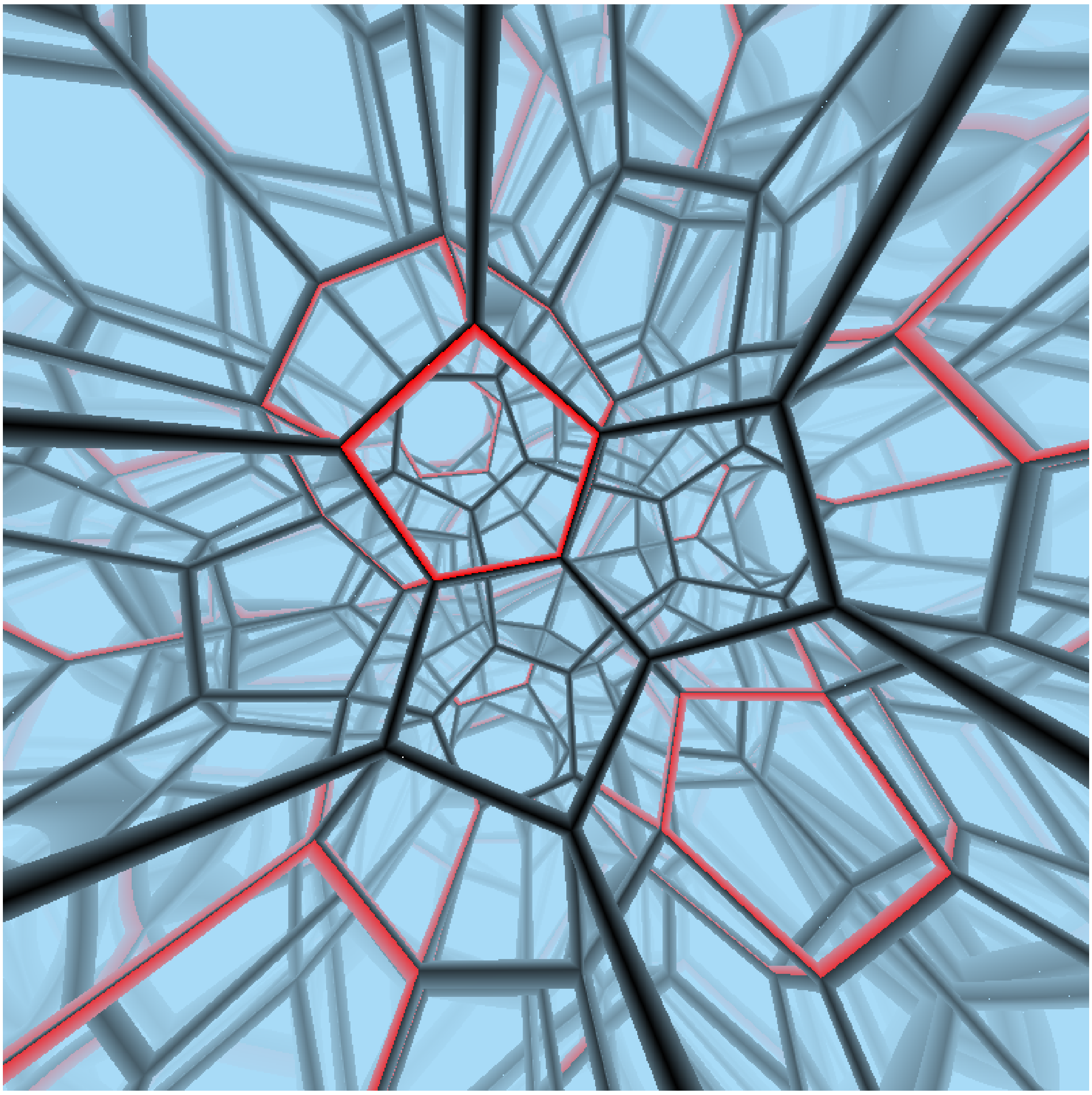
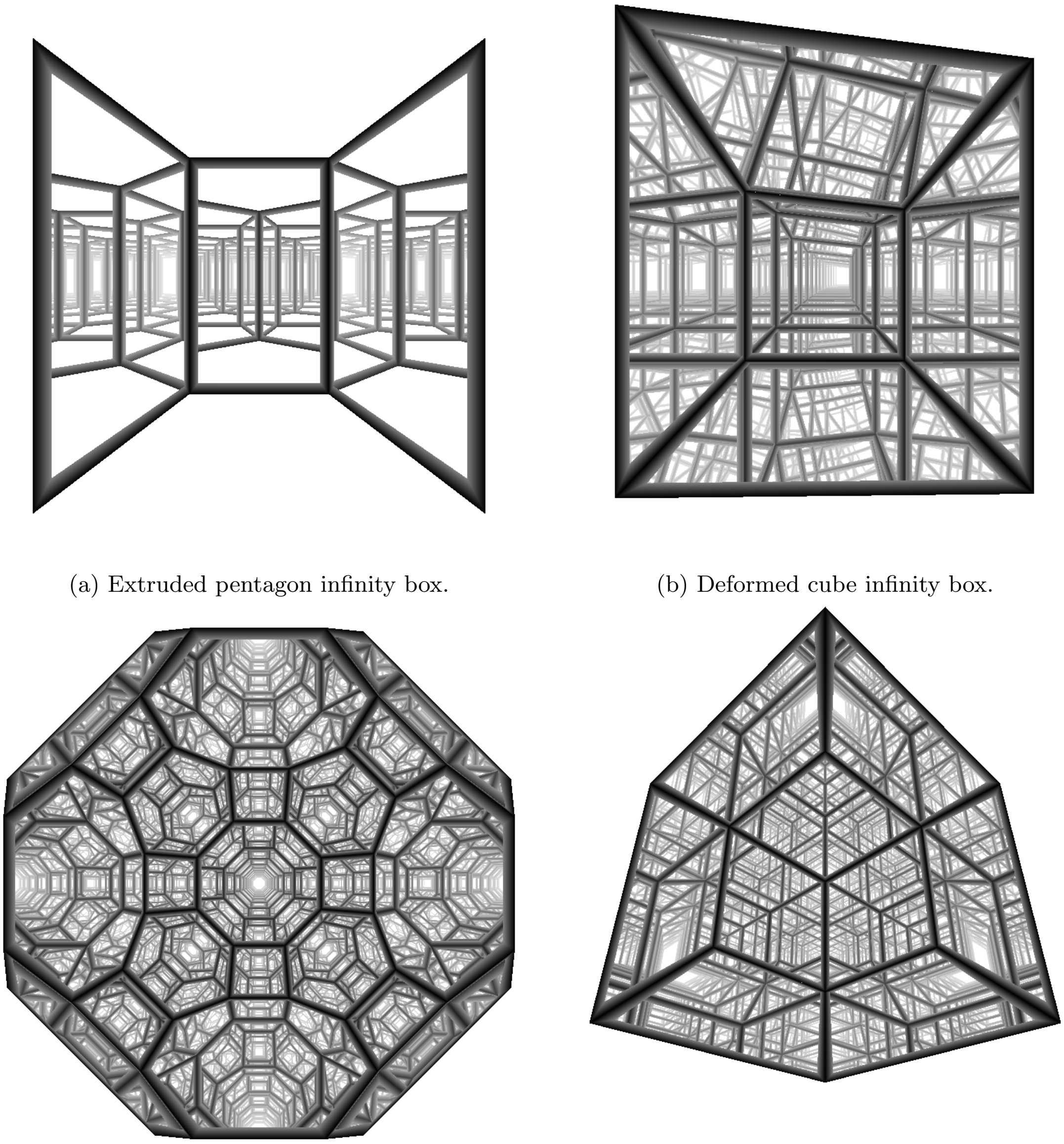


Figure 5.15: Dodecahedron infinity box with $\lambda = -11.4686$.

5.3.4 Polyhedra And Extruded Polygons



(a) Extruded pentagon infinity box.

(b) Deformed cube infinity box.

(c) Truncated cuboctahedron infinity box.

(d) Rhombic dodecahedron infinity box.

Figure 5.16: Various other flat infinity boxes which show mixed levels of symmetry and disorder. The pentagon is extruded to a 3D infinity box and provides a different perspective of the patterns from Figure 5.1. The rhombic dodecahedron is one among the rare mirror tiling polyhedra.

The extruded pentagon shows that, in general, any lower dimensional infinity box can be extruded to a higher-dimensional infinity box with similar qualitative properties. Comparing the extruded pentagon with a top-down perspective, it is clear that the 3D perspective is less insightful in this case. The deformed cube is an example of how slight perturbations in the vertices of the infinity box accumulate in the distance to create disorder.

This concludes our numerical analyses of infinity boxes in 2D and 3D.

6 Numerical Methods For Infinity Boxes

6.1 Introduction

One of the major parts of this paper involved writing a computer program to visualise infinity boxes in $\mathbb{R}^2$ and $\mathbb{R}^3$. While the mathematics of infinity boxes is the same between $\mathbb{R}^2$ and $\mathbb{R}^3$, the techniques to simulate with a computer are very different.

To visualise an infinity box, we need to cast many rays from the perspective of the camera. A high resolution is needed to make sense of the resulting image. A naive approach to ray tracing on the CPU in Python would result in code that takes over a minute to run for around 100000 rays. While this would be fine for static images, it makes it drastically inconvenient to modify parameters and understand the transition from flat to curved infinity boxes. A better way to utilize modern computers is by means of GPU programming.

The GPU or video-card is used to process pixels on a screen in a highly parallel fashion, it is what drives the responsive screens of modern devices. The power of the GPU lies in its high parallel processing capabilities thanks to its many processing cores. Since tracing a ray does not depend on other rays, we can trace our rays using the GPU in a highly parallel fashion. This is what allowed our program MiRai to achieve real-time visual change when a parameter is changed.

We used the C++ programming language to create MiRai. The code library that we used to interact with the GPU is called Raylib [10]. In the next two sections we will explain the way we rendered infinity boxes in 2D and 3D using these tools.

6.2 Rendering 2D Infinity Boxes

6.2.1 Overview

When rendering infinity boxes in 2D, we do not show a first-person image from the view of a 1D camera, contrary to 3D. We opted to visualise the infinity boxes in 2D from a top-down view. This top-down view shows all the world image points where a ray collided with a mirror. The resulting image reveals patterns from a perspective that can not be achieved in 3D, since we can not view 3D space from a top-down view in 4D.

For this reason, the rendering process is different from 3D, because we must draw dots on varying positions on the screen as opposed to tracing rays from the eye and giving a color for every pixel on the lens as in 3D. We outline the 2D rendering process below.

1. Load a flat circular infinity box as a sequence of vertices $\vec{v}_i \in \mathbb{R}^2, i = 0, \ldots, M$ that form a closed polygon. Let $b \in \mathbb{N}$ be the maximum number of reflections and $N \in \mathbb{N}$ be the total number of rays to cast.
2. **GPU Compute Pass**. For each ray $i = 0, \ldots, N-1$, we compute the distance travelled each time it collides with a mirror surface of the infinity box. For each of these collisions $j = 0, \ldots b-1$ we will store the distance travelled up until that point, where b is the maximum number of reflections. Therefore we end up with a distance matrix D with entries D_{ij} that contains for each ray i the distance to the j-th collision. The ray is given radially by
$$\vec{d}_i = \begin{pmatrix} \cos\left(\frac{2\pi i}{N}\right) \\ \sin\left(\frac{2\pi i}{N}\right) \end{pmatrix}, \quad i = 0, \ldots, N-1. \tag{6.1}$$
Where N is the total number of rays. This means that each ray i will have b values stored on the GPU indexed by j.
3. **GPU Render Pass**. Draw $N \cdot b$ points $(i, j) \in \{0, \ldots, N-1\} \times \{0, \ldots, b-1\}$ at the location
$$\vec{p}_{ij} = D_{ij}\vec{d}_i. \tag{6.2}$$
Therefore, for each of the collision distances we computed in the GPU Compute Pass, we draw a point along the ray at that distance. The justification that $\vec{p}_{ij}$ is actually the world image point obtained after reflecting the last collision in all previous mirrors is given by the Unfolding Theorem from Section 3.1.
4. Repeat Steps 2 and 3 every time a parameter is changed.

In our implementation of this scheme in MiRai, we allow N up to $500,000$ and b up to 32. Therefore the total amount of distances that is computed can be up $N \cdot b = 16 \cdot 10^6$. Each of these distances is a floating point number and requires four bytes of GPU memory. Therefore the memory needed to store all these distances is 64 Megabytes. All modern (after 2010) graphics cards guarantee GPU memory of at least 128 Megabytes, therefore memory is not a concern when running MiRai.

These four steps outline the broad outline of the 2D rendering process. However, the practical implementation of Step 2 requires more depth. This is the topic of the next section.

6.2.2 Reflection Computations

In Step 2 of the rendering process for 2D infinity boxes we must compute the distances d_{ij} containing for the i-th ray the j-th distance travelled. The exact step-by-step process to obtaining these values is given below.

1. Compute the initial ray direction $\vec{d}_i$ as given by Equation 6.1.
2. Initialize all distances for this ray to zero, $D_{ij} = 0$ for $j = 0, \ldots, b$. Initialize the total travelled distance s of this ray to zero, $s_i = 0$.
3. **Collision Testing**. For each line segment $(\vec{u}_k, \vec{v}_k), k = 0, \ldots, M-1$ with normal $\vec{n}_k$ **or** circle with center $\lambda\vec{n}_k$ and radius $\|\vec{u}_k - \lambda\vec{n}_k\|$ (depending on if we use flat or curved infinity boxes),

compute the collision step $\Phi_k > 0$. Then, perform the following checks to validate if the collision is part of the infinity box.

(a) **Collision Point Validation**. Check if the collision point $\vec{p} = \vec{o} + \Phi_k \vec{d}$ or its projection on the line segment $(\vec{u}, \vec{v})$ actually lies between the points $\vec{u}$ and $\vec{v}$. If not, ignore this collision. We are essentially applying Theorem 12 to validate the collision by checking if the projection $p^{-1}\vec{x} \in \text{Hull}(P_k)$ for the collision point $\vec{x}$, line P_k and projection constant $p^{-1} > 0$.

(b) **Minimum Step Validation**. Check if the collision distance Φ_k is greater than a minimum step size $\varepsilon > 0$. This check is required to avoid rays colliding when $\Phi_k = 0$, which are not valid collisions. If the step Φ_k is valid, add it to the set of valid steps V.

4. Take the minimum over all valid collision distances $\widehat{\Phi} = \min_{k \in V}\{\Phi_k\}$ to get the first surface that the ray collides with. If $\#V = 0$, stop computing for this ray, this can happen if the ray origin lies outside the infinity box and the ray misses all surfaces.

5. Let $\widehat{k}$ be the mirror that caused the collision at collision distance $\widehat{\Phi}$. Reflect the ray in this mirror at position $\vec{o} + \widehat{\Phi}\vec{d}$. Finally, add $\widehat{\Phi}$ to the total distance travelled s_i and set $D_{ij} = s_i$. Increment j by one.

6. If $j < b$, repeat from Step 3, else stop and continue with the next ray.

The tests for the validity of a collision point are important to avoid invalid collision points. In the implementation of MiRai, ε is generally set to 10^{-3}. Higher values of ε will ignore collisions that occur at close distances, which can result in no collisions at all. The origin point $\vec{o}$ and the curving parameter λ can be configured by the user.

This concludes the explanation of the 2D infinity box renderer in MiRai. The exact code can be found on Github [13].

6.3 Rendering 3D Infinity Boxes

6.3.1 Overview

The process for rendering infinity boxes in 3D is easier, because we directly show the view from the perspective of the camera. For each pixel on the screen, we trace a ray as if it came from the camera. If the ray hits an object in its way that is not reflective, we assign the pixel a color. These steps are outlined below.

1. Load a 3D model of an infinity boxes as a list of polygonal mirrors P_k with a normal $\vec{n}_k$. Since 3D models consist of triangles, we must perform additional processing on each model to turn co-planar and connected triangles into polygons, which we can then use as mirrors.

2. **GPU Camera Ray Computing**. For each pixel on the screen $(u, v) \in [-1, 1] \times [-1, 1]$ (in normalized screen coordinates) we compute the corresponding ray direction and origin by un-projecting the screen coordinates to world space. This un-projecting is done by inverting the screen coordinates with the projection × transform matrix $P \times M$.

$$\vec{o}_{h,uv} = (PM)^{-1}\begin{pmatrix} u \\ v \\ -1 \\ 1 \end{pmatrix}, \quad \vec{p}_{h,uv} = (PM)^{-1}\begin{pmatrix} u \\ v \\ 1 \\ 1 \end{pmatrix}. \tag{6.3}$$

In these equations, $\vec{o_h}$ and $\vec{d_h}$ are the *homogeneous* coordinates in the cameras view frustum. The first three coordinates of the vector represent the location inside the viewing frustum, which can be thought of as a deformed cube. This is due to the way projection and transformation matrices work. The fourth component must be set to 1 and is used in translation and projection.

To get the ray starting position and direction, we must divide these coordinates by their w coordinate, such that it becomes 1.

$$\vec{o}_{uv} = \frac{1}{w_{u_{h,uv}}}\begin{pmatrix} x_{u_{h,uv}} \\ y_{u_{h,uv}} \\ z_{u_{h,uv}} \end{pmatrix}, \quad \vec{p}_{uv} = \frac{1}{w_{p_{h,uv}}}\begin{pmatrix} x_{p_{h,uv}} \\ y_{p_{h,uv}} \\ z_{p_{h,uv}} \end{pmatrix}. \tag{6.4}$$

These are the world-space ray origin $\vec{o}$ and ray target point $\vec{p}$. The full ray is given by

$$\vec{r}_{uv}(t) = \vec{o}_{uv} + t\frac{\vec{p}_{uv} - \vec{o}_{uv}}{\|\vec{p}_{uv} - \vec{o}_{uv}\|}. \tag{6.5}$$

This is the ray that is traced to compute the color value for the pixel at screen location (u, v) in normalized screen coordinates. For a detailed explanation of working with projection matrices for computers and graphics programming interfaces, See [7], [2] and [9]. The notion of homogeneous coordinates is well-established in computer graphics, these three citations provide information on this topic.

3. **GPU Compute Pass**. After we have computed the ray in world-space, we can start by simulating its movement through the infinity box. For each of these rays we compute collisions and reflections up until $b \in \mathbb{N}$ reflections. When we collide with the edge of of a mirror, we assign the pixel a color value. If there have been no collisions, we set the color to match the background.

4. Repeat Steps 2 and 3 every time a parameter is changed.

In MiRai, the resolution of the screen is set to 800 × 800 by default. This is allows more than enough detail to view a high amount of reflections. This means that MiRai will always cast $800^2 = 640,000$ rays in 3D, one for each pixel on the screen. Since the frame-rate can drop below 60 frames per second when the mirror count increases, the resolution is adjusted dynamically in MiRai to increase performance. This dynamic resolution feature can be disabled. The 3D renderer on the other hand does not require storing of the collision distances D_{ij} as in the 2D case, making it slightly more efficient in memory usage.

6.3.2 Reflection Computations

The reflection computations follow the same rules and collision validations as in 2D. See Section 6.2.2. The only difference is that in order the check if a curved collision is valid , we now project back on the polygonal mirror instead of a line. If the point lies within the bounds of the polygon, then it is a valid collision. This is the three-dimensional variant of applying Theorem 12 that provides a condition if a collision point lies on the curved infinity box $S_\lambda(F)$.

In addition, after having found the next collision step $\hat{\Phi}_k$ with mirror k, we test if this collision point lies on any of the edges of the mirror. If so, we assign a color value at the pixel of that ray. This is what makes the 3D computations slightly less efficient than the 2D computations, since they require an additional edge collision test for every mirror.

6.4 Limitations

MiRai excels at visualising infinity boxes in 2D and 3D, however not all types of infinity boxes are supported. Both the 2D and 3D renderer are optimised for flat spherical infinity boxes. This means that it is not possible to curve non-spherical flat infinity boxes in MiRai. In addition, the screen resolution is fixed at 800×800, but this can be adjusted.

Both the 2D and 3D renderer can be extended to support arbitrarily shaped infinity boxes, though these can not be curved. This can open up research in non-convex infinity boxes, potentially with gaps, nooks and corners. However, for the purposes of this paper and simplicity, the current version of MiRai has not included these extensions.

7 Conclusion

In this paper we have applied infinity boxes to generate mirror-tiling patterns in 2D and 3D. We have formally defined infinity boxes and rays, and with these definitions, computed several geometrical cases related to these mirror-tiling patterns. In particular, we computed the curvature of the mirrors in 2D and 3D for which either internal angles become a pre-determined angle, or for which the first-order image points coincide.

We have built this theory in an effort to create numerical routines to simulate and visualise infinity boxes in 2D and 3D. The result is a program called MiRai that we used for (nearly) all our simulations and visualisations of infinity boxes. This program is available on the internet [13]. These numerical simulations of infinity boxes revealed the various patterns that hide in infinity boxes of regular polygons, random polygons, platonic solids and other polyhedra. We have used our analytical results to compute the exact curvatures of the mirrors for which the mirror patterns from non-tiling infinity boxes become more ordered. This has resulted in particularly pleasing images for the infinity boxes of the pentagon and the dodecahedron.

From the dodecahedron infinity box we concluded that a reasonable approximation of the interior of the 120-Cell can be obtained from a special curved dodecahedron infinity box. In general, we conclude that curved infinity boxes are able to approximate tilings in curved and hyperbolic space objects with dimension one higher than the infinity box.

Finally, we have connected infinity boxes to problems in dynamical billiards theory, which by themselves connect to many other fields of mathematics. We concluded that infinity boxes are visual solutions to dynamical billiard trajectories, and that they could be potentially used to shed more light on the chaotic patterns in these dynamical billiards. This makes the study of infinity boxes an exciting field that combines visual aesthetic with intricate patterns that relate to higher dimensional objects.

8 Discussion

To conclude this paper, we want to give some remarks on the work we have done, fields of improvement and potential further research.

First, the theory of infinity boxes we have presented in this paper apply only to a special class of infinity boxes that are derived from polytopes with vertices on the unit sphere (unit polygons, unit polyhedrons etc). These objects we're our primary object of interest because they are convenient to work with and relate to the regular solids, platonic solids and regular polygons. However, a potentially interesting class of infinity boxes is the class of non-convex infinity boxes. Curving these infinity boxes in the way we defined is not possible, but their patterns are well worth studying. In particular, concave infinity boxes can be used to further analyse solutions of dynamical billiards. The questions whether there exists a cyclic trajectory can be answered by analysing the infinity box at various positions.

A second point of interest is that of non-Euclidian infinity boxes. We have only been working with Euclidian infinity boxes because we are interested in mending problems that occur with non-tiling objects in Euclidian space by using curved mirrors. We are also interested in Euclidian infinity boxes because they can be built in real life. But, a whole new area of infinity boxes and their patterns can be studied be redefining the motion of a ray. If the motion of a ray through space we're replaced with the geodesic motion in spherical or hyperbolic space, then infinity boxes like the pentagon and dodecahedron would become mirror-tiling, which would be a more accurate way to generate the interior of the 120-Cell.

A third point is the fact the we did not find any high quality visualisations of the 120-Cell from the interior. It would be worth to create a tool to explore the 120-Cell from the inside under varying projections. This can be used as a way to compare the approximations with the dodecahedron infinity box.

A last point is to consider infinity boxes in four dimensions instead of two or three. These infinity boxes would generate an entire three-dimensional pattern that can be observed from various perspectives using a 3D camera. For this to work, a ray-tracer should be created to simulate how a 4D ray travels through 4D space and collides with 4D objects. Many of the results to achieve this are already provided in this paper, such as collision testing, mirroring and infinity box curving. The patterns that would arise from these 4D infinity boxes would without doubt be an order more difficult to understand, but they can reveal patterns in 4D objects from a different perspective than abstract algebra. In particular, it would be interesting to see how the infinity box of the 120-Cell would look like.

To conclude, there are many branches of infinity boxes that can be worked upon and this paper provides a solid foundation for several of them.

References

[1] Xinzhong (Tom) Chen. *Investigating the Penrose Unilluminable Room with Ray Optics.* URL: `https://www.comsol.com/blogs/investigating-the-penrose-unilluminable-room-with-ray-optics`.

[2] M. Baker D. Hearn. "Computer Graphics. C Version." In: (1997).

[3] How Do You? DIY. *Make an EASY Infinity Mirror Cube — NO 3D Printing and NO Programming.* URL: `https://www.youtube.com/watch?v=kKO4VUDaLxw`.

[4] Heinrich Dörrie. *100 Great Problems Of Elementary Mathematics (Dover).* 1965.

[5] Gurveer2837890. URL: `https://en.wikipedia.org/wiki/Illumination_problem#/media/File:Original-room.jpg`.

[6] Glenn Harris. "Polygonal Billiards". In: *Southern Illinois University Edwardsville* (2007).

[7] Joseph Paciorek Ingrid Carlbom. "Planar Geometric Projections and Viewing Transformations". In: (1978).

[8] Howard Masur. "Closed trajectories for quadratic differentials with an application to billiards". In: *Duke Mathematical Journal* (1986).

[9] Tom; David Blythe McReynolds. "Advanced graphics programming using openGL". In: (2005).

[10] Ramon Santamaria. *Raylib. A simple and easy-to-use library to enjoy videogames programming.* URL: `https://www.raylib.com/`.

[11] John Stillwell. "The Story of the 120-Cell". In: (2001).

[12] George Tokarsky. "An Impossible Pool Shot?" In: *Philadelphia, PA: Society for Industrial and Applied Mathematics: 107–109* ().

[13] Timothy van der Valk. *MiRai Infinity Box Renderer.* URL: `https://github.com/arceryz/mirai`.

[14] Jeff Weeks. *KaleidoPaint.* URL: `https://www.geometrygames.org/KaleidoPaint/index.html`.