Ray-marching Thurston geometries

Rémi Coulon, Elisabetta Matsumoto, Henry Segerman, Steve Trettel

RAY-MARCHING THURSTON GEOMETRIES

RÉMI COULON, ELISABETTA A. MATSUMOTO,
HENRY SEGERMAN, AND STEVE J. TRETTEL

Abstract. We describe algorithms that produce accurate real-time interactive in-space views of the eight Thurston geometries using ray-marching. We give a theoretical framework for our algorithms, independent of the geometry involved. In addition to scenes within a geometry X, we also consider scenes within quotient manifolds and orbifolds X/Γ. We adapt the Phong lighting model to non-euclidean geometries. The most difficult part of this is the calculation of light intensity, which relates to the area density of geodesic spheres. We also give extensive practical details for each geometry.

Contents

Date: October 29, 2020.

1. Introduction

In this paper we describe a project we initiated at the *Illustrating Mathematics* semester program at ICERM in Fall 2019. The goal of this project is to implement real-time simulations of the eight Thurston geometries in the *in-space view* – that is, viewed from the perspective of an observer inside of each space, where light rays travel along geodesics. See Figure 1.1. We have collected many of our simulations and videos of them at the website `http://www.3-dimensional.space`.

These simulations may be experienced with an ordinary keyboard and screen interface, and in some cases in virtual reality. We expect that these simulations will be useful in outreach, teaching, and research. Seeing and moving within a space gives a visceral experience of the geometry, often engendering understanding that is hard or impossible

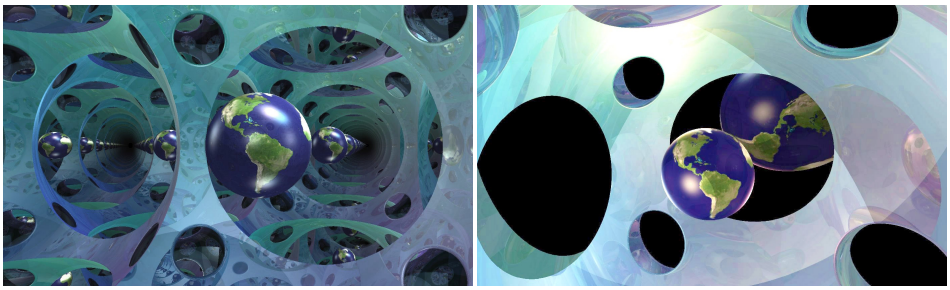
to obtain from "book learning" alone. Recent research on embodied understanding [LTWJ16, JGMR17, GPE17] addresses these advantages.

The code for our simulations is available online [CMST20c]. We hope that other researcher will be able to use and extend our work to visualize objects of interest in the Thurston geometries and beyond. In two previous expository papers, we described some surprising features of the Nil [CMST20a] and Sol [CMST20b] geometries using images from our simulations.

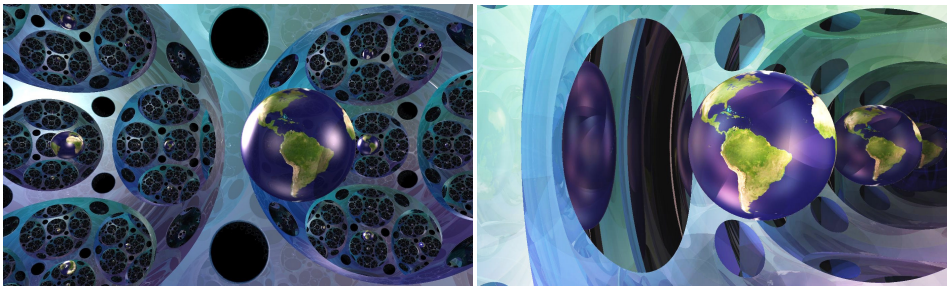
1.1. **Thurston's eight geometries.** The expansion of geometry beyond euclidean n-space traces its origins to the 19^{th} century discovery of hyperbolic geometry. From here, Klein made the following wide-reaching generalization. A *homogeneous geometry* is a pair (G, X) consisting of a smooth manifold X, equipped with the transitive action of a Lie group G. The manifold X defines the underlying space of the geometry, and the group G defines the collection of allowable motions. This convenient mathematical formalism turns some of our traditional geometric thinking upside down. Instead of defining euclidean geometry as $\mathbb{R}^n$ with a particular metric, we define it as $\mathbb{R}^n$ with a particular group of allowable diffeomorphisms (rotations, reflections, and translations), and derive as a consequence the existence of an invariant metric.

In dimension two, homogeneous geometries play an outsized role in mathematics, in large part due to the uniformization theorem. This implies that every two-dimensional manifold can actually be equipped with a geometric structure modeled on one of the homogeneous spaces $\mathbb{H}^2, \mathbb{E}^2$, or S^2. Because of this, one may often use geometric tools in settings without an obviously geometric nature. In the 1970s and 1980s, Thurston came to realize that a similar (but more complicated) result might hold in three dimensions. Thurston's geometrization conjecture stated that every closed three-manifold may be cut into finitely many pieces, each can be built from some homogeneous geometry. The proof of geometrization was completed by Perelman in 2003 [Per02, Per03a, Per03b] and provides a powerful tool in three-dimensional topology. This also resolved the Poincaré conjecture, which had been open for more than a century. The eight geometries required for geometrization can be defined abstractly as follows. A homogeneous space (G, X) is a *Thurston geometry* if it has the following four properties:

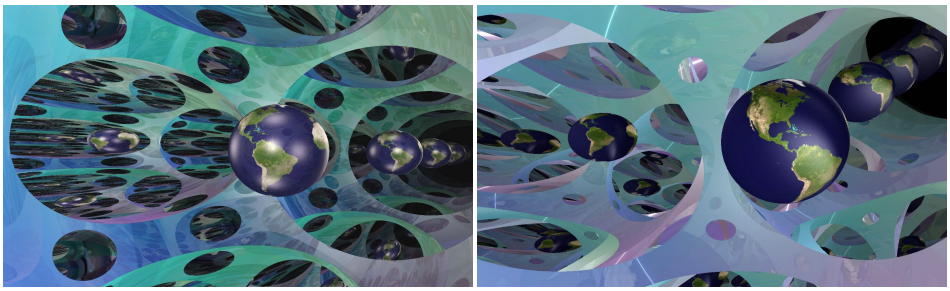
(1) X is connected and simply connected.
(2) G acts transitively on X with compact point stabilizers.
(3) G is not contained in any larger group of diffeomorphisms acting with compact stabilizers.
(4) There is at least one compact (G, X) manifold.



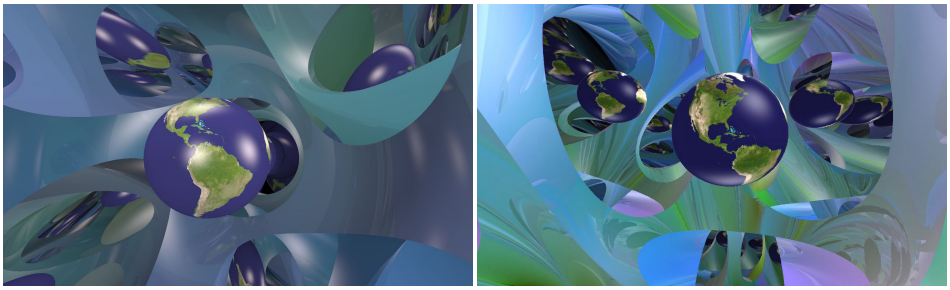
(A) $\mathbb{E}^3$ (B) S^3



(C) $\mathbb{H}^3$ (D) $S^2 \times \mathbb{E}$



(E) $\mathbb{H}^2 \times \mathbb{E}$ (F) Nil



(G) $\widetilde{\mathrm{SL}}(2,\mathbb{R})$ (H) Sol

FIGURE 1.1. Inside views of tilings within each of the eight Thurston geometries. Here we have chosen similar scenes to highlight the differences stemming from the geometries. Each scene is made from tiles as illustrated in Figure 2.1.

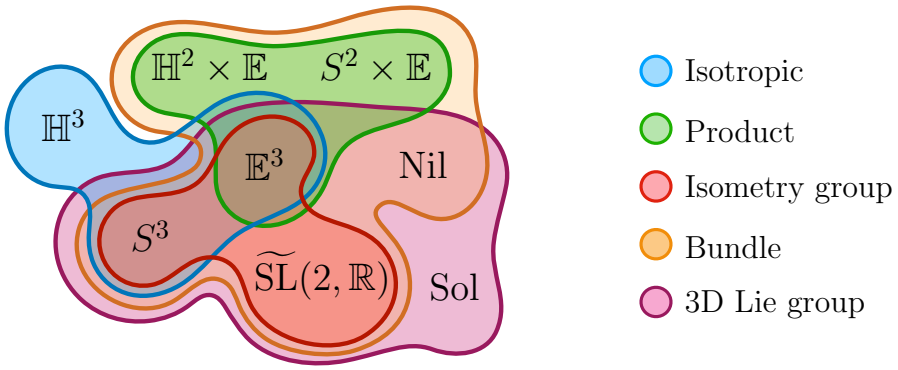


FIGURE 1.2. The Thurston Geometries, and natural families grouping geometries with similar constructions.

The first of these conditions rules out unnecessary duplicity in our classification. Every connected (G, X) geometry is covered by a simply connected universal covering geometry, so it suffices to consider these. The second condition is the group-theoretic way of requiring that X has a G-invariant riemannian metric, and the third condition is just the statement that G is actually the full isometry group. A geometry satisfying (1)–(3) is called *maximal*. The fourth condition recalls our original motivation: to study geometric structures on compact manifolds in dimension three; we need only concern ourselves with geometries which can be used to build geometric structures!

Three dimensions is small enough that all of the Thurston geometries arise from relatively simple constructions[1], growing out of either two-dimensional geometry or three dimensional Lie theory. This divides the set of Thurston geometries into a collection of overlapping families of geometries constructed by similar means. Some of these families are listed below and illustrated in Figure 1.2.

(1) **Isotropic Geometries.** A geometry (G, X) is *isotropic* if the point stabilizer contains $O(3)$. This acts transitively on the unit tangent sphere at a point. Since directions and planes are dual to each other, any G-invariant metric on X must have constant sectional curvature. Thus, this family consists of $S^3 = (O(4), S^3), \mathbb{E}^3 = (O(3) \ltimes \mathbb{R}^3, \mathbb{R}^3)$ and $\mathbb{H}^3 = (O(3,1), \mathbb{H}^3)$.
(2) **Products of Lower Dimensional Geometries.** The product of the unique one-dimensional geometry (denoted $\mathbb{E}$ in this paper) and any two-dimensional geometry gives a geometry of

[1]There are 19 maximal geometries in dimension four [Hil02], and 58 in dimension five [Gen16]. While many of these can be constructed by analogous procedures, some new phenomena also arise.

dimension three. This family consists of the three geometries $S^2 \times \mathbb{E}, \mathbb{H}^2 \times \mathbb{E}$ and $\mathbb{E}^2 \times \mathbb{E}$. The latter is not maximal: its isometry group is contained in that of $\mathbb{E}^3$.

(3) **Isometry groups of two-dimensional geometries.** Each of the two-dimensional geometries (G, X) is isotropic, so G acts transitively on the unit tangent bundle UTX. Thus we may consider the three-dimensional geometry (G, UTX), and get a maximal geometry by taking covers and extending the isometry group if necessary. This gives the geometries S^3 and $\mathbb{E}^3$, as well as the new geometry $\widetilde{\mathrm{SL}}(2, \mathbb{R})$ (built from UTS^2, $UT\mathbb{E}^2$ and $UT\mathbb{H}^2$ respectively).

(4) **Bundles over two-dimensional geometries.** Generalizing both of the previous cases, we may construct all geometries (G, X) where X has a G-invariant bundle structure over a two-dimensional geometry. This produces one new example: Nil, a line bundle over $\mathbb{E}^2$. This bundle structure has an important geometric consequence: all manifolds with these geometries are *Seifert fibered*.

(5) **Three-dimensional Lie groups.** Every three-dimensional Lie group H acts on itself freely by left translation. Starting from the homogeneous geometry (H, H), we may build a maximal geometry by taking covers and extending the group of isometries, if necessary. In addition to the unit tangent bundle geometries, this construction also recovers Nil, and produces our final geometry, Sol.

For a proof that there are only eight Thurston geometries, see for example [Pat96].

1.2. **Goals.** We have the following goals for the algorithms we use to render our in-space views.

(1) Our images must be accurate – assuming that light rays travel along geodesics, there is a correct picture of what an observer inside of a given geometry would see. Our images should accurately portray this picture.

(2) Real-time graphics algorithms must be very efficient in order to run at an acceptable frame rate. This is particularly important in virtual reality – around 90 frames per second is recommended to reduce nausea. Modern graphics cards allow for the required speed, given efficient algorithms.

(3) Our algorithms must allow for a full six degrees of freedom in the position and orientation of the camera, even when the simulated geometry may not have a natural corresponding isometry. A

user in a virtual reality headset can make such motions, and the view they see must react in a sensible way.

(4) As much as is possible, our algorithm should be independent of the geometry being simulated. The idea here is that it should be possible to change the code in a small number of places to convert between simulations of different geometries. Compartmentalizing the code in this way will make it easier to extend it to further geometries, beyond Thurston's eight.

(5) When possible, we should make our images beautiful, allowing for graphical effects including lighting, (hard and soft) shadows, reflections, fog, etc.

Some of these goals are of course in conflict. Adding features such as shadows and reflections increases the amount of work needed to be done per frame, which can reduce the frame rate. The frame rate is also dependent on the desired screen resolution. There are many trade-offs to be made between fidelity and speed.

We use the relatively new technique of *ray-marching* in our implementation. We discuss this technique and compare it with other graphics techniques in Section 2. One key feature is that the data and calculations needed to generate images for each geometry are relatively simple in comparison to other techniques, which makes it easier to write geometry independent code.

1.3. **Related work.** This project owes its existence to a long history of previous work. It is a direct descendant of the hyperbolic ray-marching program created by Nelson, Segerman, and Woodard [NSW18], which itself was inspired by previous work in $\mathbb{H}^3$ and $\mathbb{H}^2 \times \mathbb{E}$ by Hart, Hawksley, Matsumoto, and Segerman [HHMS17a, HHMS17b], all of which aim to expand upon Weeks' *Curved Spaces* [Wee] which in turn is a descendant of work by Gunn, Levy and Phillips [PG92, MLP+14] and others at the Geometry Center in the 1990's. Thurston was a driving force for much of this visualization work. He often spoke about what it would be like to be inside of a three-manifold [Thu98]. The software SnapPy [CDGW] was originally developed by Weeks to calculate the geometry on hyperbolic three-manifolds using Thurston's hyperbolic ideal triangulations. Concurrent with this project's development at ICERM in Fall 2019, Matthias Goerner implemented an inside view for hyperbolic manifolds within SnapPy, using a ray-tracing strategy.

Perhaps the earliest work concerned with rendering the inside-view of non-euclidean geometries is due to theoretical physicists predicting the appearance of black holes; this field goes back to the 1970's [Lum19].

The past few years have seen a number of independent projects building real-time simulations of inside views for the Thurston geometries, including the last three "harder" geometries. To our knowledge, Berger [Ber15, BLV15] produced the first in-space images of all eight Thurston geometries. He uses ray-tracing, with a fourth-order Runge–Kutta method for numerical integration to approximate geodesic rays.

The *HyperRogue* project [KCK19], by Kopczyński and Celińska-Kopczyńska implements all eight geometries with a triangle rasterization based strategy. They restrict the parts of the world that the viewer can see in order to avoid some issues with this approach that we identify in Section 2.3.1. For example, in certain geometries one can only see a limited distance in particular directions. They also use a fourth-order Runge–Kutta method to approximate geodesic rays, and rely in part on lookup tables for speed. Their motivation is more towards implementation for use in computer games. Here it is very useful to be able to use polygon meshes to represent the player character, enemies, and other objects in the game world. Kopczyński and Celińska-Kopczyńska [KCK20] also provide a real-time ray-tracing implementation of Nil, $\widetilde{\mathrm{SL}}(2,\mathbb{R})$ and Sol.

Novello, Da Silva, and Velho [NdSVb, NdSV20] share our interest in implementing virtual reality experiences. They also implement in-space views with a ray-tracing approach, tackling all of the Thurston geometries other than the product geometries. They use Euler's method for numerical integration to approximate geodesic rays for $\widetilde{\mathrm{SL}}(2,\mathbb{R})$ and Sol.

Other than ours, the only ray-marching approach we are aware of is due to MagmaMcFry [Mag19], who implements $\mathbb{E}^3, \mathbb{H}^3$, Nil, $\widetilde{\mathrm{SL}}(2,\mathbb{R})$, and Sol. They use a second-order Runge–Kutta method to approximate geodesic rays.

A numerical integration approach is unavoidable in some cases, for example in generic inhomogeneous geometries [NdSVa]. These approaches can also minimize the differences in the code for different geometries. However, such algorithms must take many steps along each ray to maintain accuracy, and so may be slow. This may be acceptable when the scene is "dense" – implying that few rays travel very far before hitting an object. This often happens for example, with a co-compact lattice. For scenes in which rays travel large distances we lose accuracy unless the number of steps is large, meaning that we lose rendering speed.

We instead use explicit solutions for our geodesic rays in almost all cases. This moves the problem of accuracy versus speed to the

implementation of the functions involved in the solutions. In this setting however, we have reduced the problem of understanding the long-term behavior of the geodesic flow to studying the long-term behavior of these component functions. It turns out that these functions are well-understood for the eight Thurston geometries (they are trigonometric, hyperbolic trigonometric, and Jacobi elliptic functions). Thus we can often take large steps along geodesics and achieve both accuracy and speed, even for objects that are distant from the viewer. We exploit this ability to illustrate counterintuitive, long-range behavior of geodesics in Nil and Sol [CMST20a, CMST20b]. In Appendix A we give the results of some numerical experiments comparing the performance and accuracy of Euler and Runge–Kutta numerical integration with explicit solutions in Nil and $\widetilde{\mathrm{SL}}(2,\mathbb{R})$.

Acknowledgements. This material is based in part upon work supported by the National Science Foundation under Grant No. DMS-1439786 and the Alfred P. Sloan Foundation award G-2019-11406 while the authors were in residence at the Institute for Computational and Experimental Research in Mathematics in Providence, RI, during the Illustrating Mathematics program. The first author acknowledges support from the *Centre Henri Lebesgue* ANR-11-LABX-0020-01 and the *Agence Nationale de la Recherche* under Grant *Dagger* ANR-16-CE40-0006-01. The second author was supported in part by National Science Foundation grant DMR-1847172 and a Cottrell Scholars Award from the Research Corporation for Science Advancement. The third author was supported in part by National Science Foundation grant DMS-1708239.

We thank Joey Chahine for telling us about a computable means of finding area density. We thank Arnaud Chéritat, Matei Coiculescu, Jason Manning, Saul Schleimer, and Rich Schwartz for enlightening discussions about the Thurston geometries at ICERM.

2. Ray-marching

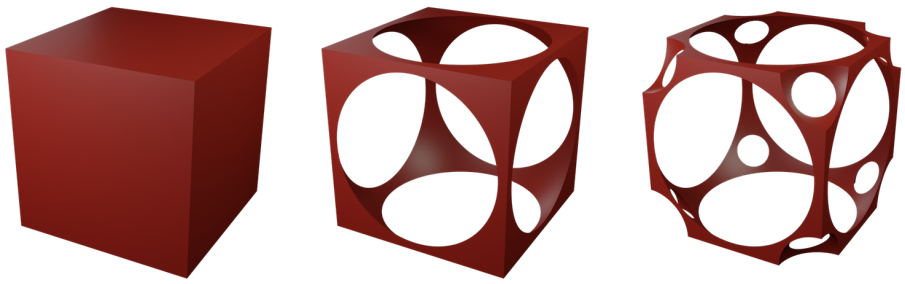
Ray-marching is a relatively new technique to produce real-time graphics using modern GPUs [Won], although its roots go back to the 1980's at least [HSK89]. Ray-marching is similar to ray-tracing in that for each pixel of the screen, we shoot a ray from a virtual camera to determine what color the pixel should be. Unlike most ray-tracing implementations however, the objects in the world that our ray can hit are not described using polygons. Instead, we use *signed distance functions*, which we describe in the following.

Definition 2.1. Let X be the ambient space, and suppose that S is a closed subset of X. We refer to S as a *scene*. We define the *signed distance function* $\sigma\colon X \to \mathbb{R}$ for S as follows. For a point $p \in X - S$, the function σ returns the radius of the largest ball centered at p whose interior is disjoint from S. For $p \in S$ the function is non-positive, and $|\sigma(p)|$ is the radius of the largest ball centered at p contained in S. $\Diamond$

We will sometimes write $\mathrm{sdf}(p, S)$ for $\sigma(p)$. We often refer to a part of a scene as an *object*. As an example, suppose that X is euclidean three-space, $\mathbb{E}^3$, and our scene S is a ball of radius R, centered at the origin. Then the signed distance function is

$$\sigma(p) = |p| - R. \tag{2.2}$$

Suppose that we have multiple scenes, described by signed distance functions σ_i. Then the signed distance function for the union of the scenes is $\min_i\{\sigma_i\}$. The complement of a scene is given by the negative of its signed distance function. We often draw a tiling in an inexpensive manner by deleting a ball from the center of each tile. See Figure 2.1 and Remark 4.4. For more examples of signed distance functions in $\mathbb{E}^3$, and more ways to combine signed distance functions, see [Quia].



(A) A tile.

(B) A ball is deleted from the center of the tile.

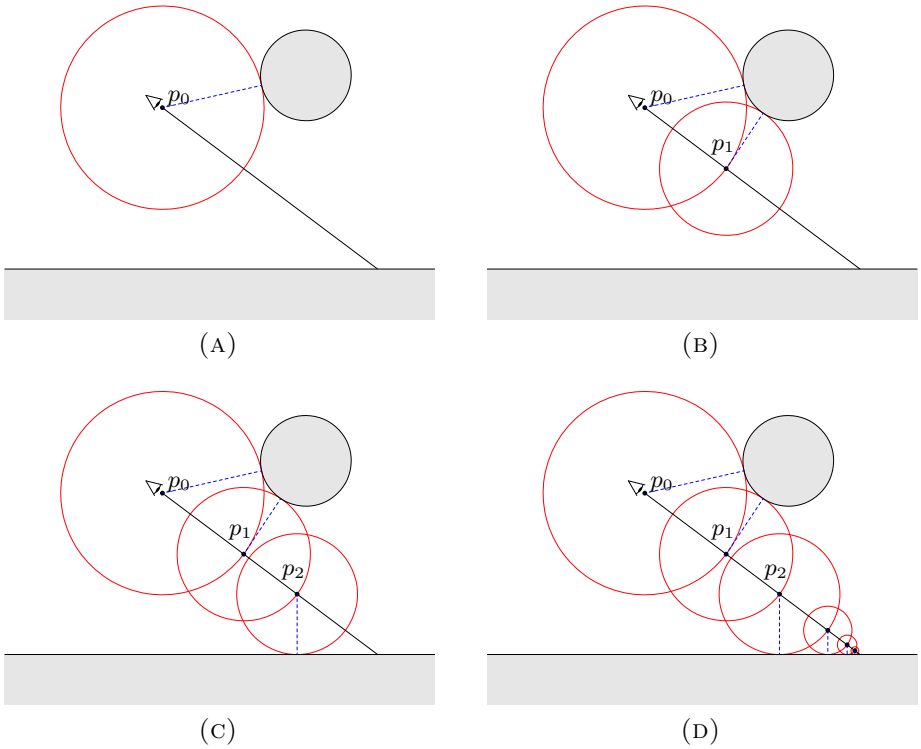
(C) A ball is deleted from the center and each vertex of the tile.

FIGURE 2.1. Extrinsic view of some scenes with inexpensive signed distance functions for a $\mathbb{Z}^3$-invariant tiling in $\mathbb{E}^3$.

To render an image of our scene, we place a virtual camera in the space X at a point p_0. We identify each pixel of the computer screen with a tangent vector at p_0, and so determine a geodesic ray for this pixel, starting at p_0. To color the pixel, we must work out what part of the scene the ray hits. The algorithm is illustrated in Figure 2.2. We start at p_0, the position of the camera, as shown in Figure 2.2a.

We assume that p_0 is not inside the scene. We evaluate the signed distance function σ at p_0. Since no part of the scene is within $\sigma(p_0)$ of p_0, we can safely march along our ray by a distance of $\sigma(p_0)$ without hitting the scene. We call the resulting point p_1. We can then safely march forward again by $\sigma(p_1)$ to reach p_2. We repeat this procedure until either we reach a maximum number of iterations, or we reach a maximum distance, or the signed distance function evaluates to a sufficiently small threshold value, ε say. In the first two cases we color the pixel by some background color. In the third case (as shown in Figure 2.2d) we declare that we have hit the scene.

In the case that we hit the scene, we may then choose a color for the pixel based on which part of the scene we hit, apply a texture, and/or apply various lighting techniques, for example the Phong reflection model [Pho75]. Note that this model requires the normal vector to the surface at the point our ray hits; this is easily approximated using the gradient of the signed distance function.



(A) (B) (C) (D)

FIGURE 2.2. Ray-marching to find the point at which a ray hits an object, for a scene in $\mathbb{E}^2$ consisting of a disk and a half-plane.

2.1. **Geometric convergence.** A concern one might have over the ray-marching algorithm is the potentially large number of steps taken before we are close enough to the scene to declare that we have hit it. Indeed, functions called in the innermost loop of the algorithm must be made as efficient as possible. However, the number of steps used is generally not prohibitive. Suppose that our scene S has a smooth boundary. In this case, when we are close enough to S its boundary may be approximated by a plane P. If our ray continues to approach P, then we converge to it as a geometric series, see Figure 2.3. The base of the exponent λ depends on the angle of incidence of the ray, approaching the worst case of $\lambda = 1$ as the ray becomes tangent to S.

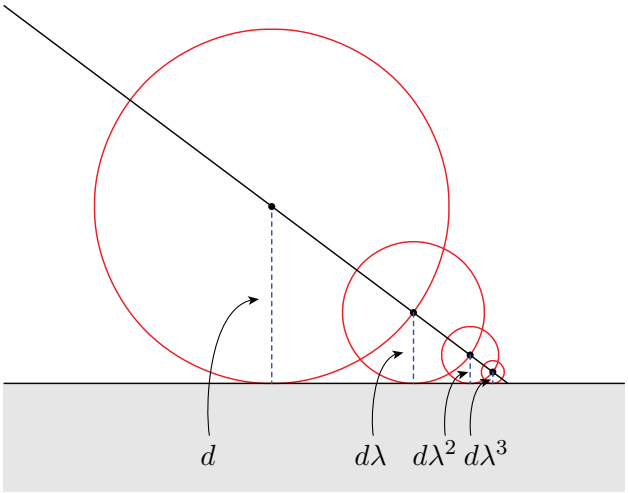


FIGURE 2.3. Typically, convergence of a ray marching into an object is geometric. If our first distance from the object is d, then subsequent distances follow a geometric sequence with the base of the exponent some number $\lambda < 1$.

Remark 2.3. If the maximum number of steps we allow before giving up is too small, then we may erroneously color pixels with the background color whose rays would eventually hit an object. This will often be most visible around the outer edges of an object in the scene, as these rays are the closest to tangent. These rays spend many steps moving a small distance close the object. Thus, they may run out of iterations before converging. ◊

2.2. **Distance underestimators.** The signed distance function for a scene may be difficult or expensive to calculate. In these cases we may wish to replace it with an easier to calculate approximation.

Definition 2.4. Suppose that $\sigma\colon X \to \mathbb{R}$ is the signed distance function for a scene S. We say that a function $\sigma'\colon X \to \mathbb{R}$ is a *distance underestimator* if

(1) The signs of $\sigma'(p)$ and $\sigma(p)$ are the same for all points $p \in X$,
(2) $|\sigma'(p)| \leqslant |\sigma(p)|$ for all $p \in X$, and
(3) If $\{p_1, p_2, \ldots\}$ is a sequence of points in X such that $\lim \sigma'(p_n) = 0$, then $\lim \sigma(p_n) = 0$. $\Diamond$

We do not require that σ' is continuous, but the second and third conditions here imply that a distance underestimator vanishes only on the boundary of S.

Lemma 2.5. *When ray-marching with a distance underestimator σ' in place of a signed distance function σ, we limit to the same point as when using σ.*

This result implies that a distance underestimator will give us essentially the same images as the signed distance function, given enough iterations and a small enough threshold ε. If a distance underestimator is significantly easier to compute than the signed distance function then trading an increased number of iterations for improved speed of computation can be advantageous. See Sections 9.6 and 10.6 for examples of distance underestimators.

Proof of Lemma 2.5. Consider a ray γ starting at a point $p \notin S$. Suppose that γ first meets the scene S at the point q. Using the distance underestimator σ', we march through a sequence of points $p = p_1, p_2, \ldots$ Consider the distances $d_n = \text{dist}_\gamma(p_n, q)$, measured along the ray γ from p_n to q. By conditions (1) and (2), we know that the sequence $\{d_n\}$ is a non-negative non-increasing sequence. Thus $\{d_n\}$ converges, and so the sequence of points $\{p_n\}$, converges. Thus the distances $\text{dist}_\gamma(p_n, p_{n+1})$ must go to zero. These are the distances we march along the ray, using the distance underestimator σ', so $\text{dist}_\gamma(p_n, p_{n+1}) = \sigma'(p_n)$. Therefore $\lim \sigma'(p_n) = 0$. By condition (3), $\lim \sigma(p_n) = 0$, and so $\lim p_n = q$. $\square$

In practice we want σ' and σ to be "coarsely the same". In particular, to get condition (3), we want $|\sigma'(p)|$ to be bounded below by some function of $|\sigma(p)|$. This also allows us to control how many extra iterations are needed in ray-marching with a distance underestimator.

Any real-world implementation cannot go all the way to the limit point q and instead stops at some sufficiently small value, ε. Thus, a distance underestimator should not return a value smaller than ε unless the signed distance function is also small.

2.3. **Advantages of ray-marching in non-euclidean geometries.** Ray-marching is an attractive technique in euclidean geometry, in part because of the simplicity of its implementation. This is also true for non-euclidean geometries. Here we discuss some alternative techniques.

2.3.1. *Z-buffer triangle rasterization.* Real-time graphics in euclidean geometry are usually rendered using *z-buffer triangle rasterization*. In this technique, objects in the scene are represented by polygon meshes. A projection matrix maps each triangle of a mesh onto the plane of the virtual camera's screen. For each pixel P, we look at the triangles whose projections contain the center of P. Of these triangles, the one closest to the camera determines the color of P.

This works well for the isotropic Thurston geometries, $\mathbb{E}^3$, S^3 and $\mathbb{H}^3$, in particular because geodesics in these geometries are straight lines in their projective models, see [Wee02]. Jeff Weeks uses these in his *Curved Spaces* software [Wee]. There is one complication with S^3 here, in that a single object is visible in two different directions: the two directions along the great circle containing the camera and the object. This means that each object must be projected twice. This is acceptable for S^3. In Nil, Sol, and $\widetilde{\mathrm{SL}}(2,\mathbb{R})$, a single object can be visible from the camera in many directions, with no uniform bound on the number of such directions. Even worse, in $S^2 \times \mathbb{E}$ a single object can be visible in infinitely many directions from a single camera position.

The projection matrix used in triangle rasterization implements the inverse of the exponential map. In the cases listed above, the exponential map is not one-to-one. This is not a problem for ray-tracing and ray-marching, which both use the forward direction of the exponential map instead.

2.3.2. *Ray-tracing.* Ray-tracing is very similar to ray-marching, with the difference being in how we determine where in the scene a ray hits. In many applications the objects in the scene are described by polygon meshes, as in triangle rasterization. The algorithm checks for intersection between the ray and the polygons of the mesh. To make this efficient for (euclidean) scenes with a large number of polygons, much effort is put into checking as few triangles for collision as possible, even though each individual check is inexpensive. However, objects described by simple equations such as spheres and other conics can also be used: all that is needed is a way to check whether or not a ray intersects the object, and at what distance along the ray. The distance is used to decide which object is closest to the camera and so should

be drawn. For a conic in euclidean space for example, this check and distance may be calculated by solving a quadratic equation.

One advantage of ray-tracing over ray-marching is that ray-tracing is well suited to rendering objects given by polygon meshes. It therefore has access to decades of development in polygon modeling techniques and rendering efficiency for polygonal models. On the other hand, depending on the geometry, checking for intersection between a ray and an object may be difficult. In place of this check in ray-tracing, for ray-marching we only need a signed distance function (or distance underestimator). If for example we make our scene from balls, then we only need to calculate distances between points.

2.4. **Accuracy.** One of our goals in this project is to be able to render features accurately, even at long distances. We identify two potential sources of error here.

2.4.1. *Floating point representation of number.* First, the representation of real numbers by floating point numbers is necessarily inaccurate. This can be a problem in a number of ways, whether one is ray-marching, ray-tracing, or using polygon rasterizing methods.

(1) In certain models, the coordinates of points grow exponentially with distance in the geometry, and floating point numbers quickly lose precision. In particular, this causes problems when rendering objects that are far from the camera. Of the eight Thurston geometries, this is an issue in $\mathbb{H}^3$, $\mathbb{H}^2 \times \mathbb{E}$, Sol, and $\widetilde{\mathrm{SL}}(2, \mathbb{R})$. This can be mitigated by the choice of model [FWW02]. Even without exponential growth in coordinates, floating point numbers cannot exactly represent geometric data.
(2) In certain regimes, a formula may be unstable. For example, the formula $(1 - \cos(t))/t^2$ approaches $1/2$ as t approaches zero. However, the available precision in the floating point representation of $(1 - \cos(t))$ near $t = 0$ is not very good in comparison to the precision of t^2. In such a regime, it is better to use a different representation of the formula. Here for example, we will get much better results by using an asymptotic expansion, say $1/2 - t^2/24 + \cdots$.

2.4.2. *Accumulation of errors.* Any iterative algorithm that takes the result from the previous step as the input for the next step may accumulate errors. These errors may come from lack of precision due to floating point representations as described above. They may also come from limitations in the methods used to calculate geodesic flow. As mentioned at the end of Section 1.3, to remove this second source of

error we avoid the numerical integration approach whenever possible, preferring explicit solutions.

3. General implementation details

As mentioned in Section 1.2, one of our goals in this project is to make as much of our code as possible independent of the geometry being simulated. Following this goal, in the next few sections we describe components needed for our simulations that are shared across geometries. Many of these apply to all eight Thurston geometries. However, it is also useful to discuss strategies for tackling smaller collections of geometries with certain geometric or group theoretic features. Thus to begin, we provide a second grouping of the Thurston geometries into overlapping families, distinct from our first grouping by method of construction in Section 1.1.

Consider the following properties:

(1) The geodesic flow is achieved by isometries. That is, every geodesic is the orbit of a point under a one-parameter subgroup.
(2) The projective model has straight-line geodesics. Each of the Thurston geometries (up to covers) has a model with $X \subset \mathbb{RP}^3$ and $G < \mathsf{GL}(4;\mathbb{R})$. With this property, the geodesics of (G, X) are projective lines in this model.
(3) The group G has a normal subgroup whose action is free and transitive on X.

Property (1) implies that parallel transport is achievable directly via elements of G. Property (2) implies that totally geodesic surfaces are planes in the projective model, which makes testing membership in polyhedral domains (for example, Dirichlet domains) efficient. Property (3) allows us to canonically identify tangent spaces at distinct points of X. This allows us to reduce certain difficult calculations (for example, the geodesic flow) to differential equations in a fixed tangent space.

The constant curvature and product geometries all have properties (1) and (2), while Nil, Sol, and $\widetilde{\mathrm{SL}}(2,\mathbb{R})$ have neither. These properties are very useful in practice, so we call the five geometries possessing them the *easier geometries*, while Nil, Sol, and $\widetilde{\mathrm{SL}}(2,\mathbb{R})$ are the *harder geometries*. However, Nil, Sol, and $\widetilde{\mathrm{SL}}(2,\mathbb{R})$ do have property (3) (along with $\mathbb{E}^3$ and S^3). See Figure 3.1.

3.1. **Notation.** Recall that the underlying space X of a Thurston geometry (G, X) is both connected and simply connected, and can be equipped with a G-invariant riemannian metric ds^2. We fix a base point

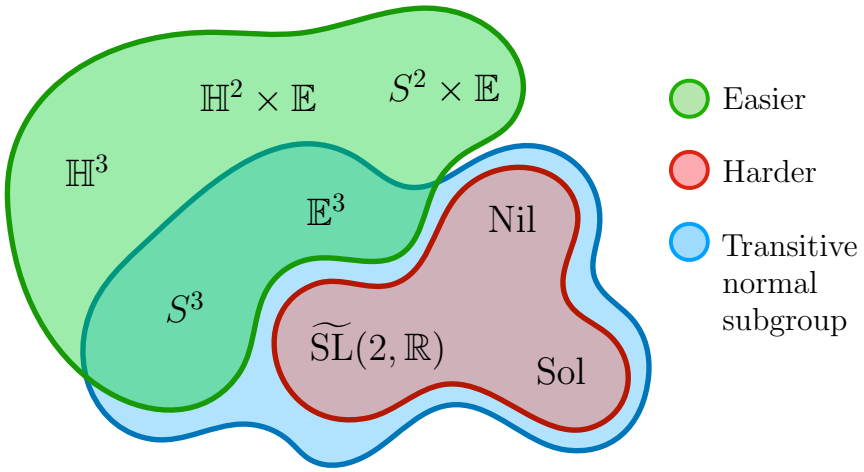


FIGURE 3.1. The Thurston geometries, grouped into useful categories for our implementation.

$o \in X$, which we call the *origin* of the space X. We denote by K the stabilizer of o in G. Thus X is isomorphic to G/K.

3.2. **Geodesic flow.** In order to follow light rays, we need to understand geodesics in X. Moreover, since we want to march along our geodesics by specified distances, they must be given by arc length parametrizations. These are paths $\gamma\colon \mathbb{R} \to X$ such that

$$\nabla_{\dot\gamma(t)}\dot\gamma(t) = 0, \quad \forall t \in \mathbb{R}.$$

where ∇ is the Levi-Civita connection on (X, ds^2). This condition corresponds to a five-dimensional second-order (non-linear) differential system. In some cases (for example, $\mathbb{E}^3$, S^3 or $\mathbb{H}^3$) these systems are comparatively easy to solve. See Table 1. Other geometries such as Nil, Sol, and $\widetilde{\mathrm{SL}}(2,\mathbb{R})$ are more subtle. Next, we describe a method to split this problem into two first-order differential systems. This strategy has both practical and theoretical advantages that we will discuss later.

3.2.1. *Grayson.* We follow here an idea of Grayson [Gra83]. Assume that G contains a normal subgroup G_0 which acts freely and transitively on X. The group G_0 provides a preferred way to compare the tangent space at different points of X. For every $x \in X$ we denote by L_x the (unique) isometry in G_0 sending the origin o to x. Let $\gamma\colon \mathbb{R} \to X$ be a geodesic of X. For every $t \in \mathbb{R}$, we denote by $u(t) \in T_oX$ the vector such that

$$\dot\gamma(t) = d_oL_{\gamma(t)}u(t) \tag{3.1}$$

It follows from the construction that u is a path on the unit sphere of the tangent space T_oX. Observe that once u is known, the trajectory γ is the solution of the first-order differential equation given by Equation (3.1).

Since geodesics are invariant under isometries, the path u satisfies a two-dimensional first-order autonomous differential system

$$\dot{u} = F(u) \tag{3.2}$$

where F does not depend on γ. In practice, Equation (3.2) is often straightforward to solve, see for example Section 9 and Section 10. The corresponding flow on the unit sphere also provides qualitative information on the geodesic flow [CS19].

Let $h \in K$ be an isometry on X fixing o. Observe that the path

$$u' = d_o h \circ u$$

is also a solution of Equation (3.2). Indeed, consider the geodesic $\gamma' \colon \mathbb{R} \to X$ defined by $\gamma' = h \circ \gamma$. Since G_o is a normal subgroup of G, for every $x \in X$ we have

$$h \circ L_x \circ h^{-1} = L_{hx}.$$

It follows that

$$\dot{\gamma}'(t) = d_o L_{\gamma'(t)} u'(t), \quad \forall t \in \mathbb{R}.$$

This proves our claim. Thanks to this observation we can take advantage of the symmetries of X to reduce the amount of computation needed to solve Equation (3.2). See for example Sections 9.3 and 10.2.

3.3. **Position and facing.** For the moment, we will think of the observer as a single camera, based at a point of X. In Section 3.6, we will consider an observer with stereoscopic vision.

In order to render the scene viewed by such an observer, we need to know its *position*, given by a point $p \in X$, and its orientation in the space (which we call its *facing*). The latter is represented by an orthonormal frame $f = (f_1, f_2, f_3)$ of the tangent space T_pX. We adopt the following convention: from the viewpoint of the observer,

- f_1 points to the right
- f_2 points upward
- f_3 points backward.

See Figure 3.2.

Let $\mathcal{O}X$ be the bundle of all orthonormal frames on X. We fix once and for all a reference frame $e = (e_1, e_2, e_3)$ at the origin o. This provides an identification of $\mathcal{O}_oX$, the space of orthonormal frames at o, with $\mathrm{O}(3)$. In particular, this induces an embedding of the stabilizer of the origin, K, into $\mathrm{O}(3)$, given by $k \mapsto d_o k$.

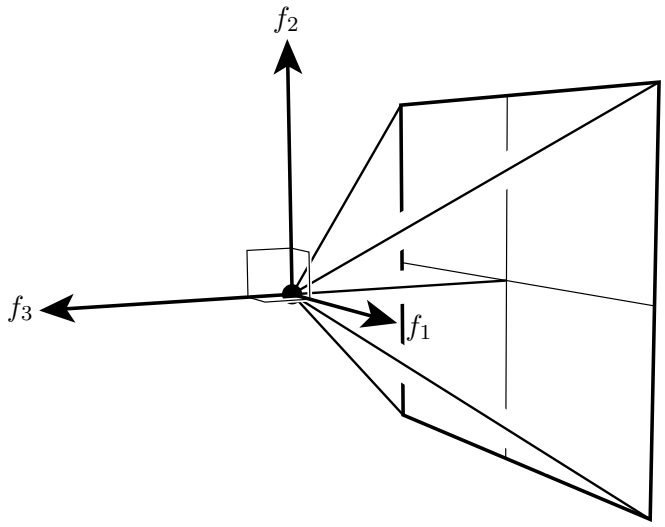


FIGURE 3.2. The initial tangent vector is of the form $sf_1 + tf_2 - f_3$, where s and t are coordinates on the screen.

3.3.1. *Parametrizing the frame bundle.* Our goal is to make simulations of Thurston geometries to better understand their properties. Our audience in this endeavor consists of entities with primary experience in $\mathbb{E}^3$, as far as we are aware. Thus, our audience will naturally expect to be able to move in any direction, and orient their view in any way they wish. Thus, the user should be able to move and rotate to achieve any element of the frame bundle $\mathcal{O}X$ (while preserving their orientation class). Therefore the data we use to record the position and facing of the user must map *onto* $\mathcal{O}X$.

When X is isotropic, G acts transitively on the frame bundle $\mathcal{O}X$. In this case one could use an element of G to record this data. However, when X is anisotropic, this action is not transitive. For example, if X is one of the product geometries $S^2 \times \mathbb{E}$ or $\mathbb{H}^2 \times \mathbb{E}$, there is no isometry that rotates in way that breaks the product structure.

Thus, we parametrize $\mathcal{O}X$ by the following map.

$$\begin{array}{ccc} G \times \mathrm{O}(3) & \to & \mathcal{O}(X) \\ (g, m) & \mapsto & d_o g \circ m(e) \end{array}$$

Since the action of G on X is transitive, there is an element g taking o to any given point $p = go$. The map $d_o g$ sends $T_o X$ to $T_p X$. By varying m, we can send the reference frame e to any frame in $T_p X$. Thus, the map is onto.

The group G acts on the left on $G \times \mathrm{O}(3)$ by multiplication of the first factor so that the map $G \times \mathrm{O}(3) \to \mathcal{O}(X)$ is G-equivariant. Note

that the stabilizer K of the origin o, also acts on the right on $G \times \mathrm{O}(3)$ as follows: for every $(g, m) \in G \times \mathrm{O}(3)$ and for every $k \in K$ we have

$$(g, m) \cdot k = \big(gk, d_o k^{-1} \circ m\big).$$

This action commutes with the left action of G. Moreover the application $G \times \mathrm{O}(3) \to \mathcal{O}X$ above induces a G-equivariant bijection from the quotient $(G \times \mathrm{O}(3))/K$ to $\mathcal{O}X$.

3.3.2. *Using a transitive normal subgroup.* For geometries with a transitive normal subgroup $G_0 < G$ of isometries, there is a natural section of the frame bundle $X \to \mathcal{O}X$ given by the G_0-orbit of the reference frame e at the origin. Using this frame, we can encode unit tangent vectors in T_pX by points of the unit sphere of $\mathbb{R}^3$. The coordinates needed to describe these unit tangent vectors are thus uniformly bounded at all points $p \in X$. This choice of representation helps reduce numerical errors, for example its implementation in Sol.

3.4. **Moving in the space.** Using the parameterization above, a pair $(g, m) \in G \times O(3)$ specifies a location $p \in X$ of the user, and a frame f in T_pX. This provides the necessary data to orient the user's virtual camera within the space. To produce a real-time simulation, we need a means of converting user input into this form.

Assume that at the current frame, the virtual camera is at a point $p \in X$. At each frame of the simulation, the virtual reality system records the position and facing of the headset in the play area, which is (very well) approximated as a subset of $\mathbb{E}^3$. We interpret the change in position between this frame and the next as a tangent vector $v \in T_pX \cong \mathbb{E}^3$, given by coordinates in the local frame $f = (f_1, f_2, f_3)$ representing the facing of the observer. Alternatively, keyboard input can provide the same information.

Remark 3.3. There is a choice to be made here in the relationship between the distance moved in the real world and the magnitude of the vector v. In our implementation, by default one meter in the real world corresponds to one unit in the virtual world. One may wish to change this relationship by a scaling factor to, for example, vary the perceived effects of curvature in $\mathbb{H}^3$ [Tre18]. ◊

We move the observer along the geodesic $\gamma\colon \mathbb{R} \to X$ such that $\gamma(0) = p$ and $\dot\gamma(0) = v$. In addition, we update the facing of the observer using parallel transport. Parallel transport along γ can be seen as a collection of orientation-preserving isometries

$$T(t)\colon T_{\gamma(0)}X \to T_{\gamma(t)}X$$

such that

$$\nabla_{\dot\gamma(t)} T(t) = 0, \quad \forall t \in \mathbb{R}. \tag{3.4}$$

In the easier geometries ($\mathbb{E}^3, S^3, \mathbb{H}^3, S^2 \times \mathbb{E}$, and $\mathbb{H}^2 \times \mathbb{E}$), for each geodesic γ through a point p, there is a one-parameter subgroup $\{g(t)\} \subset G$ such that $\gamma(t) = g(t)p$. In these cases, the parallel transport operator is $T(t) = d_p g(t)$.

3.4.1. *Using a transitive normal subgroup.* In Nil, Sol, and $\widetilde{\mathrm{SL}}(2,\mathbb{R})$, we do not have the above one-parameter subgroup. Instead, in order to compute the path of isometries $t \to T(t)$ we again use Grayson's method. Assume as above that G_0 is a connected normal subgroup of G acting freely and transitively on X. Define $u : \mathbb{R} \to T_oX$ by the relation

$$\dot\gamma(t) = d_o L_{\gamma(t)} u(t)$$

where L_p is the unique isometry of G_0 sending o to p. Similarly, we define a path $Q\colon \mathbb{R} \to \mathrm{SO}(3)$ by letting

$$T(t) \circ d_o L_{\gamma(0)} = d_o L_{\gamma(t)} \circ Q(t) \tag{3.5}$$

It turns out that for each of our harder geometries, Q satisfies a linear differential equation of the form

$$\dot Q + B(u)Q = 0 \tag{3.6}$$

where B is skew-symmetric matrix which only depends on u (and not on γ) and with initial condition $Q(0) = \mathrm{Id}$. To solve Equation (3.6) we use the following observation. By definition of parallel transport, for every $t \in \mathbb{R}$, we have $T(t)\dot\gamma(0) = \dot\gamma(t)$, hence

$$Q(t)u(0) = u(t).$$

Fix now an arbitrary vector $e_0 \in \mathbb{R}^3$ and a path $R\colon \mathbb{R} \to \mathrm{SO}(3)$ such that $R(t)u(t) = e_0$, for every $t \in \mathbb{R}$. Then

$$S(t) = R(t)Q(t)R(0)^{-1}$$

is a rotation of angle $\theta(t)$ around $\mathbb{R}e_0$. Hence, in order to compute Q, and thus T, it suffices to know the value of the angle θ. To that end, we substitute $Q(t) = R(t)^{-1}S(t)R(0)$ into Equation (3.6) and obtain a first-order differential equation on θ that we solve. This strategy gives an effective way to compute the parallel-transport operator.

Assume that $k \in K$ is an isometry of X fixing o and let $u' = d_o k \circ u$. We observed previously that u' is also a solution of Equation (3.2). With the same kind of computation we get that $Q'(t) = d_o k \circ Q(t) \circ d_o k^{-1}$ is a solution of

$$\dot Q' + B(u')Q' = 0$$

Again we can use the symmetries of X to reduce the amount of computation needed to solve Equation (3.6).

During a motion it is convenient to use the pulled-back parallel-transport operator Q to update the position and facing. Recall that we store the position and facing of the observer as a pair $(g, m) \in G \times \mathrm{O}(3)$. At time $t = 0$ the observer is at the point $\gamma(0) = go$ where $g = L_{\gamma(0)}$. Its facing is given by the frame

$$f = d_o g \circ m(e) = d_o L_{\gamma(0)} \circ m(e)$$

After moving along the geodesic γ for time t the observer reaches the point $\gamma(t)$. The observer's new facing corresponds to the frame

$$f' = T(t)f = T(t) \circ d_o L_{\gamma(0)} \circ m(e).$$

By the definition of Q, we get

$$f' = d_o L_{\gamma(t)} \circ Q(t) \circ m(e).$$

Hence the position and facing of the observer after time t is given by the pair $(L_{\gamma(t)}, Q(t)m)$.

3.5. **Rendering an image from a fixed location.** Assume that the position and the facing of the observer is given as pair $(g, m) \in G \times \mathrm{O}(3)$. In order to render what the observer would see, we proceed as follows. Let p be the point obtained by applying g to the origin o. Recall that the observer is looking in the direction $-f_3$, where $f = (f_1, f_2, f_3)$ is the frame $f = me$. The set of vectors $u \in T_pX$ such that $\langle u, f_3 \rangle = -1$ defines an affine plane P in T_pX. We identify the screen of the computer with a rectangle in P centered at $-f_3$. See Figure 3.2. The exact size of the rectangle is computed in terms of the field of view of the observer. For each vector $u \in T_pX$ in this rectangle, we follow (using the ray-marching algorithm) the geodesic starting at p in the direction of u (or more precisely the unit vector with the same direction) until it hits an object. We color the corresponding pixel on the screen with the color of this object, or more realistically, using a physical model of lighting as described in Section 5.

The formulas for geodesic flow starting from an arbitrary point p can be efficiently factored using the homogeneity of X. That is, a conjugation by g identifies the flow from o with the flow from p. In practice, for the easier geometries one might as well work at the position of the observer, p, rather than at o. However, for the harder geometries, this significantly simplifies the code.

3.6. **Stereoscopic vision.** A virtual reality headset has a separate screen for each eye. This allows it to show the two eyes slightly different images – parallax differences between these images can then be interpreted by the user's brain to give depth cues.

Given positions and facings for the left eye, $(p^\triangleleft, f^\triangleleft)$, and the right eye, $(p^\triangleright, f^\triangleright)$, we can render an image for each eye exactly as in Section 3.5. The question is how to determine the positions and facings for the two eyes. Let ℓ be the *interpupillary distance*; that is, the distance between the eyes. We track the position and facing (p, f) of the user's nose, using the sensors of the virtual reality headset as in Section 3.4. In $\mathbb{E}^3$, the canonical thing to do is to set $f^\triangleleft$ and $f^\triangleright$ equal to f, and to set

$$p^\triangleleft = p - (\ell/2)f_1 \qquad p^\triangleright = p + (\ell/2)f_1$$

recalling that f_1 is the frame vector in f pointing to the right.

This works because in euclidean space, one may naturally identify the tangent spaces at all points. For non-euclidean geometries, a natural analogue is as follows. We set $(p^\triangleleft, f^\triangleleft)$ to be the result of flowing from (p, f) for distance $\ell/2$ in the direction of $-f_1$, and we set $(p^\triangleright, f^\triangleright)$ to be the result of flowing from (p, f) for distance $\ell/2$ in the direction of f_1.

This works reasonably well for S^3, $\mathbb{H}^3$, and $\mathbb{H}^2 \times \mathbb{E}$, although there are some problems. As mentioned in [HHMS17b, Section 6], in geometries in which geodesics diverge, parallax cues tell our euclidean brains that all objects are relatively nearby. In $\mathbb{H}^3$ for example, two eyes pointing directly at an object that is infinitely far away are angled towards each other. One alternate strategy we briefly experimented with was to rotate the frames $f^\triangleleft$ and $f^\triangleright$ slightly inwards, so that geodesics emanating from $p^\triangleleft$ and $p^\triangleright$ in the directions of their forward vectors $-f_3^\triangleleft$ and $-f_3^\triangleright$ converge at infinity. This might then match the behavior our euclidean brains expect: that objects at infinity can be seen by looking straight ahead with both eyes. We did not notice much difference in our ability to perceive the space in making this change, although this line of thinking leads us to conclude that predators in hyperbolic space would evolve to look somewhat cross-eyed to us native euclideans.

In S^3, points at distance $\pi/2$ away from the user appear to be "infinitely far away", while objects further than $\pi/2$ away have depth cues reversed. One possible future direction to try to improve this experience is as follows. Modern virtual reality headsets have the ability to track where the user's eyes are looking. Based on this information, we could determine what object the user is looking at. Using the distance from the viewer to the object, we could rotate the frames $f^\triangleleft$ and $f^\triangleright$ to imitate the effects of parallax for objects at that distance in $\mathbb{E}^3$. It

remains to be seen whether or not these frequent rotations would induce nausea.

The situation is worse in $S^2 \times \mathbb{E}$, Nil, Sol, and $\widetilde{\mathrm{SL}}(2, \mathbb{R})$, where geodesics "spiral". Figure 3.3 illustrates how a small parallax in Nil can produce very different pictures: On each row, the scene consists of a single ball textured as the earth. The different images are views of this ball from slightly different positions. Using the convention that one unit represents one meter, the offset between two consecutive images is approximately half the interpupillary distance. Our euclidean brains are not able to interpret the combination of these pictures. One might think that the sphere is too small (a few centimeters) and too far away form the observer (a few meters) for our eyes to see that level of detail. However geodesic rays in Nil spiral in such a way that the angular size of the object in the observer's view is very large. This makes the object appear as if it is very close to the observer. Thus this parallax distortion cannot be ignored. New ideas are thus needed to produce stereoscopic images in all eight geometries that can be pertinently analyzed by the brain. For now, foregoing stereoscopic vision and supplying the same image to each screen of a virtual reality headset still gives a more direct experience than one gets with a keyboard and monitor interface.

3.7. **Signed distance functions in X.** The algorithms described so far render the in-space view of a scene in the geometry X, given a signed distance function $\sigma\colon X \to \mathbb{R}$ for it. In the interest of both simplicity and geometric accuracy, we focus on scenes built from intrinsically defined objects, including

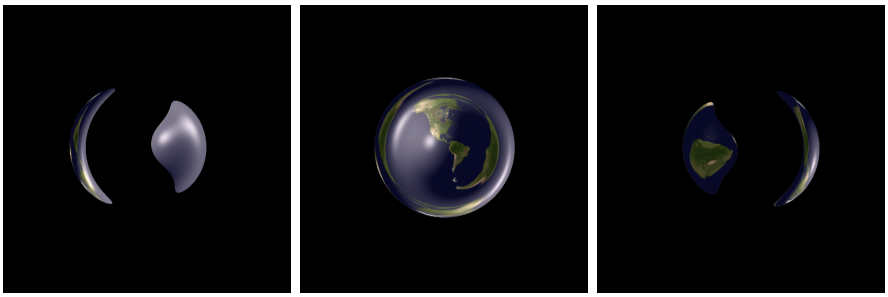
- balls (bounded by equidistant surfaces from a point),
- solid cylinders (bounded by equidistant surfaces from a geodesic), and
- half-spaces (bounded by totally geodesic codimension one submanifolds).

Note that a single object may fall into more than one of the above categories. For example, a hemi-hypersphere of S^3 is both a ball and a half-space.

3.7.1. *Simple Scenes.* In some cases, viewing and moving relative to a single simple object is all that is needed to illustrate surprising features of a geometry. In previous work for example, we qualitatively described counterintuitive features of Nil geometry [CMST20a] with a scene consisting of a single ball, and we studied a single isometrically embedded copy of the euclidean plane in Sol geometry [CMST20b]. From a collection of basic objects, many other simple scenes can be

$\Delta = -0.03$ $\Delta = 0$ $\Delta = +0.03$

(A) $r = 0.2$, $L = 7.3$, $\alpha = 68°$ and $L' = 0.36$.



$\Delta = -0.03$ $\Delta = 0$ $\Delta = +0.03$

(B) $r = 0.06$, $L = 6.55$, $\alpha = 39°$ and $L' = 0.18$.

FIGURE 3.3. Parallax in Nil makes stereoscopic vision difficult. The earth has radius r and is centered at the origin. In the middle picture the observer is located on the z-axis at a distance L from the origin. On the left and right pictures, the observer is offset by a distance Δ in the x-direction. The angular size of the ball in the observer's view is α. Note that due to the spiraling of geodesics in Nil, this angular size is much larger than it would be for an equivalent ball in euclidean space. Indeed, an observer assuming that they are in euclidean space would think that the ball is at distance L' from them.

created through finitely many applications of union, intersection and difference. These operations of constructive solid geometry are particularly suited to producing scenes in a ray-marching application, as $\{\cup, \cap, \setminus\}$ are faithfully represented on the space of signed distance functions by $\{\min, \max, -\}$ respectively [Quia].

In many cases however, the interesting features of the geometry are best exhibited by more complex, unbounded scenes, which cannot be built from the basic objects in finitely many operations.

3.7.2. *Complex Scenes and Symmetry.* Scenes which display interesting features across unbounded regions are useful to highlight various geometric features, including

- exponential growth of volume in negative curvature,
- anisotropy in the product geometries,
- non-integrability of the contact distribution in Nil, and
- the lack of any continuous rotation symmetry in Sol.

The particular details of the scene's contents do not matter so much as the requirement that the user may travel unbounded distances in any direction and still be surrounded with an approximately homogeneous collection of objects.

One way to do this is to use the homogeneity of X to build an extremely symmetric scene, by choosing a signed distance function $\sigma\colon X \to \mathbb{R}$ invariant under the action of a discrete subgroup $\Gamma < G$.

As geometric topologists however, we cannot help but note that covering space theory provides an alternative perspective. Consider a scene invariant under the action of Γ. This is described by a signed distance function $\sigma\colon X \to \mathbb{R}$ with $\sigma \circ \gamma = \sigma$ for all $\gamma \in \Gamma$. Such maps are in natural correspondence with maps from the quotient $\overline{\sigma}\colon X/\Gamma \to \mathbb{R}$.

Indeed, the view from a point $q \in X/\Gamma$ of a signed distance function $\overline{\sigma}$ is identical to the view from a lift $\widetilde{q} \in X$ of a signed distance function σ invariant under Γ. This follows from the above topological correspondence together with the fact that the covering map is a local isometry.

This suggests exploring the unbounded geometry of X indirectly, through the geometry of its quotients X/Γ.

4. Non-simply connected manifolds

Let (G, X) be a homogeneous geometry. A (G, X)-manifold is a smooth manifold M together with an atlas of charts

$$\{(U_\alpha \subset M, \quad f_\alpha\colon U_\alpha \to X)\}$$

with transition maps in $G = \mathrm{Isom}(X)$. The elementary theory of such (G, X)-manifolds shows that one may globalize this atlas into a *developing map* from $\widetilde{M}$ to X, equivariant with respect to a *holonomy homomorphism* from $\pi_1 M$ to G [Gol]. Furthermore, if M is geodesically complete, then the developing map is a diffeomorphism and $M \cong X/\Gamma$ is a quotient, where $\Gamma \cong \pi_1(M)$ is the image of the holonomy homomorphism. The simplest (G, X)-manifold is X itself, and we have seen above how to ray-march simple scenes in X. Covering space

theory implies that X is the unique complete simply connected (G, X)-manifold, but non-simply connected (G, X)-manifolds abound. Indeed the classification of compact hyperbolic manifolds up to diffeomorphism is still incomplete. Additionally, while there are only ten euclidean manifolds up to diffeomorphism, there are uncountably many distinct euclidean structures in each diffeomorphism class. Simulating not just the Thurston geometry X but also various (G, X)-manifolds is a natural extension of our original goals. These manifolds may or may not have finite volume, corresponding to the discrete subgroups $\Gamma < G$ being lattices or not. Generalizing further, our algorithms can also simulate (G, X)-orbifolds and incomplete (G, X)-manifolds. Thus we may experience both the three-dimensional homogeneous spaces, and also the atomic building blocks of geometrization.

In the next section, we describe a method to ray-march (or ray-trace) within a quotient manifold, using a fundamental domain. Similar ideas are outlined in [BLV15] and [KCK20].

4.1. **Teleporting.** Let Γ be a discrete subgroup of G, and $M = X/\Gamma$. To produce an intrinsic simulation of M, we wish to reuse as much as possible the work that goes into producing a simulation of X. To that end, we describe M using a connected fundamental domain $D \subset X$ with $2n$ faces $\{F_i^{\pm}\}_{i=1\dots n}$. (Alternatively, one could embed M in a higher-dimensional ambient space, and try to implement the techniques of Section 3 in that context.) The quotient manifold M is obtained by identifying each F_i^- with F_i^+ via an isometry $\gamma_i \in \Gamma$. These face pairings form a generating set $\{\gamma_1, \dots, \gamma_n\}$ for Γ. This allows us to ray-march using the geodesic flow on $D \subset X$, and calculate parallel transport and position/facing using the parametrization of $\mathcal{O}X$ restricted to D. Indeed, given a signed distance function $\sigma\colon X/\Gamma \to \mathbb{R}$ pulled back to D, the only substantial change is that we must modify the ray-march algorithm to keep the geodesic flow in D. We can do this by using the face pairings. Similarly, when the user moves outside of D, we move them by an isometry to keep them inside of D. In either case, we call this process *teleporting*. See Figure 4.1.

Remark 4.1. As a side benefit, the quotient manifold approach helps with floating point errors. At each step of our ray-marching algorithm, the basepoint of our ray is within D. In the case that M is compact for example, the coordinates of our basepoint are bounded by a function of the diameter of D. This then avoids problem (1) of Section 2.4.1. In our experience, we see less noise in images such as Figure 1.1c with this strategy, despite the potential accumulation of errors (see Section 2.4.2)

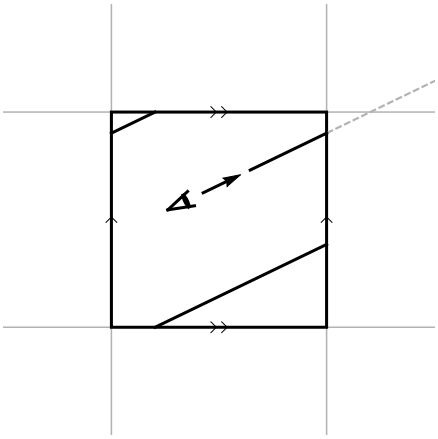


FIGURE 4.1. A light ray traveling in a domain D must teleport at the boundary to simulate the view within a torus.

introduced by repeatedly teleporting a ray's position and tangent vector back inside of D. ◊

Remark 4.2. It may be useful to employ teleporting even when we are simulating a scene inside of the simply connected geometry X rather than inside a quotient manifold. That is, we have a discrete subgroup of isometries and a fundamental domain D, and we use teleportation to keep the viewer always within D. Whenever we teleport the user, we also teleport all other objects in the scene, and update the signed distance function as appropriate. The advantage here is that rays begin inside of D, where their coordinates are small. Therefore floating point errors only accumulate to a noticeable degree for objects which are far from the viewer. For some geometries, such distant objects will be very small on the visual sphere. Alternatively, they may be hidden by fog. ◊

4.1.1. *Teleporting with a Dirichlet domain.* A simple, geometry independent implementation involves choosing the Dirichlet domain D for the action of Γ, centered at the origin $o \in X$. To determine whether or not a point p is outside of D, we compare the distance $d(p, o)$ with $d(p, \gamma_i^{\pm} o)$ for each face pairing isometry γ_i. When $d(p, o) > d(p, \gamma_i^{\pm} o)$, the point p can be brought back closer to o via an application of $\gamma_i^{\mp}$. Iterating this (relabelling our point as p after each step) until $d(p, o) \leqslant d(p, \gamma_i^{\pm} o)$, we ensure that p is inside of D.

An advantage of this approach is that one does not need an analytic description of the boundary ∂D to accurately adjust the ray-march. When the intrinsic distance d is expensive to calculate however, this adds a significant extra computational burden.

4.1.2. *Teleporting with a projective model and linear algebra.* A second implementation that removes the need to calculate distances is possible for the Thurston geometries. Up to covers (in the cases of $\widetilde{\mathrm{SL}}(2,\mathbb{R})$ and S^3), these have *projective models*: a representation of the geometry as an open subset $r\colon X \hookrightarrow \mathbb{RP}^3$, together with a linear representation $\mathrm{Isom}(X) \to PGL(4;\mathbb{R})$ [Mol97].

To lighten the notation in this section, we identify X with its image under r. We choose our fundamental domain D for the action of Γ such that $D = \bigcap_i H_i^{\pm}$, where $\{H_i^{\pm}\}$ is a collection of $2n$ half-spaces of X. The point p is outside of D if and only if there is a half-space $H_i^{\pm}$ such that $p \notin H_i^{\pm}$. Each half-space H of $\mathbb{R}^3$ is in natural correspondence with a linear functional $\phi\colon \mathbb{R}^3 \to \mathbb{R}$, where $v \in H$ if and only if $\phi(v) \geqslant 1$, so we can check if $p \in H_i^{\pm}$ by computing the value $\phi_i^{\pm}(p)$. The embeddings $r\colon X \to \mathbb{RP}^3$ are inexpensive to compute in our models (see Table 1): for $S^3, \mathbb{H}^3, S^2 \times \mathbb{R}, \mathbb{H}^2 \times \mathbb{R}$ we divide by the fourth coordinate, and $\mathbb{E}^3$, Nil, Sol are already affine patches. The situation for $\widetilde{\mathrm{SL}}(2,\mathbb{R})$ is slightly more complicated, but similar ideas work for the fundamental domains we have implemented. Thus, we reduce the problem to a quick calculation in linear algebra.

Knowing which of the half-planes p is not contained in, we now must find the element of Γ which moves p back into D. We iteratively construct this element from the $\gamma_i^{\pm}$ for which (at each step) $\phi_i^{\pm}(p) > 1$. In many cases (for example when Γ is a finite index subgroup of a reflection group), it does not matter which such $\gamma_i^{\pm}$ we choose at each step. In other cases, for reasons of efficiency, one must be more careful with the ordering, see for example Section 9.9.

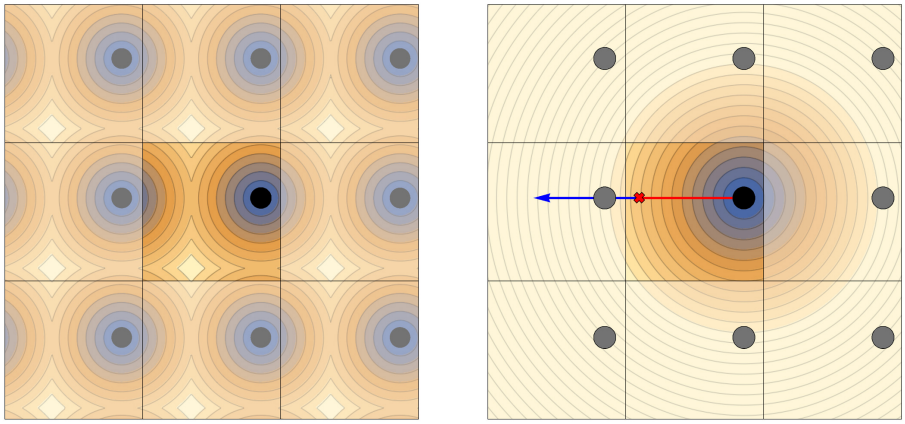
Since we have projective models for the eight Thurston geometries, we use this strategy rather than the Dirichlet domain strategy.

Remark 4.3. In practice, when using the projective model we can take $S = \{\gamma_i\}$ to be an arbitrary generating set for Γ. We then generate the half-spaces $H_i^{\pm}$ from S. Their intersection forms a fundamental domain D. Note that multiple faces of D may lie in the boundary of a single half-space, and the face pairings of D may involve elements of Γ other than those in S. However, we need only use elements of S to implement teleportation. See Section 9.9 for a detailed example. $\Diamond$

4.2. **Signed distance functions in X/Γ.** With the addition of teleportation, we may draw scenes in any complete (G,X)-manifold using the same algorithms as we use in X itself, given the input data of a signed distance function mapping X/Γ to $\mathbb{R}$ describing the scene. Unfortunately, efficiently calculating a signed distance function (or even

a distance underestimator) for a scene in a quotient manifold is often non-trivial. In practice, we will often use an approximation.

We can construct a very simple approximation for a scene S as follows. Let $D \subset X$ be a fundamental domain for the quotient manifold X/Γ. We then view S as a subset of D. For a point $p \in D$, we may then return the signed distance from p to S, where we measure distance in X, ignoring the quotient manifold structure entirely. Let us call this simplest approximation $\sigma\colon X \to \mathbb{R}$. (Here we implicitly extend the signed distance function from D to X.)



(A) The signed distance function for a disk in a torus, drawn in the universal cover.

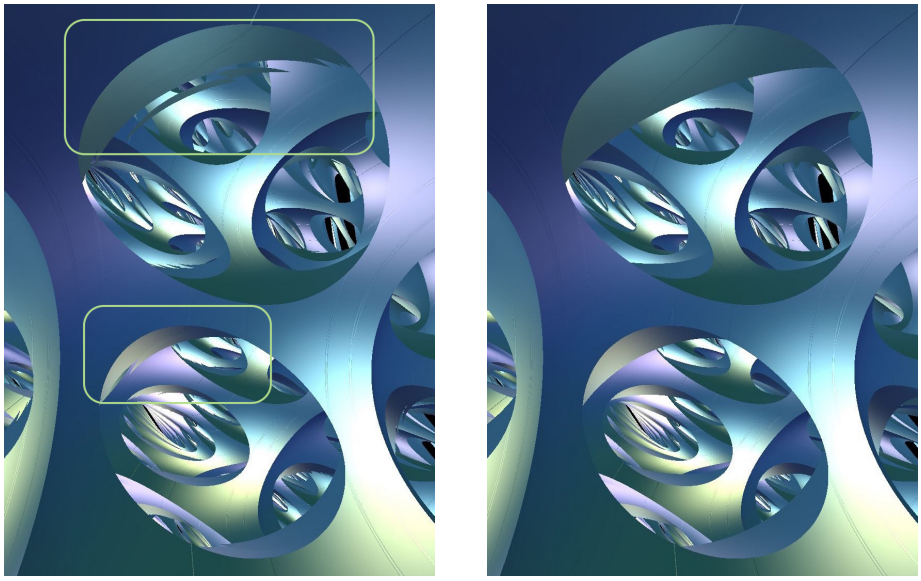
(B) The simplest approximation to the signed distance function, σ.

FIGURE 4.2. Functions on a torus. We indicate the level sets by bands of color.

As an example, Figure 4.2a shows the correct signed distance function for a disk in a square torus, while Figure 4.2b shows σ. For such a square torus, $\sigma|_D$ will be the correct signed distance function for the quotient torus only if the disk is centered in the square. Using $\sigma|_D$ in place of the correct signed distance function can lead to some serious visual artifacts. For example, consider a ray starting at the position p marked with a small red "$\times$" in Figure 4.2b and heading to the left. This ray should leave through the left side of D, teleport to the right side of D, then hit the disk. However, the function $\sigma|_D$ reports that the distance from p to the disk (indicated with the red interval) is more than half the width of the square. A march along the ray by this distance is shown with the blue arrow: we jump straight through the disk. The result is that this lift of the disk is invisible when viewed from p.

A similar but less extreme form of visual artifact is shown in Figure 4.3a. Here we see jagged errors on the boundaries between cells. In some places near the boundary of D we erroneously jump through points of the scene. Whether or not we make such a jump depends on how close to the boundary of D we land before jumping across the boundary. The variability in this leads to the jaggedness. Figure 4.5a shows related artifacts.

4.2.1. *Creeping over the boundary of D.* One strategy to avoid these kinds of errors uses the observation that flowing by the distance given by σ is only dangerous if our ray leaves D. Thus, we should detect when a ray passes outside of D, and stop just outside. As usual, we are teleported back inside of D, and continue ray-marching.



(A) No Creeping. (B) Creeping.

FIGURE 4.3. Allowing the ray-march to leave the fundamental domain can cause visual artifacts on objects near its faces. Creeping up to the boundary fixes this.

Detecting when a ray hits ∂D is a similar problem to that of detecting when the ray hits an object in the scene. We employ a variety of different methods, as follows.

(1) One way to do this is to use ray-tracing: we solve for the intersection between the ray and the boundary, and measure the distance between this intersection point and the start of the ray.

(2) If it is difficult to solve for this point of intersection, but the faces of D have computable signed distance functions, then we can instead use ray-marching. We flow by the minimum of σ and the distance to ∂D.[2]
(3) When the faces do not have computable signed distance functions but we can still detect whether or not we are inside of D, we proceed as follows: We flow by the distance given to us by σ, and ask if the result puts us outside of D. If it does, then we perform binary search on the distance we flow to find a point just outside of D.

Creeping just over the boundary solves the problem shown in Figure 4.3a, giving the correct image, Figure 4.3b. In general, creeping produces the correct pictures as long as all objects in the scene are contained within the domain D. However, this breaks down if we wish to, for example, move a ball from one domain to another. When a ball intersects ∂D, calculating the approximation σ requires measuring the distance to the center of the ball in D, and at least one translate of its center under some element of Γ. See Figure 4.4. Without this extra calculation, one sees objects cut in half by the boundary of D. See Figure 4.5b. Solving this problem led us to the following alternate (or additional) strategy to creeping.

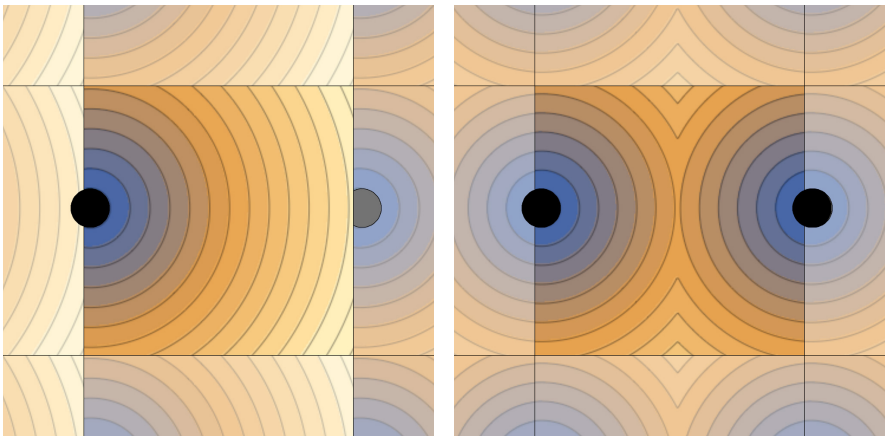
4.2.2. *Nearest neighbors signed distance functions.* Here we use a signed distance function on D that takes into account the effects of the nearby translates of D.

Let $A \subset \Gamma$ be a set of isometries. Define

$$\sigma_A = \min_{a \in A} \{\sigma \circ a\}$$

For example, $\sigma_{\{\mathrm{id}\}}$ is just σ, and σ_Γ is the correct Γ-invariant signed distance function. If Γ is infinite, then we cannot calculate σ_Γ directly. However, if the tiling of X by copies of the fundamental domain is locally finite, then there is a finite subset $A \subset \Gamma$ such that σ_A and σ_Γ are equal on D. Indeed, we may choose for A the set of all $\gamma \in \Gamma$ such that the distance from D to $\gamma(D)$ is at most the diameter of D. Depending on the shape of the fundamental domain and how it is glued to itself however, the size of A may be large. If so, calculating this signed distance function may be prohibitively expensive.

[2]In practice, we allow a march of the distance to the nearest wall plus some small ε: this prevents wasting many steps approaching the boundary to no appreciable theoretical disadvantage: the teleportation scheme returns us to D immediately upon overstep.



(A) An incorrect calculation of σ, using only the disk whose center is in D.

(B) The correct calculation of σ requires calculation of the distance to at least two points.

FIGURE 4.4. Calculating σ for a disk overlapping the boundary of D.

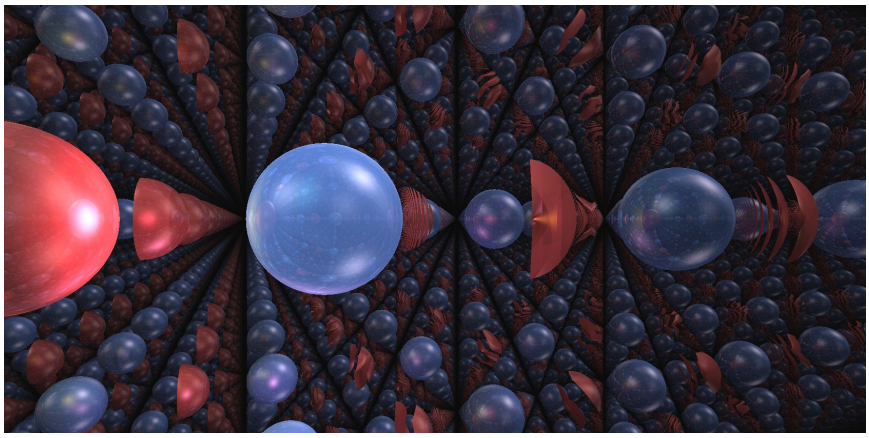
We find that most visual artifacts can be resolved without the use of creeping by using σ_A, where $A = \{\mathrm{id}\} \cup \{\gamma_i^{\pm}\}$. That is, we use σ in D and its nearest neighbors, directly connected by face pairings. See Figure 4.5c. In some circumstances this may not be enough; see for example Figure 4.6. Here a ray passing close to a vertex of the tiling may not see an object diagonally adjacent to the starting domain. In three dimensions the equivalent problem can appear for rays crossing close to an edge of the tiling.

In general, depending on the circumstance, either creeping or using a nearest neighbors signed distance function, or some combination of the strategies may be the most efficient strategy to obtain correct images. Even the combination of both strategies can produce errors in some circumstances. In Figure 4.7, the only solution would be to use more translates of σ than just the nearest neighbors.

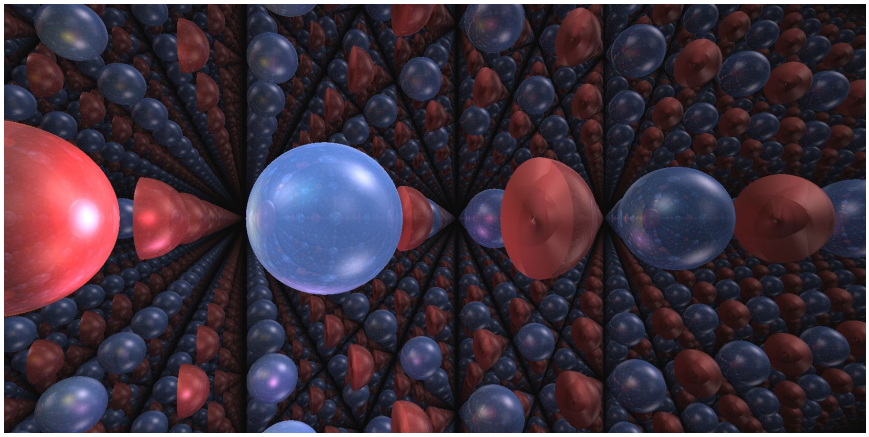
Remark 4.4. We would like to choose a scene for X/Γ which illustrates the geometry and topology while having a signed distance function that is very efficient to calculate. We often use the following strategy. We delete from a fundamental domain D a large ball (or solid ellipsoid). The signed distance function for the complement of a ball in D is

$$\sigma(p) = r - \mathrm{dist}(o, p).$$

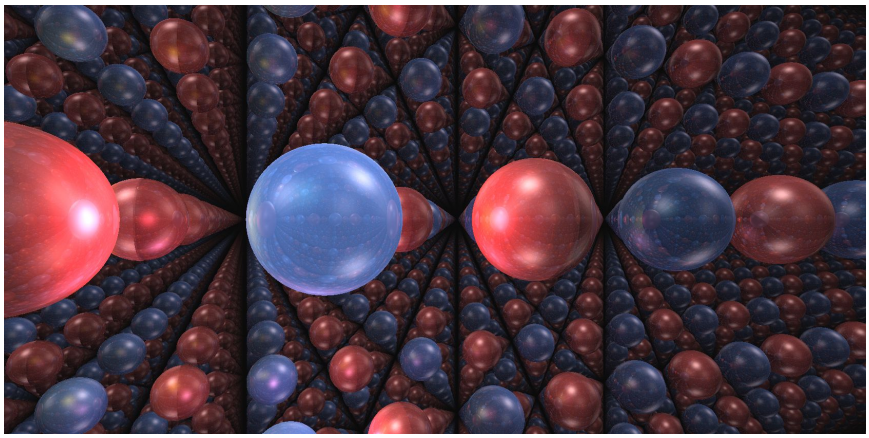
Here r is a sufficiently large radius so that the deleted ball opens windows into neighboring fundamental domains. The corresponding



(A) Signed distance function restricted to D. Note the striped artifacts in various copies of the red ball.



(B) Creeping to the boundary of D. The striped artifacts are gone, but we can see only half of the red ball.



(C) Using a nearest neighbors signed distance function, without creeping.

FIGURE 4.5. Difficulties when ray-marching in a fundamental domain D. The blue sphere is contained fully in D. The red sphere is only half contained in D.

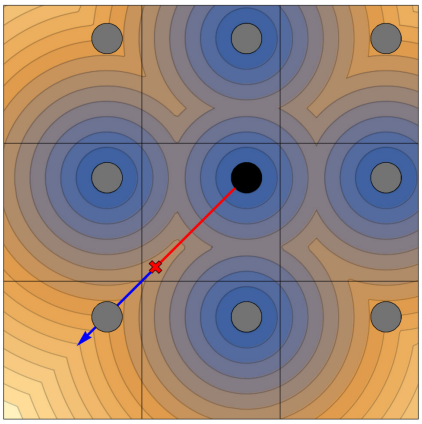


FIGURE 4.6. For rays traveling near to a vertex, only using the nearest neighbors of a tile may not be enough to remove all visual artifacts without creeping.

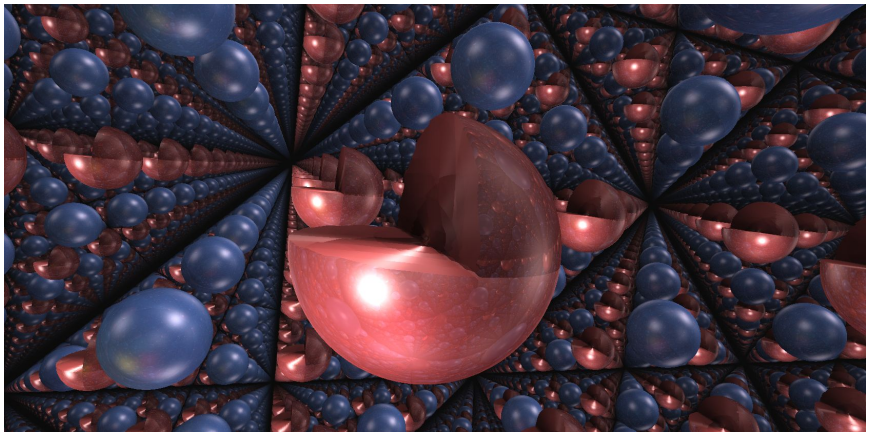


FIGURE 4.7. Even combining creeping to the boundary with nearest neighbors may not fix all problems. Here the scene consists of a ball that overlaps an edge of a cubical domain D.

tile for the cubic lattice in $\mathbb{E}^3$ is shown in Figure 2.1b. Depending on the geometry, we may also remove a sphere centered at each vertex of the fundamental domain, as in Figure 2.1c. ◊

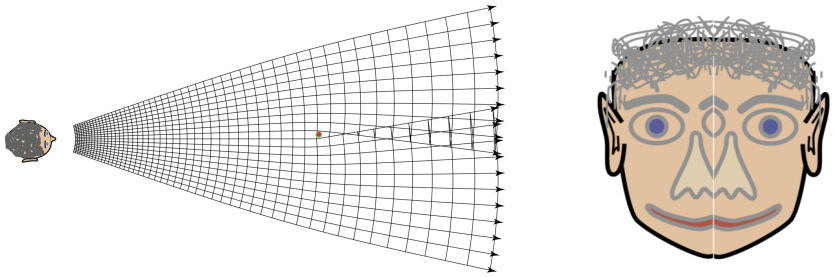
4.3. **Orbifolds and incomplete structures.** In our discussion so far we have assumed that X/Γ is a manifold, but in fact nothing is lost by generalizing to orbifolds. Briefly, an orbifold is a topological space locally modeled on patches of $\mathbb{R}^n/G$ for G some finite group of diffeomorphisms. When G is the trivial group, this reduces to the definition of a manifold. This additional flexibility in the definition

allows for certain controllable singularities, such as cone axes (with cone angle π/k for some integer $k > 0$), while still behaving very similarly to the manifold case. Indeed, many topological notions such as fundamental groups, covering spaces, and geometric structures carry over directly to orbifolds. Geometric structures on orbifolds are defined similarly to those on manifolds (see the beginning of Section 4), with the main difference being that the action of the fundamental group under the holonomy homomorphism need not be free. However, as the image Γ of the holonomy homomorphism is still discrete, we may find a fundamental domain D for its action and draw pictures of the quotient orbifold X/Γ as before. There is however little change in visual effect: by [CHK00, Corollary 2.27], every orbifold with a (G, X) structure is finitely covered by some (G, X) manifold. Thus, up to a finite amount of local information in the scene, the large scale picture will look the same as its manifold cover.

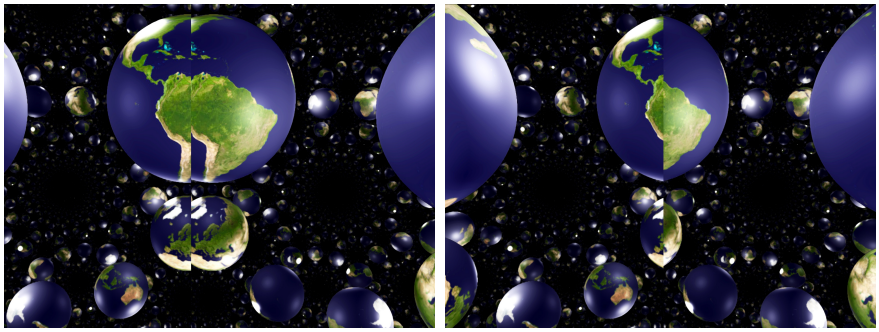
We can generalize still further. Manifolds and orbifolds have complete geometric structures, meaning that the developing map is a diffeomorphism. This allows the identification $M \cong X/\Gamma$. The more general notion of *incomplete* (G, X)-manifolds are also fundamental objects in geometric topology. Allowing general immersions as developing maps $\widetilde{M} \to X$ naturally captures various kinds of singularities, such as cone axes (where the cone angle can now be any real number) or punctures. This sort of flexibility is crucial in some core results of geometric topology. For example, the natural extension of the Geometrization Theorem to orbifolds requires the analysis of incomplete hyperbolic structures. However, incomplete structures are typically difficult to deal with, as the image of the holonomy homomorphism is indiscrete. Previous work here includes hand-drawn examples by Thurston (including two-dimensional structures in chapter three of [Thu97], and a three-dimensional drawing reproduced here in Figure 4.8a from [Thu98]) and tilings of $\mathbb{H}^2$ by Bonahon [Bon09].

Our ray-marching procedure for quotient manifolds extends without change to incomplete structures, allowing the accurate rendering of these as well. Note that throughout the algorithm, only local data is required: the existence of a fundamental domain D and face pairings $\{\gamma_i^{\pm}\}$. Both of these exist equally well for incomplete structures. Here the inside view is quite different than the complete case. The ability to render incomplete structures may aid in visualization projects, such as animating hyperbolic Dehn surgery or geometric transitions. Indeed, version 2.8 of SnapPy [CDGW] implements the inside view of hyperbolic manifolds undergoing hyperbolic Dehn surgery. However, interpreting

these requires more mathematical sophistication than for more familiar manifolds and orbifolds, so we will not focus on them in this paper.



(A) A cone axis of angle $2\pi - \varepsilon$ causes double images. These images are Figures 1 and 3 in Thurston's paper *How to See Three Manifolds* [Thu98].



(B) Hyperbolic cone manifold with cone axis of angle $2\pi - \varepsilon$.

(C) Hyperbolic cone manifold with cone axis of angle $2\pi + \varepsilon$.

FIGURE 4.8. The inside view of a manifold with a cone axis has double imaging of some points when the cone angle is slightly less than 2π, and hidden regions when the cone angle is slightly greater than 2π.

Remark 4.5. We create some of our spaces by directly constructing a fundamental domain D, then later figure out which manifold, orbifold, or incomplete manifold it is. In other cases, we start with a desired manifold, or lattice $\Gamma < G$, and have to work out a fundamental domain D. For the easier geometries, this generally involves (spherical, hyperbolic, or vanilla) trigonometry. We discuss the construction of fundamental domains for the harder geometries in Sections 9.9, 10.9, and 11.7. ◊

5. LIGHTING

Common physics-based shading techniques in computer graphics (diffuse and specular lighting, reflections, shadows, ambient occlusion,

and atmospheric effects) are all computed from geometric data, and so generalize naturally to riemannian geometry. Below we briefly review some of these techniques, and the modifications required.

The effect from each light source in the scene can be computed separately, and the final color determined through a weighted (by intensity) average of each light's contribution. Thus it suffices to describe the contribution of a single light source. However, in the geometries with positive sectional curvatures ($S^3, S^2 \times \mathbb{E}$, Nil, Sol, $\widetilde{\mathrm{SL}}(2,\mathbb{R})$), non-uniqueness of geodesics may cause even a single light source to illuminate an object from multiple directions. As these individual contributions also combine linearly to the total, we may further reduce the problem to understanding single-source lighting from a single direction at a time.

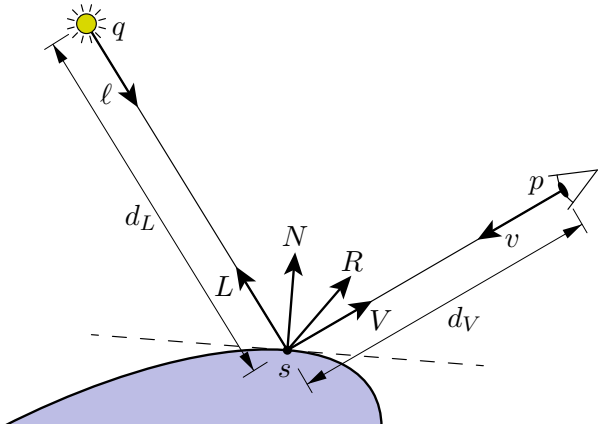


FIGURE 5.1. The geometric data required to calculate the color observed when looking from the point p in the direction $v \in T_pX$ at a point s, lit by a light at a point q from the direction $L \in T_sX$.

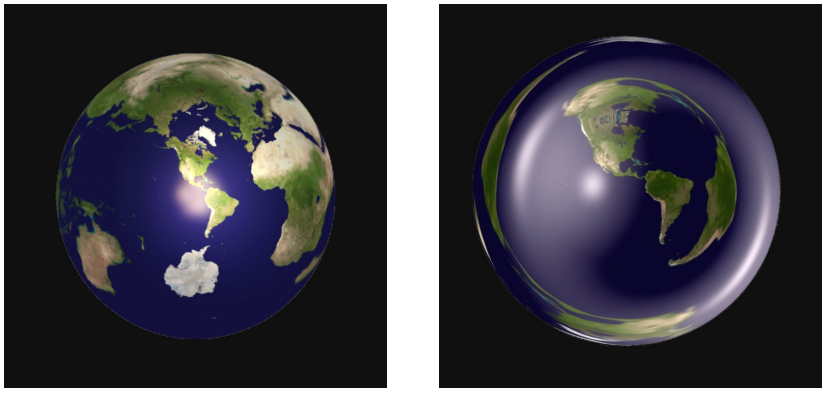
To fix notation, let S be a scene in X given by a signed distance function σ, lit by a light source at $q \in X$. See Figure 5.1. Let C_s be the base color of the point s of the scene, (represented as a three-vector storing its red-green-blue components), let C_{light} be the color of light source, and I_{light} be its intensity. Now suppose that we are at a point $p \in X$, looking in the direction $v \in T_pX$. Assume that this line of sight ends by impacting the point $s \in S$ of the scene. To compute the aforementioned lighting effects, we need the following data:

- $N \in T_sX$: unit outwards normal to ∂S at s,
- $L \in T_sX$: unit vector at s pointing to q,
- $R \in T_sX$: reflection of $-L$ with respect to N,
- $V \in T_sX$: unit vector at s pointing to p,
- $v \in T_pX$: unit vector at p pointing to s,

- $\ell \in T_q X$: unit vector at q pointing to s,
- d_L: distance from s to q along the geodesic with tangent L,
- d_V: distance from s to p along the geodesic with tangent V, and
- I_L: the light intensity experienced at s from the direction L.

Here we employ the convention that vectors in the tangent space at s are written in upper case, while vectors in tangent spaces at other points are written in lower case.

Remark 5.1. The base colour C_s for a point s of the scene can be a single colour for each object, or we can texture objects in a more complicated way. For example, we sometimes texture balls as the Earth. This provides a globally recognized coordinate system and allows one to infer the final endpoints of geodesics leaving your eye. See Figure 5.2. ◊



(A) A ball in spherical geometry: more than half of its surface is visible.

(B) A ball in Nil geometry: the non-uniqueness of geodesics causes a triple image of South America.

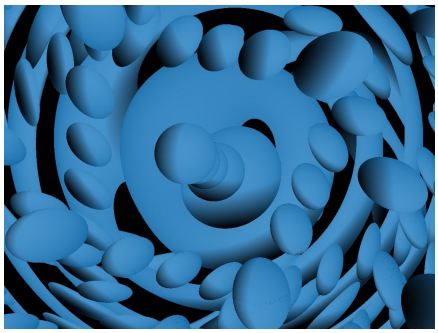
FIGURE 5.2. Balls textured as the Earth.

5.1. **Phong lighting model.** An empirical formula for accurate diffuse and specular reflection in computer graphics was published by Phong in his 1975 dissertation [Pho75] and now bears his name. The *Phong lighting model* (also called the Phong reflection model) decomposes the total color of the surface as a sum of three components: *ambient, diffuse* and *specular*. The ambient contribution is simply the base color C_s of the object at s. The remaining two terms are proportional to the light color C_{light} and the intensity I_L of the light source, as well as a third geometric quantity, as follows. Diffuse lighting is also proportional to the cosine of the angle between the light direction and the surface

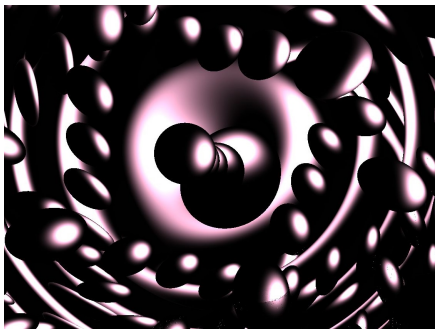
normal. Specular reflection is proportional to some power of the cosine of the angle between the viewer and reflected ray directions. This power is a parameter controlling the "shininess" of the material of the object. When either of these angles is obtuse, the corresponding lighting contribution is taken to be zero. This allows us to express the total lighting contribution of Phong lighting using the riemannian metric at s:

(5.2)
$$\text{Phong}(N, L, R, V, I_L) = k_{\text{amb}} C_s + \big(k_{\text{diff}} \langle N, L\rangle + k_{\text{spec}} \langle R, V\rangle^{\alpha}\big) I_L C_{\text{light}},$$

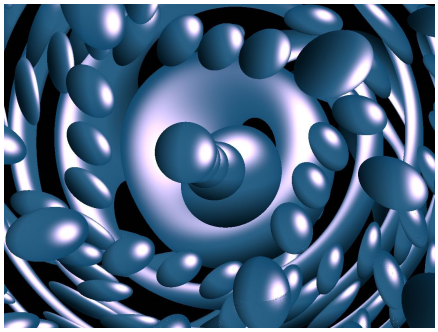
where the constants are chosen to satisfy $k_{\text{amb}} + k_{\text{diff}} + k_{\text{spec}} = 1$. These control the relative contribution of each of these factors.



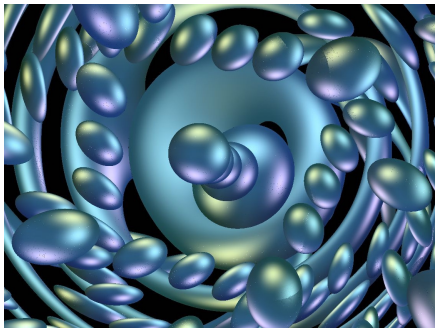
(A) Diffuse lighting.



(B) Specular highlights.



(C) Phong model: ambient, diffuse, and specular.



(D) Phong lighting with multiple light sources provides realistic depth cues.

FIGURE 5.3. A collection of balls in Nil geometry.

Remark 5.3. Phong justifies his model empirically, by comparing a render with a real-life photograph of a (euclidean) scene. We use his model far outside of the setting in which it was designed for, so one

could question whether or not it produces accurate results in our non-euclidean spaces. A reasonable test would be to compare our results with a more physically correct ray-tracer. ♢

5.2. **Shadows.** Phong lighting calculates the contribution of the observed color at s due to a light source in the direction L using only local computations in T_sX. While efficient, this ignores the existence of other objects in the scene, effectively rendering them transparent to the lighting calculation.

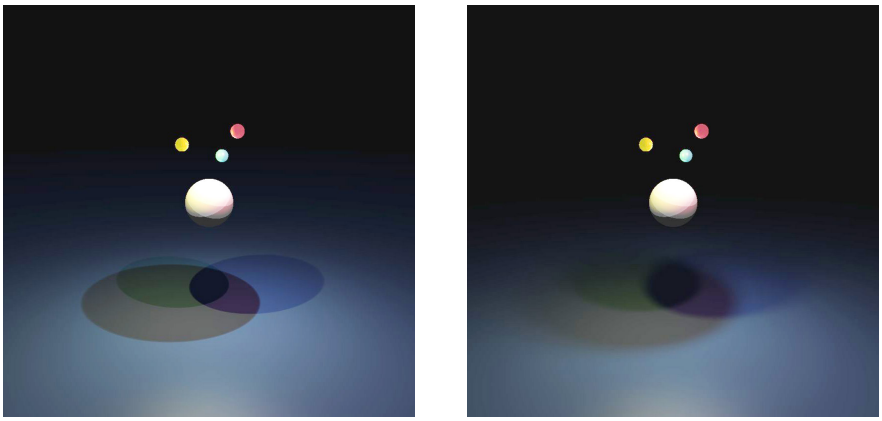
Happily there is a simple solution to detecting objects which block the path from s to the light: simply ray-march starting at s in the direction towards the light and see if you hit anything. If you do then there is no need to calculate the Phong lighting contribution for that light/direction, as s is in shadow. When modeling lights as point sources, this produces *hard* shadows. Realistic light sources which emit light over an area instead produce *soft* shadows, as there are points in space where the light source is only partially obscured. While modeling an extended source is computationally demanding, a multitude of empirical formulas for approximating soft shadows with point source lights have been developed in computer graphics. We briefly discuss a solution particularly well suited for ray-marching below. See [Quib] for more details.

Instead of a simple binary value, the shadow is modeled as a scaling factor to be multiplied by the Phong lighting contribution, smoothly interpolating between zero and one. To compute this value, we track the distance of the light ray from other objects in the scene as we follow it backwards from s in the direction L. Let $\gamma\colon [0,T] \to X$ be the arc length parametrized geodesic from s to the light at q with initial tangent L. The degree of shadow imparted by the surrounding scene at a point $\gamma(t)$ is modeled by the distance of $\gamma(t)$ from an object in the scene, normalized by the distance traveled from s. The total degree of shadow is proportional to the minimal value of this ratio over the path, or

$$\text{Shadow}(s,L) = \min\left\{1, K\frac{\sigma(\gamma(t))}{t} : t \in [0,T]\right\}. \tag{5.4}$$

Here $K \geqslant 1$ is a parameter determining softness. As $K \to \infty$ this reproduces the hard shadows above. In practice, we approximate this by computing this ratio at each step of the ray-march from s to q, and then take the minimum.

5.3. **Atmospheric Effects.** The fact that computing the total distance traveled along a path is trivial in a ray-marching application makes the



(A) Hard Shadow. (B) Soft Shadow ($K = 5$).

FIGURE 5.4. A comparison of different shadow rendering techniques with a sphere lit by three light sources above a plane in euclidean space.

above soft shadow approximation efficient. This almost free availability of path lengths also lends itself well to volumetric rendering: accounting for contributions to the lighting from atmospheric media encountered along the path. The simplest such effect, *distance fog*, is computationally inexpensive to implement and provides helpful distance cues in complex scenes. This replaces a fraction of the color of a pixel with a "fog" color, C_{fog}, depending on the distance the ray travels before hitting an object.

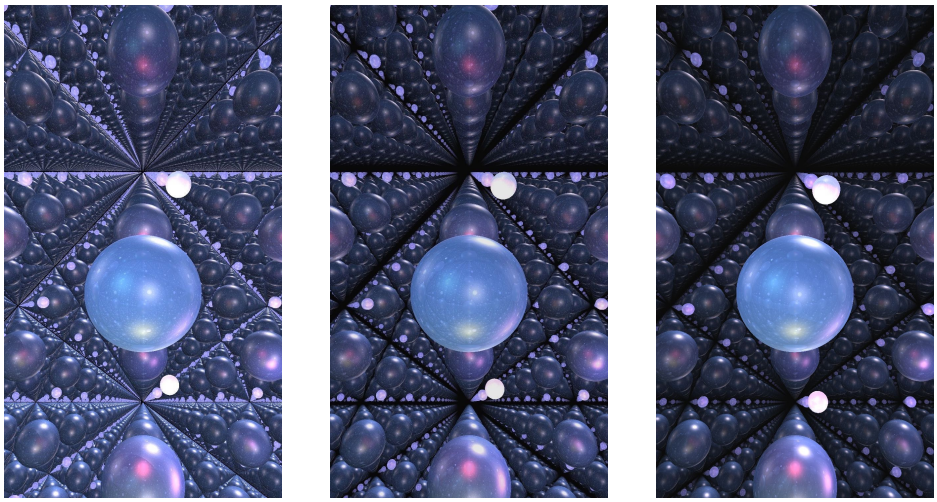
In many computer graphics applications, this fraction is linear in path length. This has the advantage that there is a distance at which all of the pixel is given the fog color, and no further calculation is necessary. However, a physically realistic model based on scattering along a path (the Beer-Lambert law in physics) implies that the fraction is actually exponential in the path length. We give these two models below.

$$\text{Fog}(d_V) = 1 - \min\left\{\frac{d_V}{K}, 1\right\}, \qquad \text{Fog}(d_V) = e^{-Kd_V} \tag{5.5}$$

Here $K > 0$ is a constant determining the rate of scattering. Each of these are extremely easy to implement, as they are standard functions of the already-available path length.

Combining the contributions from both shadows and fog, we obtain the following.

$$\begin{aligned}\text{Col} = {} & \text{Fog}(d_V) \cdot \text{Shadow}(L) \cdot \text{Phong}(N, L, R, V, I_L) + \\ & (1 - \text{Fog}(d_V)) \cdot C_{\text{fog}}\end{aligned} \tag{5.6}$$



(A) Without fog. (B) With linear fog. (C) With exponential fog.

FIGURE 5.5. A lattice of balls in euclidean space.

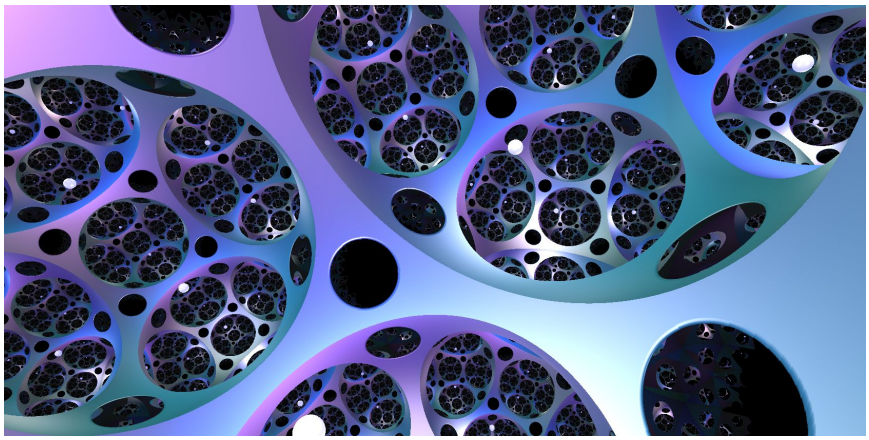
Outside of this section, our in-space images use exponential fog unless otherwise noted. We always set C_{fog} to be black.

5.4. **Reflections.** It is also relatively simple to allow for reflective materials in ray-marching, Upon impacting a reflective surface at s_1, one simply initiates a new ray-march from s_1 in the direction of the reflected ray. This ray-march may impact another object, at s_2 say. If so, we may reflect again. Computing the observed colors Col_i at the points s_i as above, the final color is an average, weighted by the reflectivity $r_i \in [0, 1]$ of the material at s_i. This can be carried out iteratively with no additional difficulty (other than increase in computation time). The weighted averages for one and two reflections are given below.

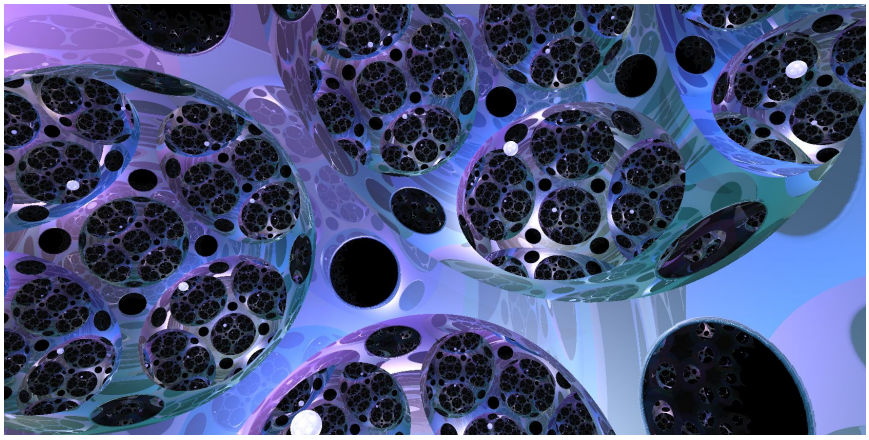
$$(1 - r_1)\text{Col}_1 + r_1\text{Col}_2 \qquad (1 - r_1)\text{Col}_1 + r_1((1 - r_2)\text{Col}_2 + r_2\text{Col}_3)$$

5.5. **Computing the necessary geometric quantities.** As the above sections illustrate, it is relatively straightforward to calculate accurate lighting, given the geometric quantities listed at the beginning of this section. Here we turn to the issues involved in computing these. Some of these quantities are available directly from the ray-march itself.

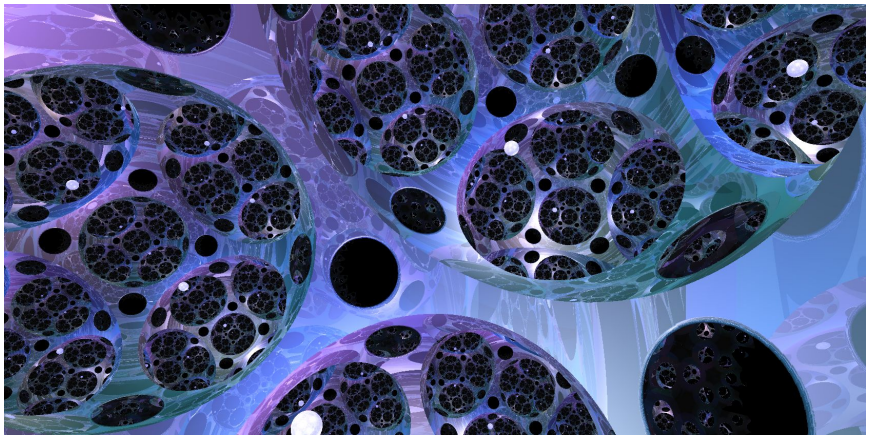
5.5.1. *Computing v.* The vector $v \in T_pX$ pointing from the viewer to the observed point $s \in S$ is the initial tangent vector for the ray-march.



(A) No reflections.



(B) A single reflection pass.



(C) Two reflection passes.

FIGURE 5.6. Reflections in a complicated scene in hyperbolic space.

5.5.2. *Computing V.* The vector $V \in T_sX$ pointing back at the viewer is the negation of the final tangent vector for the ray-march.

5.5.3. *Computing d_V.* The distance d_V from the viewer to the observed point is the path length returned by the ray-march.

Other quantities require further computation.

5.5.4. *Computing N.* The unit surface normal $N \in T_sX$ is computable directly from the signed distance function σ. It is the gradient vector $\operatorname{grad}\sigma(s)$ dual to $d_s\sigma$ via the riemannian metric. As in multivariable calculus, fixing a basis $\{f_1, f_2, f_3\}$ for T_sX, this is approximated for some small $\varepsilon > 0$ by

$$\operatorname{grad}\sigma(s) \simeq \sum_{i=1}^{3} \frac{\sigma(s+\varepsilon f_i) - \sigma(s-\varepsilon f_i)}{2\varepsilon} f_i$$

While in principle any choice of basis of T_sX suffices, even slight discontinuities in the normal field over a surface are plainly visible in the output of the Phong lighting model. To prevent this source of error, we make a globally continuous choice of basis by selecting a section of the frame bundle. A simple construction of such a section follows from the transitivity of the G-action. Let $B \subset G$ be a subset (not necessarily a subgroup) of the isometry group such that the orbit map $B \to X$ defined by $g \mapsto g.o$ is a diffeomorphism. (for example, when G has a subgroup acting simply transitively, we may take this as B). The inverse of this orbit map provides a section $X \to G$ with image B, sending $s \in X$ to $g(s)$. We promote this to a section of $\mathcal{O}X$ by choosing an orthonormal frame $f = \{f_1, f_2, f_3\}$ for T_oX and translating by the G-action. This assigns to $s \in X$ the frame $d_o g(s) f$.

5.5.5. *Computing R.* The unit normal provides a means of reflecting rays in the surface. Given any vector $U \in T_sX$ we may compute its reflection in the surface by

$$\operatorname{Refl}(U) = U - 2\langle U, N\rangle N$$

Thus, given the direction to the light source $L \in T_sX$, we may find the final direction needed for Phong lighting, $R = -\operatorname{Refl}(L)$. This leaves only four quantities to be computed, all dealing with the location of the light source; two directions L, ℓ and two scalars d_L, I_L. These require global information about the geometry of X. We discuss this next.

5.6. **Computing lighting directions, L, ℓ, and distance d_L.** Calculating the direction L in which a light is visible from a point on the surface (and the other related quantities) cannot be reduced to linear algebra in some tangent space: it involves the global geometry of X. This requires a procedure that takes two points $s, q \in X$ and returns the set of *lighting pairs* $\mathcal{L}_s(q) \subset T_sX \times \mathbb{R}_+$. Here each element $(L, d_L) \in \mathcal{L}_s(q)$ represents the direction, L, of a geodesic γ connecting s to q, and the length, d_L, of the geodesic segment γ connecting s to q. Since we use explicit formulas for the geodesic flow, one can directly compute from (L, d_L) the direction $\ell \in T_qX$ and the reverse geodesic γ' joining q to s. In all cases, we may use the homogeneity of X to reduce the problem to understanding geodesics from the origin, and focus on calculating the lighting pairs $\mathcal{L}_o(q)$ for $q \in X$.

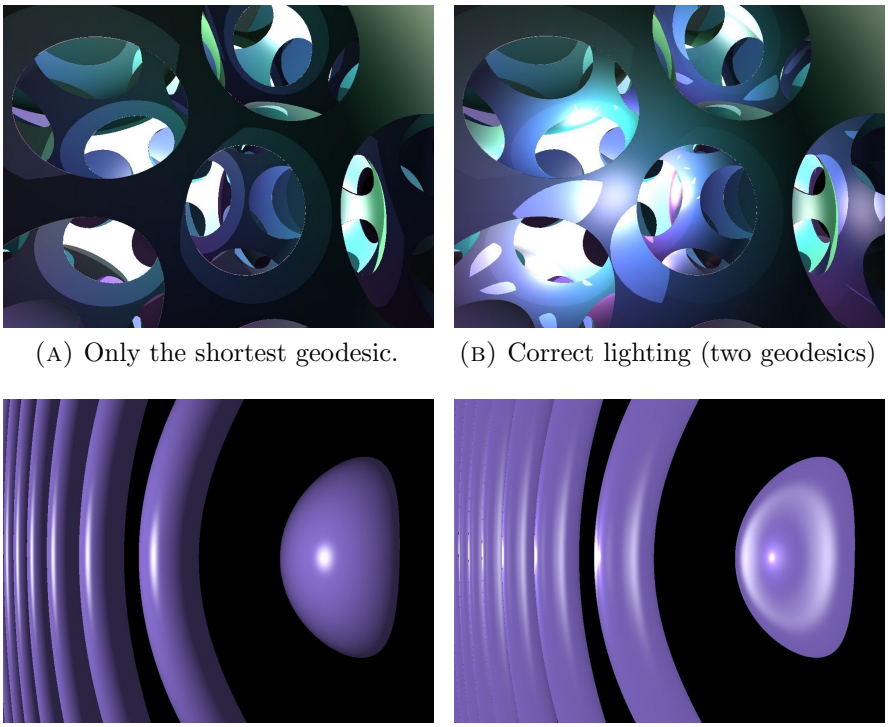
In geometries with nonpositive sectional curvature, geodesics are unique by Cartan-Hadamard. Thus for each $q \in X$ the set $\mathcal{L}_o(q)$ is a singleton. In other geometries $\mathcal{L}_o(q)$ may be a singleton, finite, countably infinite, or uncountably infinite, depending on q. See Figure 5.7 for examples of lighting along multiple geodesics in S^3 and $S^2 \times \mathbb{E}$. There is no uniform approach to calculate $\mathcal{L}_o(q)$, so we deal with this computation in later, geometry-dependent sections of this paper.

5.7. **Computing the light intensity I_L.** We have one remaining quantity to compute: I_L, the intensity of the light source at q, as observed at s from direction L. We model our light source as isotropic with constant intensity I_{light}. To fix some notation, for any distance $t > 0$ and unit direction vector $u \in T_qX$, let $I(t, u)$ be the intensity arriving from the light source after traveling along the geodesic ray in the direction u for distance t. For any solid angle $\Omega \subset T_qX$, let $\Omega_t \subset X$ be the surface formed by flowing outwards from q along geodesics in the directions in Ω by distance t. See Figure 5.8.

We assume that the total energy flux through the surface Ω_t is constant, independent of the distance traveled. (Energy is transported by the light rays along geodesics, but not created or destroyed along the way.) This relates $I(t, u)$ directly to the area density of geodesic spheres. That is, for any Ω, t we have

$$\int_\Omega I_{\text{light}} dA = \int_\Omega I(t, u) dA'$$

where dA is the standard area form on the unit sphere in the tangent space, and dA' is the pullback of the area form on $\Omega_t \subset X$ to $\Omega \subset T_qX$. We may express dA' in terms of dA; the resulting scale factor is the area density $dA' = \mathcal{A}(t, u)dA$. Thus, the quantity $\int_\Omega I(t, u)\mathcal{A}(t, u)dA$



(A) Only the shortest geodesic. (B) Correct lighting (two geodesics)

(C) Only the shortest geodesic. (D) 200 Geodesics.

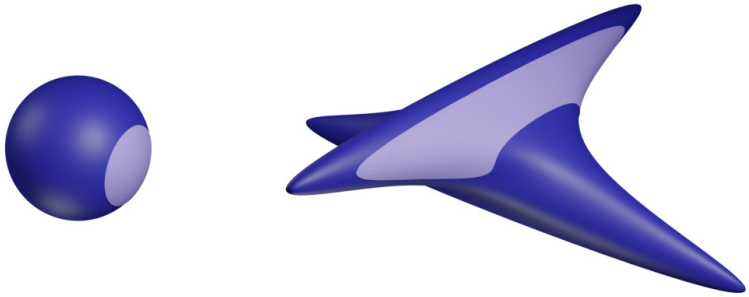
FIGURE 5.7. A single light in S^3 (top) and $S^2 \times \mathbb{E}$ (bottom). This demonstrates the necessity of dealing with multiple directions in $\mathcal{L}_p(q)$.

is constant in r for every solid angle $\Omega \subset T_qX$. Assuming continuity and taking the limit over shrinking solid angles promotes this to a pointwise invariant: $I(t,u)\mathcal{A}(t,u)$ is independent of t. Thus, I is inversely proportional to $\mathcal{A}$, and

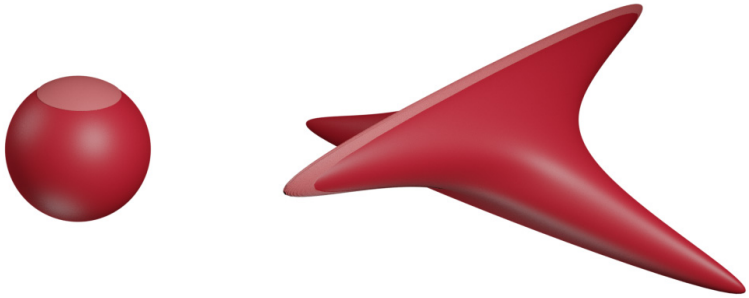
$$I(t,u) = \frac{I_{\text{light}}}{\mathcal{A}(t,u)}. \tag{5.7}$$

Remark 5.8. The intensity I_L experienced at s from the direction L is then just $I_L = I(d_L, \ell) = I_{\text{light}}/\mathcal{A}(d_L, \ell)$. A further correction to I_L can occur when we add fog. Here the intensity drops off due both to (1) divergence/convergence of geodesics, and (2) distance traveled through the medium. A physically correct model for scattering from an isotropic source is already complex in euclidean space. However, as the primary goal of modeling fog is to provide useful depth cues (and hide sins), we treat these sources of loss as if they were independent, and use

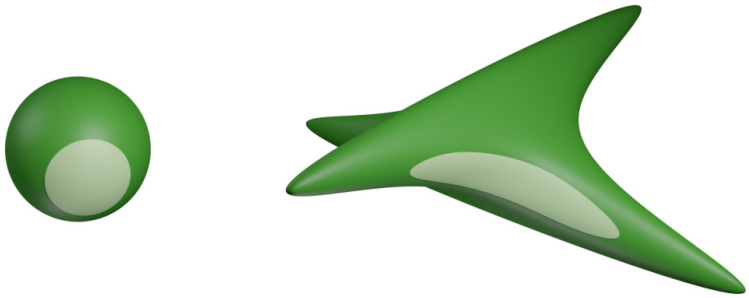
$$I_L^{\text{fog}} = \text{Fog}(d_L) \cdot I_L(d_L, \ell) = e^{-Kd_L} \frac{I_{\text{light}}}{\mathcal{A}(d_L, \ell)}$$



(A) Solid angle around the x-axis.



(B) Solid angle around the z-axis. The image of the lighter area is a tiny strip on the top of the Sol sphere.



(C) Solid angle around a diagonal line

FIGURE 5.8. Extrinsic views of spheres in Sol. In each figure, the left hand picture represents the unit sphere in the tangent space at the origin of Sol. The lighter areas correspond to solid angles Ω with the same measure, but pointing in different directions. The right hand picture shows an extrinsic view of the image of the unit tangent sphere after following the geodesic flow for time $r = 3$. The lighter area is the image Ω_t of Ω.

when distance-dependent attenuation (fog) is desired. ◊

Equation (5.7) reduces the calculation of lighting intensity directly to the area density $\mathcal{A}$. In the next section, we calculate this area density by following infinitesimal patches of area along the geodesic flow.

5.7.1. *Area density under the geodesic flow.* Fix $q \in X$ to be the location of a light source, and let $F\colon T_qX \to X$ be the exponential map. For fixed $t > 0$, define $f_t(u) = F(tu)$, so $f_t\colon S^2 \to X$ is a map of the unit sphere $S^2 \subset T_qX$ into X, formed by flowing along geodesics from q for distance t. Note that the image is not the sphere of radius t about q when t is greater than the injectivity radius of X. Recalling the notation above Ω_t is defined as $f_t(\Omega)$, for a solid angle $\Omega \subset S^2$. Consistent with this, we denote the entire image as $S^2_t = f_t(S^2)$. Let dA be the standard area form on $S^2 \subset T_qX$, and let dA_t be the area form on $S^2_t \subset X$. Recall that the area density $\mathcal{A}(t,u)$ is the proportionality factor of the pullback $f_t^* dA_t$ to dA. We may compute this given any choice two non-collinear vectors $\{v, w\}$ in $u^\perp$ as

$$\mathcal{A}(t,u) = \frac{(f_t^* dA_t)(v,w)}{dA(v,w)} = \frac{dA_t((df_t)_u v, (df_t)_u w)}{dA(v,w)}.$$

The area forms dA and dA_t measure the areas in X of infinitesimal parallelograms in T_qX and $T_{f_t(u)}X$ respectively, and so may be evaluated in the algebra of bivectors on TX, where the area spanned by $v, w \in T_pX$ is given by

$$\|v \wedge w\| = \sqrt{\langle v, v\rangle\langle w, w\rangle - \langle v, w\rangle^2}$$

Thus, we have

$$\mathcal{A}(t,u) = \frac{\|(df_t)_u v \wedge (df_t)_u w\|}{\|u \wedge w\|}. \tag{5.9}$$

As computing area elements requires nothing more than some evaluations of the metric, this reduces the calculation of area density to the computation of the differential df_t.

Recall that $f_t(u) = F(tu)$, where F is the exponential map. We see that $(df_t)_u v = dF_{tu} v$ for all $u \in T_qX$ and $v \in T_u(T_qX)$. To lighten notation, for the rest of this paragraph we identify $T_{u'}(T_qX)$ with T_qX for every $u' \in T_qX$. Given u in $S^2 \subset T_qX$ and v in $u^\perp$ of unit length, this allows an explicit computation of $(df_t)_u v$ in terms of the exponential map, as follows. Let $\eta(v,s) = \cos(s)u + \sin(s)v$ be the unit vector in T_qX making angle s with u in the plane spanned by $\{u, v\}$. Note that $\eta'(0) = v$ so we may calculate $(df_t)_u v$ as

$$(df_t)_u v = dF_{tu} v = \left.\frac{d}{ds}\right|_{s=0} F(t\eta(v,s)).$$

For each fixed s, the map $t \mapsto F(t\eta(v,s))$ is a unit speed geodesic in X, and the derivative $dF_{tu}v \in T_{F(tu)}X$ is a vector field along this geodesic. Computed as above, we see this is a particularly nice vector field: it is the derivative of the geodesic flow along a one-parameter family of geodesics. Such vector fields are called *Jacobi fields*.

Given a smooth one-parameter family of geodesics $\{\gamma_s(t)\}$ through $\gamma_0 = \gamma$, the *Jacobi field* associated to γ_s is given by $J(t) = \partial\gamma_s(t)/\partial s|_{s=0}$. In general, one may bypass explicit computations involving γ_s, and compute such Jacobi fields by solving a differential equation. The Jacobi field J_v along γ with initial conditions $J(0) = 0, \dot{J}(0) = v$ satisfies the so called *Jacobi equation*,

$$\ddot{J}_v = \mathfrak{R}(J_v, \dot{\gamma})\dot{\gamma} \tag{5.10}$$

where $\mathfrak{R}$ is the Riemann curvature tensor. For us then, $(df_t)_u v$ and $(df_t)_u w$ are the Jacobi fields along $f_t(u)$ corresponding to the variations $F(t\eta(v,s))$ and $F(t\eta(w,s))$ respectively, so

$$(df_t)_u v = J_v(t) \quad \text{and} \quad (df_t)_u w = J_w(t). \tag{5.11}$$

In the isotropic geometries and product geometries, Equation (5.10) reduces to a second-order differential equation with constant coefficients. In any geometry where one may solve Equation (5.10), the area density is given as follows. For fixed $u \in S^2$, choose two vectors $v, w \in u^\perp$ with $\|v \wedge w\| = 1$ and solve the Jacobi equation for the two Jacobi fields J_v, J_w. Then using Equations 5.9 and 5.11, we have

$$\mathcal{A}(t,u) = \|J_v(t) \wedge J_w(t)\| \tag{5.12}$$

In the harder geometries, solving Equation (5.10) is more challenging. Following Section 3.2.1, one could use Grayson's method to replace Equation (5.10) with a system of differential equations on T_oX. However, we already use Grayson's method to to compute the exponential map F. We then directly compute the differential dF_{tu}.

Let r, θ, ϕ be the standard spherical coordinates on T_qX, with ϕ the angle measured from the north pole. Let $u \in S^2$ have coordinates $[\theta, \phi]$. Note that as the coordinate vector fields $\partial_\theta, \partial_\phi$ are orthogonal to ∂_r, we may use them to make a uniform choice $v = \partial_\phi, w = \partial_\theta$, and compute

$$\mathcal{A}(r,u) = \frac{\|dF_{ru}(\partial_\phi) \wedge dF_{ru}(\partial_\theta)\|}{\|\partial_\phi \wedge \partial_\theta\|} = \frac{\|\frac{\partial F}{\partial \phi}(r,\theta,\phi) \wedge \frac{\partial F}{\partial \theta}(r,\theta,\phi)\|}{\sin\phi}.$$

In practice, due to the rotational symmetry in Nil and $\widetilde{\mathrm{SL}}(2,\mathbb{R})$ about a single axis, it is more convenient to perform this computation in

cylindrical coordinates, with $\rho = r\cos\phi$ and $z = r\sin\phi$. For ease of notation, we retain $r = \sqrt{\rho^2 + z^2}$ from spherical coordinates to denote the distance traveled along the geodesic.

$$\mathcal{A}(r,u) = \frac{2}{r}\left\| \left(\frac{\partial F}{\partial \rho} - \frac{\rho}{z}\frac{\partial F}{\partial z} \right) \wedge \frac{\partial F}{\partial \theta} \right\| \tag{5.13}$$

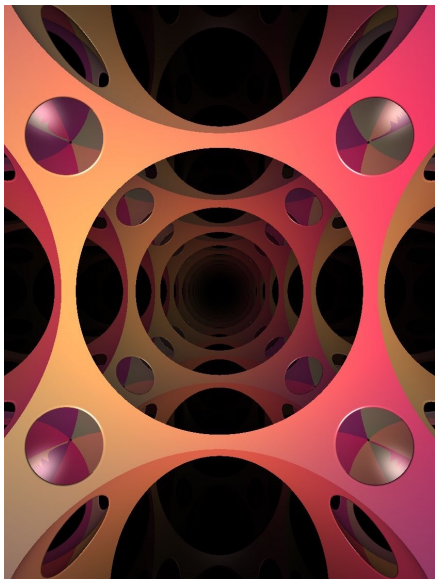
Using either Equation (5.12) or Equation (5.13), the computation of area density is necessarily geometry-dependent, so we give details for each geometry in the corresponding section later. See Sections 7.5, 8.4, 9.8, and 10.8.

5.8. **Lighting in quotient manifolds.** The basic algorithms for lighting remain virtually unchanged in a quotient manifold. Phong lighting is still computed in the tangent space, and the only modification to the computation of shadows and reflections is to modify the ray-march as in Section 4. There is only one major change worthy of discussion: the calculation of direction vectors pointing from the surface to a given light. This is even more necessarily multi-valued here, as light may travel in loops around the manifold before impacting the surface. Indeed, a light in X/Γ is the same as a Γ-equivariant collection of lights in X. When required for disambiguation, we will denote the set of lighting pairs in a space Y as $\mathcal{L}^Y$. For the location of a light q in D, thought of as the fundamental domain for X/Γ, the lighting pairs $\mathcal{L}_p^{X/\Gamma}(q)$ can be written in terms of the lighting pairs $\mathcal{L}_p^X$ of Section 5.6:

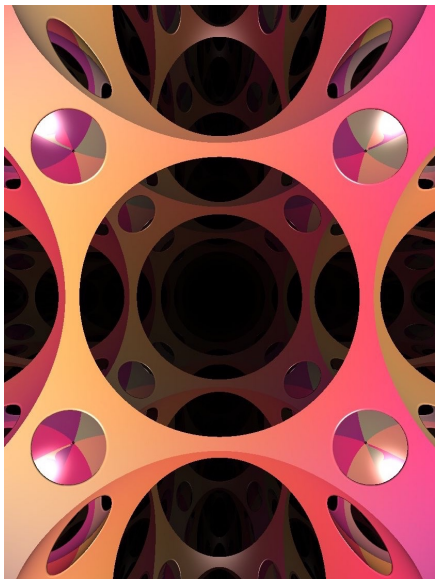
$$\mathcal{L}_p^{X/\Gamma}(q) := \bigcup_{\gamma\in\Gamma} \mathcal{L}_p^X(\gamma.q)$$

Note that there is no sense in which $\mathcal{L}_p^X(\gamma.q)$ is some sort of "γ-translate" of $\mathcal{L}_p^X(q)$: the individual sets in this union may not even have the same cardinality. This occurs for instance in Nil, where even if the distance from p to q is less than the injectivity radius, there may be a $\gamma \in \Gamma$ with arbitrarily many geodesics from p to $\gamma.q$. As lighting is calculated individually for each direction and summed weighted by intensity, it is in general impossible to compute this exactly for any manifold with infinite fundamental group. Instead, for all but spherical manifolds and orbifolds, we must approximate the lighting by computing only for those paths with significant intensity.

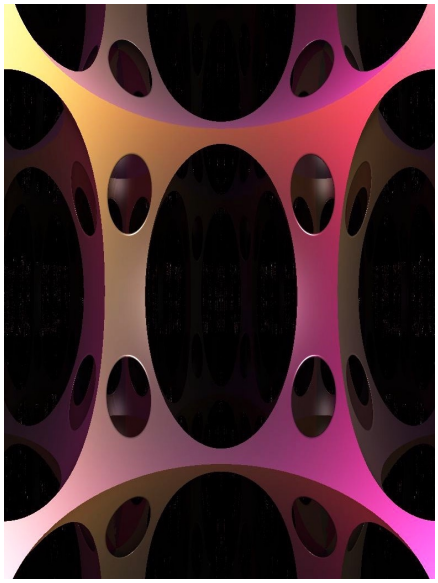
Light intensity is inversely correlated with geodesic length of a segment from p to q in geometries with non-positive sectional curvature, and in all geometries if we use fog. Thus we get a reasonable approximation to the correct image by restricting to directions corresponding to 'sufficiently



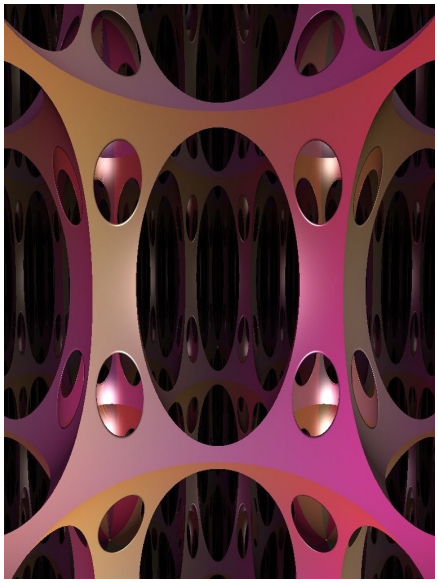
(A) Correct lighting, view in the $\mathbb{E}$ direction.



(B) Isotropic lighting, view in the $\mathbb{E}$ direction.



(C) Correct lighting, view in an $\mathbb{H}^2$ direction.



(D) Isotropic lighting, view in an $\mathbb{H}^2$ direction.

FIGURE 5.9. A lattice lit by a single light in $\mathbb{H}^2 \times \mathbb{E}$. In isotropic lighting, the intensity $I(r, u)$ is inversely proportional to the area of geodesic spheres. The distance between the centers of neighboring cells of the lattice is the same in all directions. With correct lighting, we see many cells in the $\mathbb{E}$ direction, while we can barely see our neighbor in an $\mathbb{H}^2$ direction. With isotropic lighting, cells dim with distance equally in all directions. (Note that there is no fog in these images.)

short' geodesics. Considering only the directions from lights within D (that is, when $\gamma = \mathrm{id}$) is not enough, as some nearby translates $\gamma.q$ still contribute significantly. Compare Figure 5.10a with Figure 5.10c. The latter shows the correct lighting in the quotient of the three-sphere by the binary tetrahedral group. The former shows lighting using one of the 24 light sources. An improved approximation is to use the 'nearest neighbors' idea from Section 4.2.2, and consider only tangent directions at p which reach the light at $q \in D$, or its translates *through the faces of* D. See Figure 5.10b.

This is even an issue in euclidean manifolds. Note that there is a discontinuity in the lighting of the red balls in Figure 4.5c. The left and right hemispheres are lit by different collections of lights, since they sit in different fundamental domains.

In geometries with positive sectional curvatures, light can converge again over long distances, meaning that there are certain directions where even long geodesics make significant contributions to the overall sum unless we use fog. Which translates of the lights to include in a calculation then depends on both the geometry and the scene. So far, we have only a heuristic understanding of how to choose translates appropriately, based on the light intensity function for each geometry.

5.9. **Cheating.** Accurate lighting and shading is a complex problem, requiring many calculations, and many ray-marches per pixel to perform correctly. As we strive to produce as accurate a simulation as possible, we have worked to implement lighting, shadows, reflections, and fog as described above. However, insistence on complete "physical" accuracy is not ideal for all applications. Sometimes lighting is best thought of as a means for euclidean humans to better perceive the geometry, rather than as a feature of the geometry in itself. This is analogous to astrophysical simulations, where it is more important to correctly render the size and position of celestial bodies, rather than to faithfully reproduce the brightness of the sun. In these situations it is often desirable to purposely employ nonphysical lighting to improve speed and/or visibility.

We find that the most often useful change to make is in the relationship of light intensity I_L with distance. There are two main problems that we can solve here.

- First, correct lighting may give intensities of vastly different magnitudes for different parts of the same scene. This means that parts of the scene will be too dark for our eyes to see any structure. Alternatively, we can increase the brightness of the lights, but then other parts of the scene will be oversaturated.

(A) Lighting from within D only.

(B) Lighting from within D and its eight neighbors.

(C) Lighting from all 24 cells.

FIGURE 5.10. Lighting of the quotient of S^3 by the binary tetrahedral group, with a single point source light. There are no reflections: the patterns are the result of (hard) shadows cast by the scene.

- Second, and more subtly, we use variation in brightness as a depth cue, telling us how far away an object is from a light source.

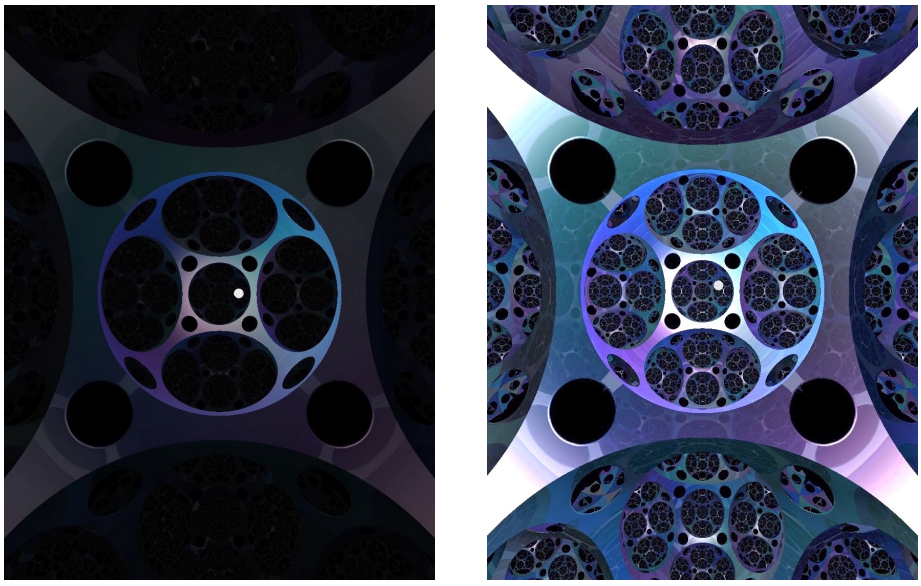
Figure 5.11a shows a scene in $\mathbb{H}^3$ lit by a single light. Here, exponential falloff in intensity with distance leaves everything other than the central cell shrouded in darkness. We see similar behavior in Figure 5.9c, when looking in a hyperbolic direction in $\mathbb{H}^2 \times \mathbb{E}$. When we look in a euclidean direction in Figure 5.9a, we do see neighboring cells, giving the impression that cells are closer in that direction than in the hyperbolic directions. In Figure 5.12a, the correct lighting calculations in $S^2 \times \mathbb{E}$ give an approximately even brightness over the whole image, even though only the ball at the center is particularly close to the viewer. The space $S^2 \times \mathbb{E}$ works like a fiber-optic cable – on average, the intensity of the light does not decrease with distance as we move along the cable.

Instead of the correct lighting intensity I_L, we may cheat, and use an artificial slowly decreasing intensity (say, inversely proportional to geodesic length). This provides more helpful depth cues and may also be less expensive to compute. See Figures 5.11b, 5.9b, 5.9d, and 5.12b. As a side benefit, this also allows one to see distant reaches of a negatively curved space with only a few light sources. This also reduces computational cost.

When it comes to improving speed, we may pare down the lighting pipeline to focus on giving accurate depth cues. This means preserving Phong lighting and fog, while perhaps ignoring shadows, or not using reflective materials. Another efficiency gain which does not affect the intelligibility of the scene is to consider only the direction to the light along the *shortest geodesic*, instead of the set of all directions. Even when attempting accurate rendering, it is often acceptable to ignore lighting along all but the shortest few geodesics. This is the case when using fog, or when the intensity fall-off makes the contribution to the weighted average along longer geodesics negligible.

However, using fewer geodesics can introduce very visible errors. In a quotient manifold, as we saw in Figures 5.10 and 4.5c we may lose shadows, or introduce discontinuities in the perceived light intensity. In some geometries, using fewer geodesics can in fact remove discontinuities in lighting intensity that should be there.

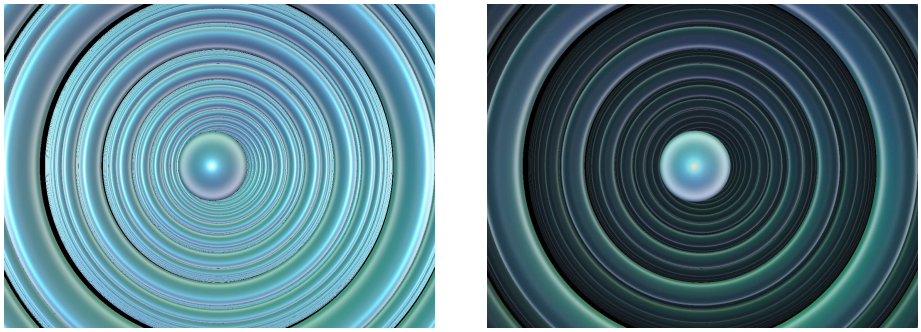
We usually indicate the position of a light with a ball in the scene centered on the light source, making sure that the shadow calculation for that light ignores the ball. To remove visual complication, we sometimes choose to not render these balls. Along these lines, in some situations



(A) Correct intensity calculation.

(B) Intensity inversely proportional to geodesic length.

FIGURE 5.11. A single light in hyperbolic space.

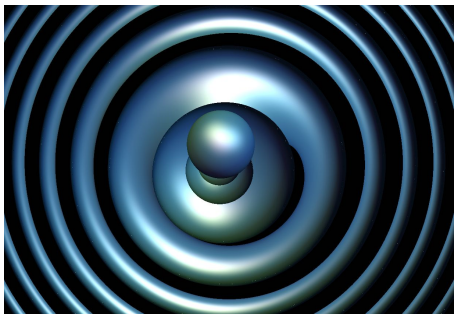


(A) Correct intensity calculation.

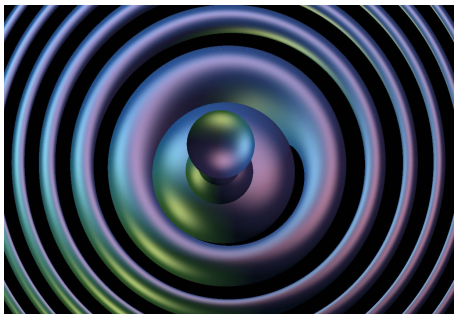
(B) Intensity inversely proportional to geodesic length.

FIGURE 5.12. A line of balls in $S^2 \times \mathbb{E}$ lit by a single light. Each ball is also visible as a collection of rings, seen along rays that wrap around the S^2 direction at least once.

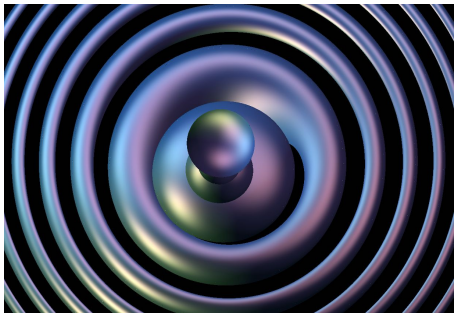
we may not actually care, or may not be able to efficiently calculate, the lighting pairs $\mathcal{L}_s(q)$. Instead, we may simply choose for each light source a continuously varying direction field $X \to TX$. We give up on correctness, but still provide a seamless view and give visual cues. Figure 5.13 compares different choices of illumination in Nil.



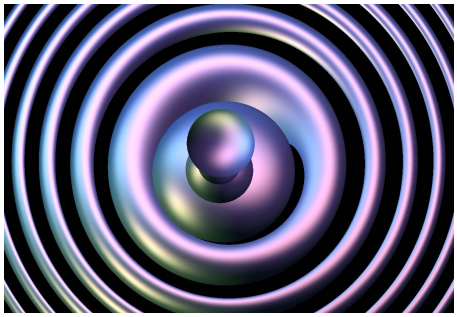
(A) Artificial direction field (straight line in $\mathbb{R}^4$ from $s \in X \subset \mathbb{R}^4$ to the light position).



(B) Direction of the shortest geodesic only.



(C) At most two geodesics.



(D) At most three geodesics.

FIGURE 5.13. A line of balls in Nil along the z-axis, lit by three light sources (cyan, yellow, and magenta). The magenta light is sufficiently far away from the first ball that they are connected by several geodesics. The intensity attenuation has been turned off to emphasis the contribution of each source of light.

6. Implementing specific geometries

In previous sections we have described our strategies in a more-or-less geometry independent manner. Here we begin to give specific details for each of the eight Thurston geometries. To summarize the previous sections, for each geometry, we require the following:

(1) A model for X with action of the group of isometries G. That is, we must now be explicit about how points and isometries are described by vectors or matrices of floating point numbers.
(2) Arc length parametrized geodesics in the model. That is, a way to flow a position and tangent vector at that position along the ray by a given distance, as described in Section 3.2.
(3) Signed distance functions in the model.

In order to render a quotient manifold with this geometry, we also need:

(4) A fundamental domain D with face pairings $\{\gamma_i\} \subset G$.

For the Phong reflection model of lighting, we need:

(5) For a point s (where a ray hits a surface) and the location of a light source q, the set of lighting pairs $\mathcal{L}_s(q)$ of geodesics joining s to q and vice versa. See Section 5.6.

To allow the user to move, we also require

(6) Parallel transport along geodesic arcs. (Used to translate movement of the user's frame in $\mathbb{R}^3$ into isometries of X.)

For each of the eight Thurston geometries, we list some of these ingredients in Table 1. All of our models are subsets of $\mathbb{R}^4$.

We give further details in the following sections. We consider the isotropic geometries in Section 7, the product geometries in Section 8, and Nil, $\widetilde{\mathrm{SL}}(2,\mathbb{R})$, and Sol in Sections 9, 10, and 11 respectively.

Geometry	**Model** (Set, Metric, Origin o)	**Geodesic from o in direction $\mathbf{v}$**	**Isometries**	**Example Lattices**
$\mathbb{E}^3$	$\mathbb{R}^4$, $w=1$, $ds^2=dx^2+dy^2+dz^2$, $o=\mathbf{e}_w$	$t\mathbf{v}$	$\mathbb{R}^3 \rtimes \mathrm{O}(3)$	$\mathbb{Z}^3$
S^3	$\mathbb{R}^4$, $x^2+y^2+z^2+w^2=1$ $ds^2=dx^2+dy^2+dz^2+dw^2$, $o=\mathbf{e}_w$	$\cos(t)\mathbf{e}_w+\sin(t)\mathbf{v}$	$\mathrm{O}(4)$	The eight element quaternion group.
$\mathbb{H}^3$	$\mathbb{R}^{3,1}$, $x^2+y^2+z^2-w^2=-1$ $ds^2=dx^2+dy^2+dz^2-dw^2$, $o=\mathbf{e}_w$	$\cosh(t)\mathbf{e}_w+\sinh(t)\mathbf{v}$	$\mathrm{O}(3,1)$	The isometry group of Seifert-Weber space.
$S^2\times\mathbb{E}$	$\mathbb{R}^3\times\mathbb{R}$, $x^2+y^2+z^2=1$ $ds^2=dx^2+dy^2+dz^2+dw^2$, $o=\mathbf{e}_z$	$\left(\cos(\lambda t)\mathbf{e}_w+\sin(\lambda t)\frac{\mathbf{v}_{S^2}}{\lambda}, s\mathbf{v}_{\mathbb{E}}\right)$ where $\mathbf{v}=(\mathbf{v}_{S^2},\mathbf{v}_{\mathbb{E}})$ and $\lambda=\lVert\mathbf{v}_{S^2}\rVert$	$\mathrm{O}(3)\times\mathrm{Isom}(\mathbb{R})$	$\Lambda\times\mathbb{Z}$ where Λ is a discrete subgroup of $\mathrm{Isom}(S^2)$
$\mathbb{H}^2\times\mathbb{E}$	$\mathbb{R}^{2,1}\times\mathbb{R}$, $x^2+y^2-z^2=-1$ $ds^2=dx^2+dy^2-dz^2+dw^2$, $o=\mathbf{e}_z$	$\left(\cosh(\lambda t)\mathbf{e}_w+\sinh(\lambda t)\frac{\mathbf{v}_{\mathbb{H}^2}}{\lambda}, s\mathbf{v}_{\mathbb{E}}\right)$ where $\mathbf{v}=(\mathbf{v}_{\mathbb{H}^2},\mathbf{v}_{\mathbb{E}})$ and $\lambda=\lVert\mathbf{v}_{\mathbb{H}^2}\rVert$	$\mathrm{O}(2,1)\times\mathrm{Isom}(\mathbb{R})$	$\Lambda\times\mathbb{Z}$ where Λ is a discrete subgroup of $\mathrm{Isom}(\mathbb{H}^2)$
Nil	$\mathbb{R}^4, w=1$, See Section 9.2, $o=\mathbf{e}_w$	See Section 9.3	$\mathrm{Nil}\rtimes\mathrm{O}(2)$	$\mathbb{Z}^2\rtimes_M\mathbb{Z}$ with $M\in\mathrm{SL}(2,\mathbb{Z})$, parabolic
$\widetilde{\mathrm{SL}}(2,\mathbb{R})$	$\mathbb{R}^{2,1}\times\mathbb{R}$, $x^2+y^2-z^2=-1$ See Section 10.1, $o=\mathbf{e}_z$	See Sections 10.2 and 10.3	$\widetilde{\mathrm{SL}}(2,\mathbb{R})\rtimes\mathrm{O}(2)$	"Lift" of $\pi_1(\Sigma_g)$ with Σ_g compact genus g surface
Sol	$\mathbb{R}^4, w=1$, $ds^2=e^{-2z}dx^2+e^{2z}dy^2+dz^2$, $o=\mathbf{e}_w$	See Section 11.2	$\mathrm{Sol}\rtimes D_8$	$\mathbb{Z}^2\rtimes_M\mathbb{Z}$ with $M\in\mathrm{SL}(2,\mathbb{Z})$, hyperbolic

TABLE 1. The eight Thurston geometries. We denote the canonical basis $\{\mathbf{e}_x,\mathbf{e}_y,\mathbf{e}_z,\mathbf{e}_w\}$. We write (x,y,z,w) for the coordinates of a vector vector $\mathbf{v}$ in this basis. Note that $\mathrm{Isom}(\mathbb{R})\cong\mathbb{R}\rtimes\mathbb{Z}/2$.

7. Isotropic geometries

In this section we give implementation details for $\mathbb{E}^3$, S^3 and $\mathbb{H}^3$. For further background, we refer the reader to [BH99, Chapter I.2]. See also [Wee02]. Many details for these three geometries are very similar; for the convenience of the reader, we list these explicitly. In particular, we give distance functions for some simple shapes in standard positions. They can be conjugated by isometries to give signed distance functions for these shapes in general position. We also reference the possible discrete groups (or equivalently, manifolds) for each geometry.

7.1. **Euclidean space.** We represent $\mathbb{E}^3$ as the affine subspace $X = \{w = 1\}$ of $\mathbb{R}^4$. The origin is the point $o = [0, 0, 0, 1]$. The distance between two points $p_1 = [x_1, y_1, z_1, 1]$ and $p_2 = [x_2, y_2, z_2, 1]$ is given by

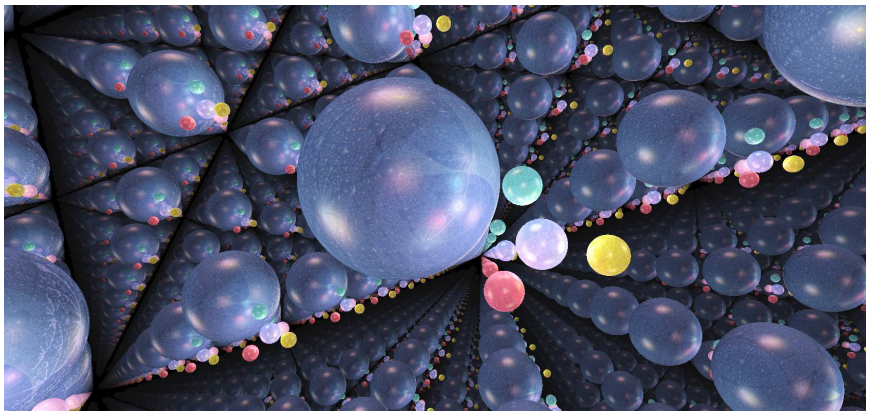
$$\operatorname{dist}(p_1, p_2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2}.$$

Using a hyperplane to represent $\mathbb{E}^3$ is standard in computer graphics because the isometry group of $\mathbb{E}^3$ acts on X by linear transformations of $\mathbb{R}^4$ preserving X. We identify the tangent space T_pX at a point $p \in X$ with the linear subspace $\{w = 0\}$ of $\mathbb{R}^4$. The arc length parametrized geodesic $\gamma(t)$ starting at p and directed by the unit vector $v \in T_pX$ is simply $\gamma(t) = p + tv$. In Table 2, we list signed distance functions for some simple objects in $\mathbb{E}^3$.

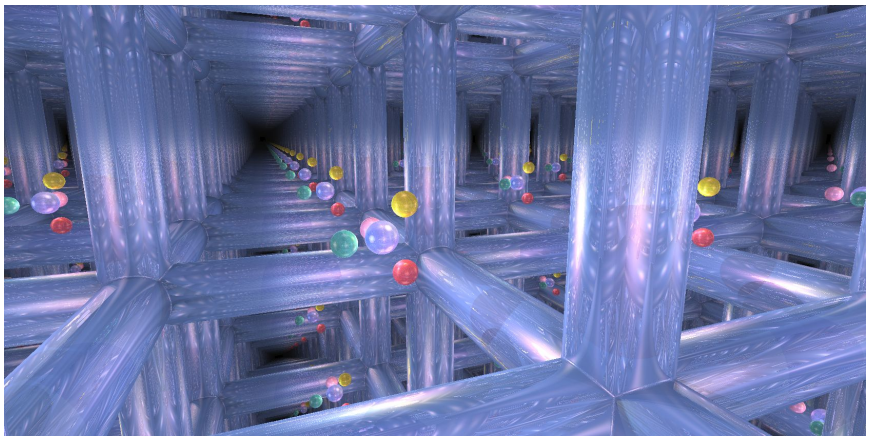
Object	Signed distance function
Ball of radius r centered at the origin o	$\sigma(p) = \sqrt{x^2 + y^2 + z^2} - r$
Solid cylinder of radius r with axis the geodesic $\gamma(t) = o + t\mathbf{e}_z$	$\sigma(p) = \sqrt{x^2 + y^2} - r$
Half-space $\{z \leqslant 0\}$	$\sigma(p) = z$

Table 2. Examples of signed distance functions in $\mathbb{E}^3$.

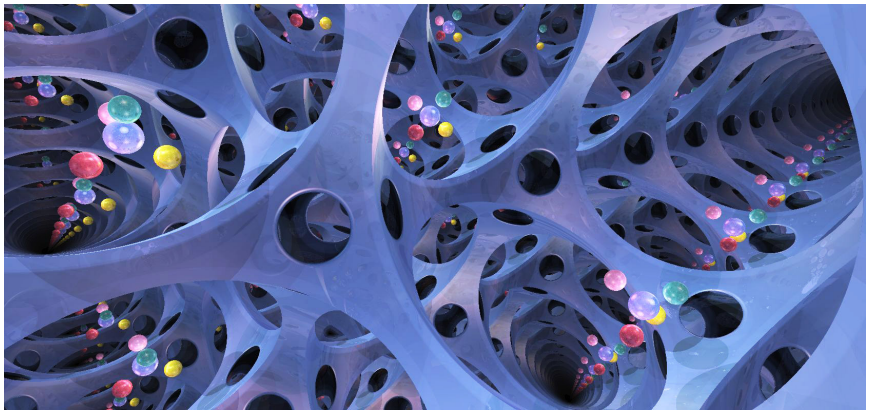
From a group theoretic point of view, the co-compact discrete subgroups of $\mathbb{E}^3$ have been classified. These are the crystallographic groups [BBC72]. Note that every finite volume euclidean three-manifold is finitely covered by the three-torus. In Figure 7.1, we show the in-space view for various scenes within the regular three-torus, rendered with a multicolor collection of five lights. In these images, light intensity falls off proportional to the inverse square of distance. An object receives lighting from the cell it is contained in and that cell's nearest neighbors.



(A) A single large ball.



(B) Solid cylinders around the edges of a fundamental domain.



(C) Edges of the fundamental domain rendered by deleting a large ball from the center and smaller balls from the vertices, as in Figure 2.1c.

FIGURE 7.1. Scenes in the regular three-torus, lit by a collection of lights represented by balls.

7.2. **The three-sphere.** We endow $\mathbb{R}^4$ with the standard scalar product. That is, given $p_1 = [x_1, y_1, z_1, w_1]$ and $p_2 = [x_2, y_2, z_2, w_2]$ we let

$$\langle p_1, p_2 \rangle = x_1x_2 + y_1y_2 + z_1z_2 + w_1w_2.$$

We view S^3 as the set X of points $p \in \mathbb{R}^4$ satisfying the identity $\langle p, p \rangle = 1$. We choose for the origin the point $o = [0, 0, 0, 1]$. The distance between two points p_1 and p_2 is characterized by

$$\cos(\mathrm{dist}(p_1, p_2)) = \langle p_1, p_2 \rangle.$$

The isometry group of S^3 acts on X by linear transformations of $\mathbb{R}^4$ preserving the scalar product and so X. We identify the tangent space T_pX at a point p in X with the linear subspace

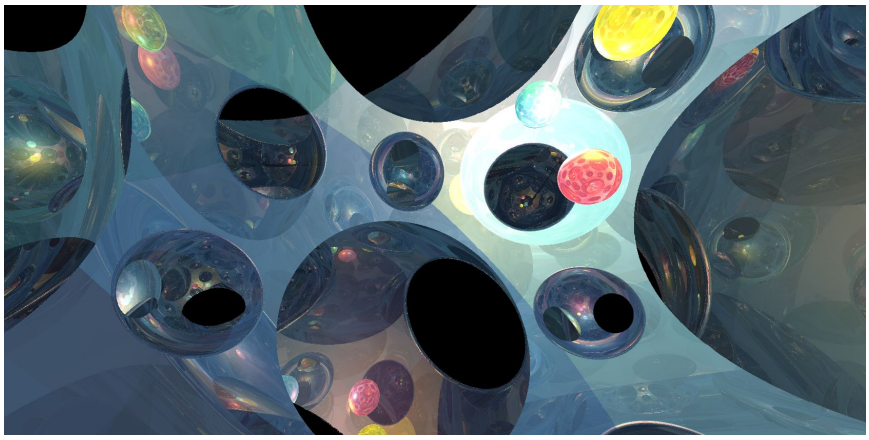
$$\big\{ v \in \mathbb{R}^4 \mid \langle p, v \rangle = 0 \big\}$$

of $\mathbb{R}^4$. The arc length parametrized geodesic $\gamma(t)$ starting at p and directed by the unit vector $v \in T_pX$ is given by $\gamma(t) = \cos(t)p + \sin(t)v$. In Table 3, we list a few examples of signed distance functions in S^3.

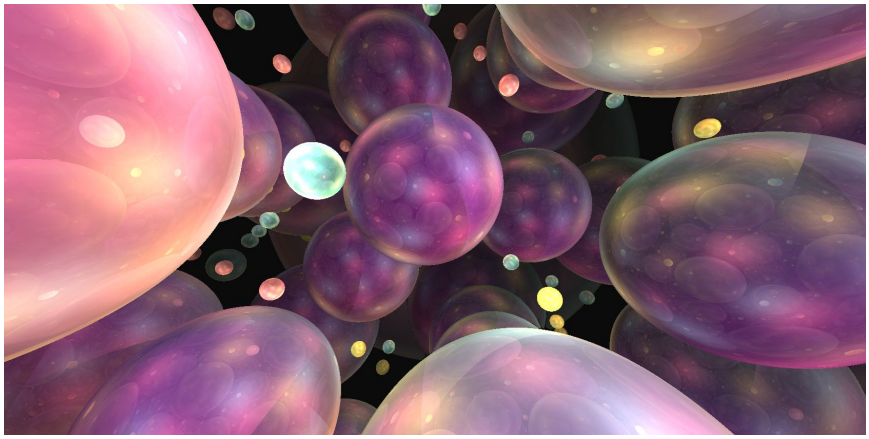
Object	Signed distance function
Ball of radius r centered at the origin o	$\sigma(p) = \arccos(w) - r$
Solid cylinder of radius r whose axis is the geodesic $\gamma(t) = \cos(t)o + \sin(t)\mathbf{e}_z$	$\sigma(p) = \arccos(\sqrt{w^2 + z^2}) - r$
Half-space $\{z \leqslant 0\}$	$\sigma(p) = \arcsin(z)$

TABLE 3. Examples of signed distance functions in S^3.

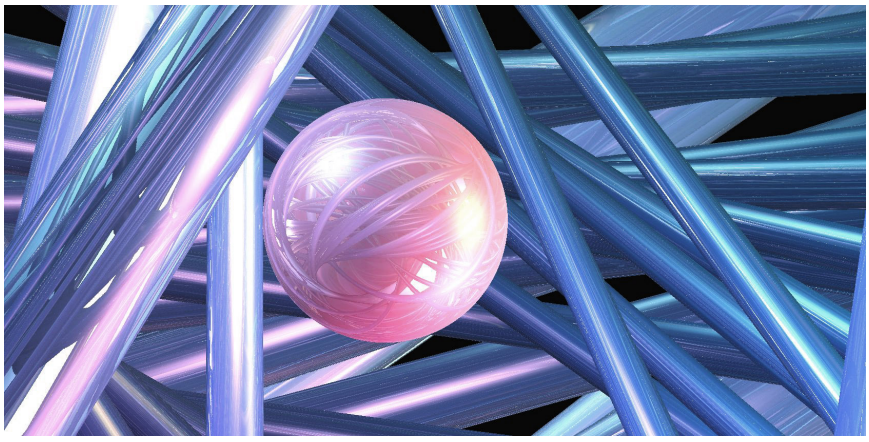
The finite subgroups of O(4) are classified in [Sco83, page 449]. In Figure 7.2 we show the in-space view for various scenes in spherical geometry. Figure 7.2a shows the quotient of S^3 by the quaternion group of order eight, Q_8. Edges of the fundamental domain are shown as in Figure 2.1c, but with balls also deleted from the edge midpoints. Figure 7.2b shows a single mirrored ball and three light sources in Poincaré dodecahedral space. Figure 7.2c shows the lifts of some randomly chosen fibers of the unit tangent bundle over S^2 (the Hopf fibration), and their reflected images in a ball. These are the fibers of the Seifert fiber space structure on spherical three-manifolds.



(A) The quotient of S^3 by the quaternion group of order eight, Q_8.



(B) Poincaré dodecahedral space.



(C) Hopf fibration.

FIGURE 7.2. Spherical Geometry.

7.3. **Hyperbolic space.** We endow $\mathbb{R}^4$ with a lorentzian inner product: for every $p_1 = [x_1, y_1, z_1, w_1]$ and $p_2 = [x_2, y_2, z_2, w_2]$ we let

$$\langle p_1, p_2 \rangle = x_1x_2 + y_1y_2 + z_1z_2 - w_1w_2.$$

We use the hyperboloid model of $\mathbb{H}^3$. This consists of the set X of points $p = [x, y, z, w]$ in $\mathbb{R}^4$ such that $\langle p, p \rangle = -1$ and $w > 0$. We choose for the origin the point $o = [0, 0, 0, 1]$. The distance between two points p_1 and p_2 is given by

$$\cosh(\mathrm{dist}(p_1, p_2)) = -\langle p_1, p_2 \rangle.$$

The isometry group of $\mathbb{H}^3$ acts on X by linear transformations of $\mathbb{R}^4$ preserving the lorentzian product and so X. We identify the tangent space T_pX at a point $p = [x, y, z, w]$ in X with the linear subspace

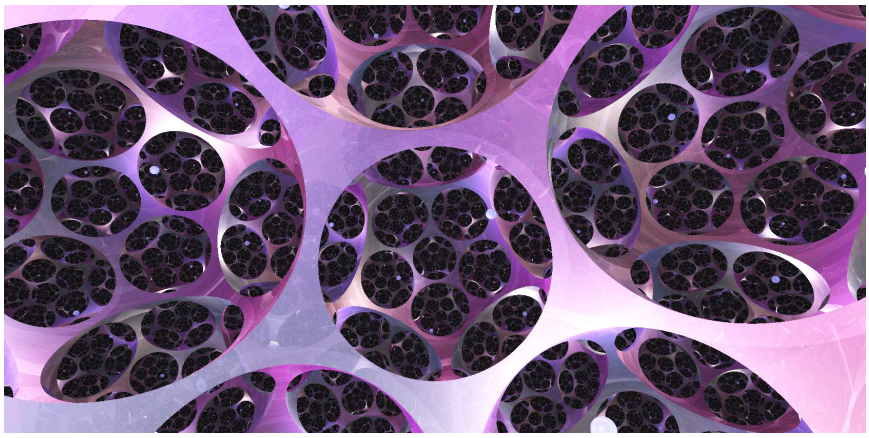
$$\big\{v \in \mathbb{R}^4 \mid \langle p, v \rangle = 0\big\}$$

of $\mathbb{R}^4$. The arc length parametrized geodesic $\gamma(t)$ starting at p and directed by the unit vector $v \in T_pX$ is given by $\gamma(t) = \cosh(t)p + \sinh(t)v$. In Table 4, we list a few examples of signed distance functions in $\mathbb{H}^3$.

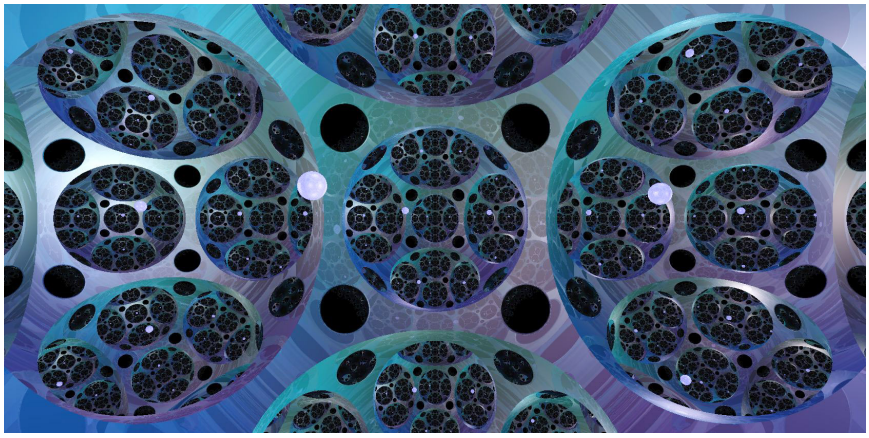
Object	Signed distance function
Ball of radius r centered at the origin o	$\sigma(p) = \mathrm{arccosh}(w) - r$
Solid cylinder of radius r whose axis is the geodesic $\gamma(t) = \cosh(t)o + \sinh(t)\mathbf{e}_z$	$\sigma(p) = \mathrm{arccosh}(\sqrt{w^2 - z^2}) - r$
Half-space $\{z \leqslant 0\}$	$\sigma(p) = \mathrm{arcsinh}(z)$

TABLE 4. Examples of signed distance functions in $\mathbb{H}^3$.

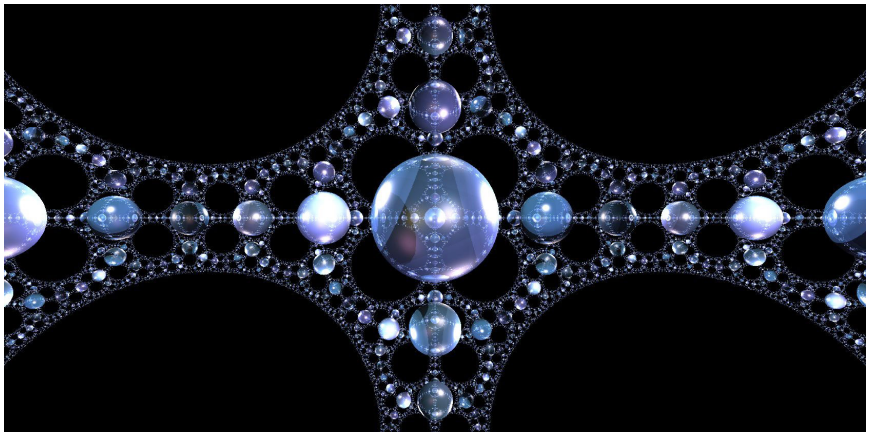
Of the eight Thurston geometries, the classification of hyperbolic manifolds (and orbifolds) is the least well understood. The software SnapPy [CDGW] lists numerous censuses of finite volume hyperbolic manifolds. In Figure 7.3 we show the in-space view for various scenes in hyperbolic geometry. Figure 7.3a shows Seifert-Weber dodecahedral space, with a fundamental domain drawn in a style similar to Figure 2.1b. Figure 7.3b shows the finite volume cusped orbifold formed from an ideal cube (with dihedral angles of $\pi/3$), by identifying opposite faces with a $\pi/2$ turn. The underlying manifold is S^3/Q_8 (see Figure 7.2a) minus the vertices of the cube, with cone angles of π at each edge of the cube. Figure 7.3c shows a sphere in an infinite volume hyperbolic orbifold formed from a hyperideal cube [NS17, Section 6.1] (with dihedral angles of $\pi/4$), by identifying opposite faces by translation. The limit set is



(A) Seifert-Weber dodecahedral space.



(B) A finite volume hyperbolic orbifold.



(C) An infinite volume hyperbolic orbifold.

FIGURE 7.3. Hyperbolic geometry.

the visible as the limiting pattern of spheres. The underlying manifold is the three-torus, minus a ball around the vertex, with cone angles of π at each edge of the cube.

7.4. **Facing and parallel transport.** By definition, for each isotropic geometry X, the isometry group $G = \mathrm{Isom}(X)$ acts transitively on the unit tangent bundle of X. It follows that the position and facing of an observer can be captured by a single isometry, as explained in Section 3.3. Nevertheless, to keep the code as geometry-independent as possible, we encode our position and facing by a pair (g, id) where g is an isometry of X and $\mathrm{id} \in \mathrm{O}(3)$ is the identity.

As we noted in Section 3.4, given any geodesic $\gamma\colon \mathbb{R} \to X$ starting at $p \in X$, there is a one-parameter orientation preserving subgroup $h\colon \mathbb{R} \to G$ such that $\gamma(t) = h(t)p$. Thus the corresponding parallel transport operator $T(t)\colon T_{\gamma(0)}X \to T_{\gamma(t)}X$ is simply $T(t) = d_p h(t)$. This considerably simplifies the computations: if an observer starts at (g, id) and follows γ for time t, then the observer's new position and facing are $(h(t)g, \mathrm{id})$.

7.5. **Lighting.** The calculation of lighting intensity for the isotropic geometries is straightforward in comparison to the other geometries. Recall from Equation (5.7) that the intensity $I(r, u)$ is inversely proportional to the area density of geodesic spheres.Equation (5.12) relates area density directly to Jacobi fields along the geodesic in the direction u. Here, all sectional curvatures are equal, so all Jacobi fields are parallel along geodesics, and have magnitude controlled by the curvature. Precisely, if $v \in u^\perp$ and v_t is the parallel transport of v along the geodesic with initial tangent u, the corresponding Jacobi fields J are below.

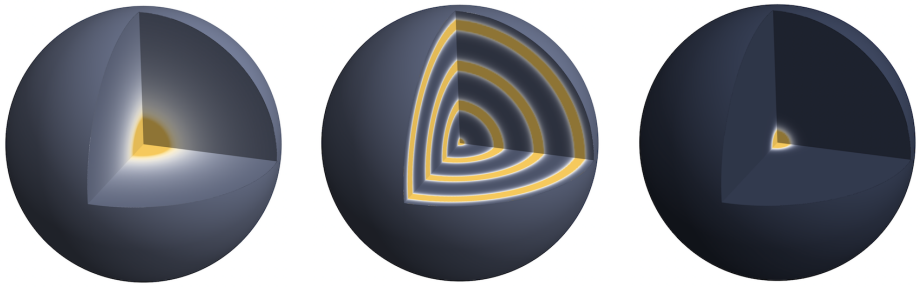
$$J_{\mathbb{E}^3}(t) = tv_t \qquad J_{S^3}(t) = \sin(t)v_t \qquad J_{\mathbb{H}^3}(t) = \sinh(t)v_t$$

Choosing a pair of orthonormal initial conditions and using Equation (5.12) gives the area densities:

$$\mathcal{A}_{\mathbb{E}^3}(r, u) = r^2 \qquad \mathcal{A}_{S^3}(r, u) = \sin(r)^2 \qquad \mathcal{A}_{\mathbb{H}^3}(r, u) = \sinh(r)^2.$$

Thus light intensity falls off quadratically with distance in euclidean space, and exponentially in hyperbolic space. In the three-sphere, the intensity initially decreases with distance, but beyond a distance of $\pi/2$, it increases as all light rays begin to converge towards the antipode. Figure 7.4 shows graphs of the intensity function $I(r, u)$ on the tangent space $T_q X$. A point at distance r from the origin in the direction u is colored by the value of $I(r, u)$. Dark blues represent low intensity, and yellows represent high intensity. Each plot depicts a ball of radius ten.

Note that $I(r, u)$ for the three-sphere diverges to infinity along spheres with $r = \pi n$ as, under the exponential map, all light refocuses at the light source or its antipode.



(A) Euclidean intensity. (B) Spherical intensity. (C) Hyperbolic intensity.

FIGURE 7.4. Graphs of the lighting intensity functions $I(r, u)$ for the isotropic geometries, drawn in the tangent space at the light source.

We now turn to the calculation of the lighting pairs $\mathcal{L}_s(q)$: the set of pairs (L, d_L) of initial tangent vectors L to geodesics joining s to q, and their corresponding lengths d_L. In all three isotropic geometries, this can be calculated using linear algebra in the ambient space $\mathbb{R}^4$ where the models reside.

In euclidean space, geodesics are unique. Given $s, q \in \mathbb{E}^3$, the required direction vector is simply $q - s$.

$$\mathcal{L}_s^{\mathbb{E}^3}(q) = \left\{ \left(\frac{q-s}{\|q-s\|}, \|q-s\| \right) \right\}$$

In spherical geometry, given $s, q \in S^3$ non-antipodal, let $\theta = \arccos\langle q, s\rangle$ be the acute angle between them. The shortest geodesic from s to q has length θ and direction $v = q - \langle s, q\rangle q$, appropriately rescaled. The second geodesic points in the opposite direction, with length $2\pi - \theta$.

$$\mathcal{L}_s^{S^3}(q) = \left\{ \left(\frac{v - \cos(\theta)s}{\sin\theta}, \theta \right), \left(\frac{\cos(\theta)s - v}{\sin\theta}, 2\pi - \theta \right) \right\}$$

Remark 7.1. Strictly speaking, we should also include copies of the above pairs with distances modified by adding $2\pi n$ for all integers $n > 0$. However, if either the light source or the scene is opaque then these copies are never relevant. ◊

In practice, we don't worry about s and q being antipodal: in a generic render, no pixels will involve such a situation. Moreover, if we are exceedingly unlucky and do have such a pixel, GPU code does

not crash when asked to, for example, divide by zero. It just gives up and moves on to the next pixel. However, one could special-case this situation: for a pair of antipodal points s, q, all directions from s reach q after traveling a distance π, and so we find that the set $\mathcal{L}_s^{S^3}(q)$ is uncountable. As the lighting intensity diverges to infinity as one approaches such a configuration, the pixels should be colored as bright as possible.

In hyperbolic geometry we proceed analogously to the three sphere, except that we use the Minkowski inner product. Given $s, q \in \mathbb{H}^3$, let $\delta = \operatorname{arccosh} |\langle q, s\rangle|$ be the hyperbolic distance between them. Geodesics between pairs of points in $\mathbb{H}^3$ are unique, so $\mathcal{L}_s^{\mathbb{H}^3}(q)$ is again a singleton:

$$\mathcal{L}_s^{\mathbb{H}^3}(q) = \left\{ \left(\frac{v - \cosh(\delta) s}{\sinh \delta}, \delta \right) \right\}.$$

8. Product geometries

Before describing the product geometries, we quickly introduce model spaces for S^2 and $\mathbb{H}^2$.

8.1. **Models of S^2 and $\mathbb{H}^2$.** Our models for S^2 and $\mathbb{H}^2$ are the same as those for S^3 and $\mathbb{H}^3$, with one fewer dimension:

- We view S^2 as the set $\mathcal{S}$ of points $q = [x, y, z]$ in $\mathbb{R}^3$ such that $\langle q, q\rangle = 1$, where $\langle \cdot, \cdot \rangle$ is the canonical scalar product in $\mathbb{R}^3$.
- We represent $\mathbb{H}^2$ as the set $\mathcal{H}$ of points $q = [x, y, z]$ in $\mathbb{R}^3$ such that $\langle q, q\rangle = -1$, where $\langle q_1, q_2\rangle = x_1 x_2 + y_1 y_2 - z_1 z_2$ is the lorentzian product in $\mathbb{R}^3$.

8.2. **Product geometries.** Our model for $S^2 \times \mathbb{E}$ (respectively $\mathbb{H}^2 \times \mathbb{E}$) is the subset $X = Y \times \mathbb{R}$ of $\mathbb{R}^4$, where $Y = \mathcal{S}$ (respectively $Y = \mathcal{H}$). We choose for the origin the point $o = [0, 0, 1, 0]$. The space X is equipped with the product distance. That is, given two points $p_1 = (q_1, w_1)$ and $p_2 = (q_2, w_2)$ in $Y \times \mathbb{R}$ we have

$$\operatorname{dist}_X(p_1, p_2)^2 = \operatorname{dist}_Y(q_1, q_2)^2 + |w_1 - w_2|^2.$$

The tangent space $T_p X$ at a point $p = (q, w)$ naturally splits as $T_q Y \times \mathbb{R}$. Given a vector $v \in T_p X$ we denote by v_Y and $v_{\mathbb{E}}$ its components in $T_q Y$ and $\mathbb{R}$ respectively. The arc length parametrized geodesic $\gamma(t)$ starting at $p = (q, w)$ in the direction of the unit vector $v \in T_p X$ is given by

$$\gamma(t) = \big(\gamma_Y(\|v_Y\| t), w + t v_{\mathbb{E}}\big),$$

where $\gamma_Y \colon \mathbb{R} \to Y$ is the geodesic ray in Y starting at q with initial tangent vector $v_Y / \|v_Y\|$.

Next, we consider signed distance functions. As usual, the distance formula gives us the signed distance function for a ball. We call an object $\mathcal{V}$ *vertical* if it is the pre-image of a non empty subset $\mathcal{U} \subset Y$ by the projection $\pi\colon X \to Y$. The signed distance function for such an object $\mathcal{V}$ is given by

$$\sigma(p) = \mathrm{dist}_X(p, \mathcal{V}) = \mathrm{dist}_Y(\pi(p), \mathcal{U}).$$

We define *horizontal* objects, and obtain their signed distance functions in an analogous way. Tables 5 and 6 list a few examples of such signed distance functions.

Object	Signed distance function
Solid cylinder of radius r with axis the geodesic $\gamma(t) = o + t\mathbf{e}_w$	$\sigma(p) = \arccos(z) - r$
Half-space $\{y \leqslant 0\}$	$\sigma(p) = \arcsin(y)$
Half-space $\{w \leqslant 0\}$	$\sigma(p) = w$

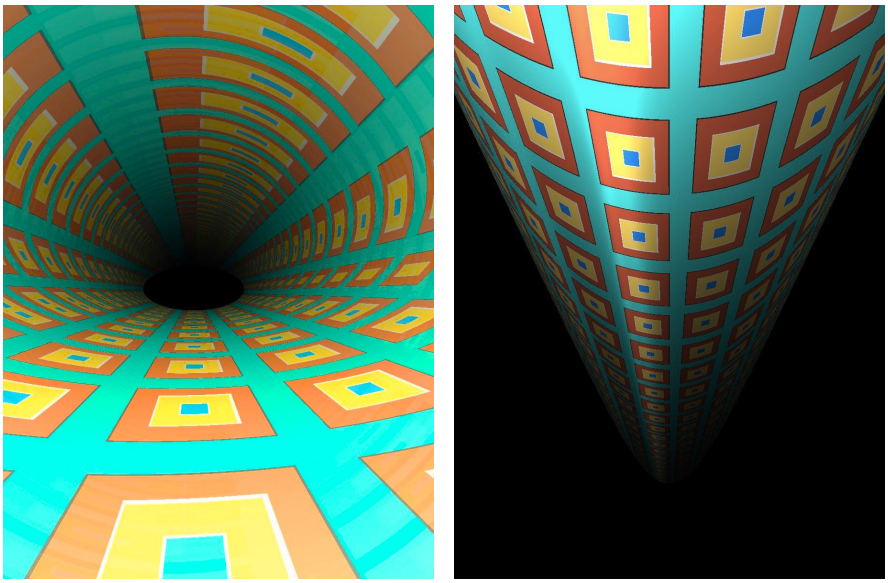
TABLE 5. Examples of signed distance functions in $S^2 \times \mathbb{R}$.

Object	Signed distance function
Cylinder of radius r whose axis is the geodesic $\gamma(t) = o + t\mathbf{e}_w$	$\sigma(p) = \mathrm{arccosh}(z) - r$
Half-space $\{y \leqslant 0\}$	$\sigma(p) = \mathrm{arcsinh}(y)$
Half-space $\{w \leqslant 0\}$	$\sigma(p) = w$

TABLE 6. Examples of signed distance functions in $\mathbb{H}^2 \times \mathbb{R}$.

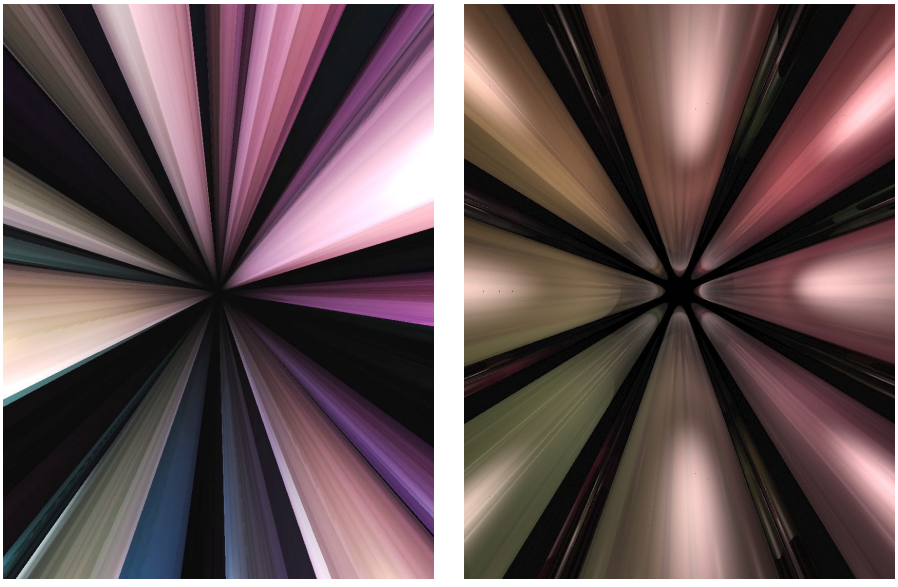
Figure 8.1 shows vertical half-spaces in the product geometries. Figure 8.2 shows solid cylinders around some fibers in the $\mathbb{E}$ direction for the product geometries. In Figure 8.2a we place solid cylinders around fibers above the vertices of an icosahedron in the S^2 factor. In Figure 8.2b the solid cylinders are around fibers in the $\mathbb{E}$ direction.

8.3. **Facing and parallel transport.** Unlike for the isotropic geometries, the position and facing of the observer cannot be encoded with a single element of $G = \mathrm{Isom}(X)$. Hence we represent it by a pair $(g, m) \in G \times \mathrm{O}(3)$ as explained in Section 3.3. Nevertheless, if



(A) $S^2 \times \mathbb{E}$. (B) $\mathbb{H}^2 \times \mathbb{E}$.

FIGURE 8.1. Vertical half-spaces in the $S^2 \times \mathbb{E}$ and $\mathbb{H}^2 \times \mathbb{E}$ geometries.



(A) Fibers in the $S^2 \times \mathbb{E}$. (B) Fibers in $\mathbb{H}^2 \times \mathbb{E}$.

FIGURE 8.2. Fibers of the Seifert fiber space structures in manifolds with product geometry.

$\gamma\colon \mathbb{R} \to X$ is a geodesic starting at the observer's position p, there is still a one-parameter orientation preserving subgroup $h\colon \mathbb{R} \to G$ such that $\gamma(t) = h(t)p$. Thus after moving along γ for a time t, the observer's new position and facing is $(h(t)g, m)$.

8.4. **Lighting.** We again use Equation (5.12) to reduce the calculation of area density (and hence light intensity) to the computation of Jacobi fields. Let $q \in X$, choose a unit vector $u \in T_qX$ and let γ be the geodesic starting at q with initial tangent u. General Jacobi fields need not be parallel along γ, and may rotate in the presence of a gradient in sectional curvature. When $v \in u^\perp$ is such that the curvature κ of the plane spanned by $\{u, v\}$ is a local extremum however, then the Jacobi field with initial condition $\dot{J}(0) = v$ is parallel along γ. In this case, its magnitude is determined by κ, as in Section 7.5.

If u is vertical (that is, $u_Y = 0$), then X is symmetric under rotation about u, and all planes containing u have zero sectional curvature. If $v \in u^\perp$ has parallel translate v_t along γ, then the corresponding Jacobi field is $J(t) = tv_t$. Choosing two such orthonormal conditions, Equation (5.12) implies that $\mathcal{A}_X(r, u) = r^2$.

In general, suppose that u makes an angle of β with the vertical. Then u is contained in a unique vertical plane V, which again has zero sectional curvature. This realizes one of the extremal curvatures at u (it is a maximum for $\mathbb{H}^2 \times \mathbb{E}$ and a minimum for $S^2 \times \mathbb{E}$). Choosing $v \in T_qX$ extending u to an orthonormal basis for V, the Jacobi field with initial condition v is $J(t) = tv_t$ as above.

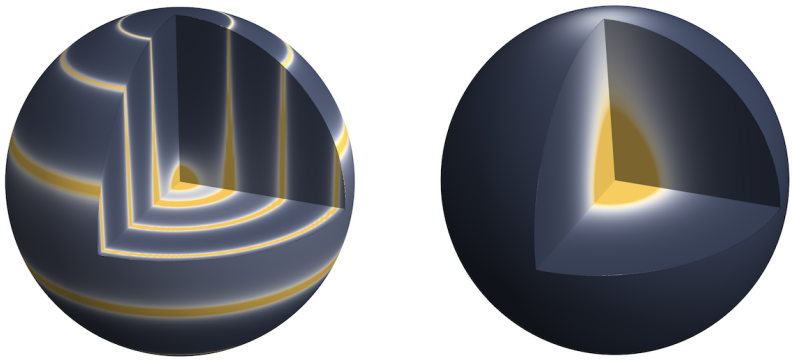
The other extremal curvature is realized by the plane P, orthogonal to V and containing u. Using the bilinearity of the Riemann curvature tensor, one can calculate this extremal curvature from the angle β that u makes with the vertical, and the curvature $K(H) = \pm 1$ of the horizontal H plane H:

$$K(P) = \cos^2(\beta)K(V) + \sin^2(\beta)K(H) = \pm \sin^2(\beta)$$

Let $w \in T_qX$ extend u to an orthonormal basis for P, and w_t be its parallel translate along γ. The Jacobi field with initial condition w is $J(t) = \frac{f(t \sin \beta)}{\sin \beta} w_t$, where f is either sine or hyperbolic sine as $K(P)$ is greater or less than zero respectively. Combining these with Equation (5.12) gives the area density for each of the product geometries below.

$$\mathcal{A}_{S^2\times\mathbb{E}}(r, u) = r\frac{\sin(r \sin \beta)}{\sin \beta} \qquad \mathcal{A}_{\mathbb{H}^2\times\mathbb{E}}(r, u) = r\frac{\sinh(r \sin \beta)}{\sin \beta} \tag{8.1}$$

Figure 8.3 shows the behavior of $I(r,u) = 1/\mathcal{A}(r,u)$ on a ball of radius ten in the tangent space at q for the two product geometries. Figure 8.4 shows some effects of this behavior on the in-space view.



(A) The intensity in $S^2 \times \mathbb{E}$ periodically blows up.

(B) The intensity in $\mathbb{H}^2 \times \mathbb{E}$ drops off exponentially away from the $\mathbb{E}$ direction.

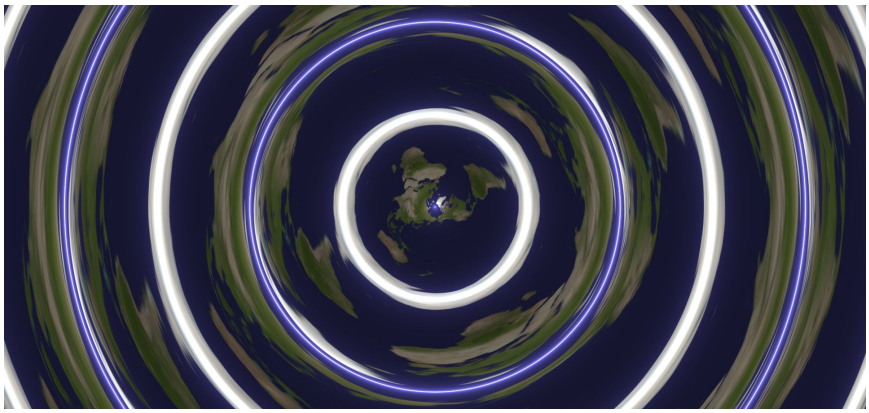
FIGURE 8.3. The lighting intensity functions $I(r,u)$ in the product geometries.

Finally, we must also compute the directions from a point $s \in X$ to the light source at $q \in X$. To simplify the notation here, we will write each lighting pair of $\mathcal{L}_s(q)$ not as a pair (L, d_L), but as a vector $d_L L$ of length d_L in the direction L. Let $s, q \in X$ and let $d_Y = \text{dist}_Y(s_Y, q_Y)$, $d_{\mathbb{E}} = |q_{\mathbb{E}} - s_{\mathbb{E}}|$ be the distances between their projections into the respective factors of $X = Y \times \mathbb{E}$. Recall that the standard basis vector $\mathbf{e}_w$ points along the $\mathbb{E}$ direction. We compute the unit vector $v_Y \in T_{s_Y} Y$ pointing along the shortest geodesic from s_Y to q_Y as in Section 7.5. The element of $\mathcal{L}_s(q)$ corresponding to the shortest geodesic is then $d_Y v_Y + d_{\mathbb{E}} \mathbf{e}_w$. In $\mathbb{H}^2 \times \mathbb{E}$ geodesics are unique, so with this we are done:

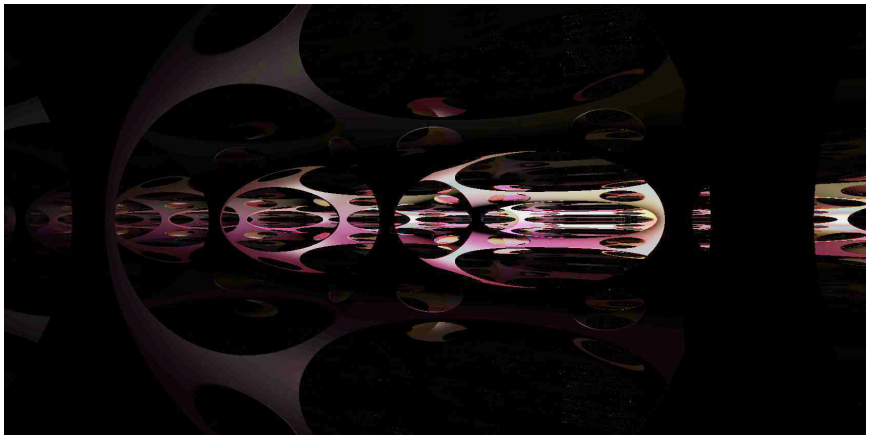
$$\mathcal{L}_s^{\mathbb{H}^2 \times \mathbb{E}}(q) = \{d_Y v_Y + d_{\mathbb{E}} \mathbf{e}_w\} = \left\{ d_Y \frac{s_Y - \cosh(d_Y) q_Y}{\sinh(d_Y)} + d_{\mathbb{E}} \mathbf{e}_w \right\}$$

In $S^2 \times \mathbb{E}$, there are three cases to deal with: first the generic case, second when s, q lie on the same horizontal S^2, and third when s_Y, q_Y are antipodal. As for S^3, in the implementation we don't worry about the non-generic cases; the lighting intensity at such points is the limit of the lighting intensity for the generic case.

In the generic case, there are countably many geodesics between s and q. All of these geodesics lie on the cylinder formed by taking the product of the $\mathbb{E}$ direction with the great circle containing s_Y and



(A) The sphere $S^2 \times \{0\}$ in $S^2 \times \mathbb{E}$, lit by a single light above the north pole. The viewer is in the same position as the light, looking along the $\mathbb{E}$ direction. The light intensity blows up at both the north and south poles of $S^2 \times \{0\}$. The viewer sees each pole as a collection of concentric rings, together with a point for the north pole directly below.



(B) A tiling with a single light source in $\mathbb{H}^2 \times \mathbb{E}$. The light source is in the center of one of the bright tiles in front of the viewer. From a distance, the exponential fall-off in the hyperbolic directions makes the light look like a spotlight shining along the $\mathbb{E}$ direction.

FIGURE 8.4. In-space views highlighting consequences of the lighting intensities for the product geometries.

q_Y. For each natural number $n \geqslant 0$, there are two geodesics – one starting by traveling the 'short way' around the S^2 factor, followed by n additional full turns, and the other the 'long way' followed by n

additional turns. All together, this gives the set of directions

$$\mathcal{L}_s^{S^2\times\mathbb{E}}(q) = \bigcup_{n\geqslant 0}\Big\{(2\pi n + d_Y)v_Y + d_{\mathbb{E}}\mathbf{e}_w, (2\pi(n+1) - d_Y)v_Y + d_{\mathbb{E}}\mathbf{e}_w\Big\}.$$

If $s_{\mathbb{E}} = q_{\mathbb{E}}$ and s_{S^2}, q_{S^2} are not antipodal, then we just set $d_{\mathbb{E}} = 0$ above. As in Remark 7.1, all but the shortest two are irrelevant if either the light source or the scene is opaque.

In the third case, where s_Y, q_Y are antipodal in S^2, there are uncountably many geodesics joining s to q. Their directions are a countable sequence of rings in the unit sphere in T_sX accumulating on the horizontal equatorial circle.

There are only seven manifolds with $S^2 \times \mathbb{E}$ geometry. These are listed in [Sco83, page 457]. In Figure 8.5, we show the in-space view for various scenes in the Hopf manifold $S^2 \times S^1$. Figures 8.5a and 8.5b show a collection of spheres spaced at the vertices of a regular dodecahedron. Figure 8.5c shows a slab $S^2 \times [-\epsilon, \epsilon]$, with holes cut out at the vertices of the dodecahedron.

The manifolds with $\mathbb{H}^2 \times \mathbb{E}$ geometry are classified in [Sco83, Theorem 4.13]. In Figure 8.6, we show the in-space view for various scenes in $\mathbb{H}^2 \times \mathbb{E}$ geometry. All of these images show the orbifold which is the product of a circle with a torus T containing a cone point of angle π. Figures 8.6a and 8.6b show a collection of spheres, four in each fundamental domain. Figure 8.6c shows a slab $T \times [-\epsilon, \epsilon]$, with four holes cut from the fundamental domain of T, and a further hole cut around the cone point.

9. Nil

9.1. **Heisenberg model.** There are several models for Nil. Probably the most commonly used is the Heisenberg model. The Heisenberg group H is the group of 3×3 upper triangular matrices of the form

$$\begin{bmatrix} 1 & x & z \\ 0 & 1 & y \\ 0 & 0 & 1 \end{bmatrix}.$$

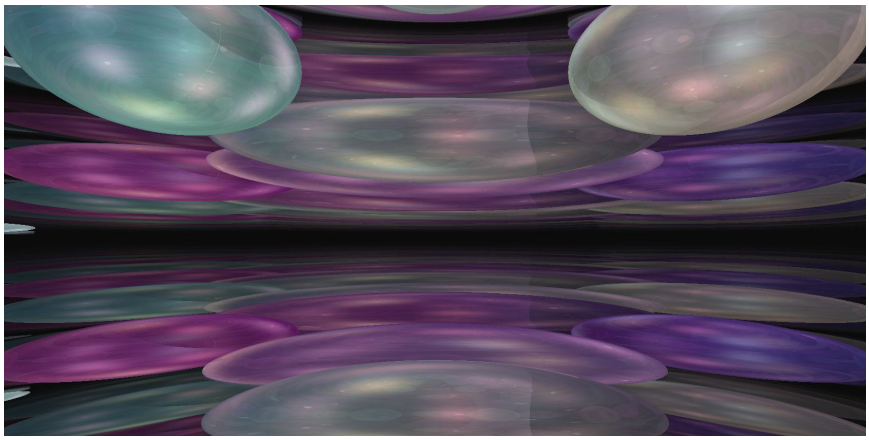
We identify this with $\mathbb{R}^3$ through the x-, y-, and z-coordinates. The metric

$$ds^2 = dx^2 + dy^2 + (dz - xdy)^2$$

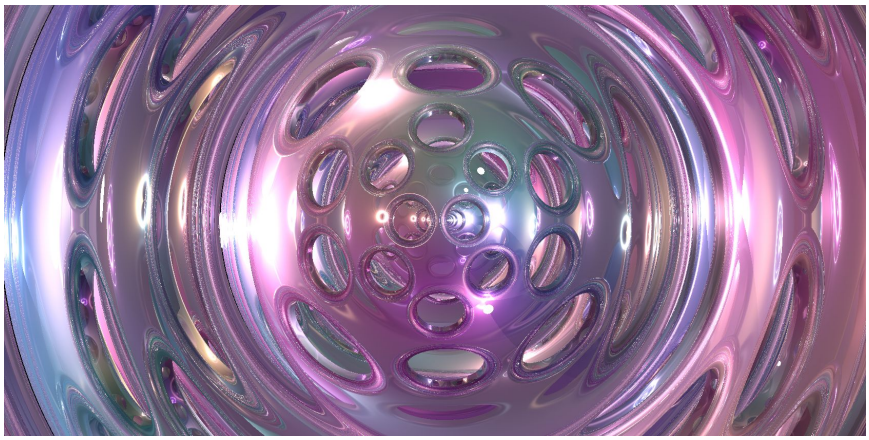
is invariant under the left action of H on itself.

The space (H, ds^2) has a major drawback for our purposes. To see this, let o be point $[0, 0, 0]$ (corresponding to the identity matrix) which we see as the origin of the space. The group of isometries of (H, ds^2) fixing o is isomorphic to $\mathrm{O}(2)$. In particular, it contains a one-parameter

(A) Looking along the $\mathbb{E}$ direction.

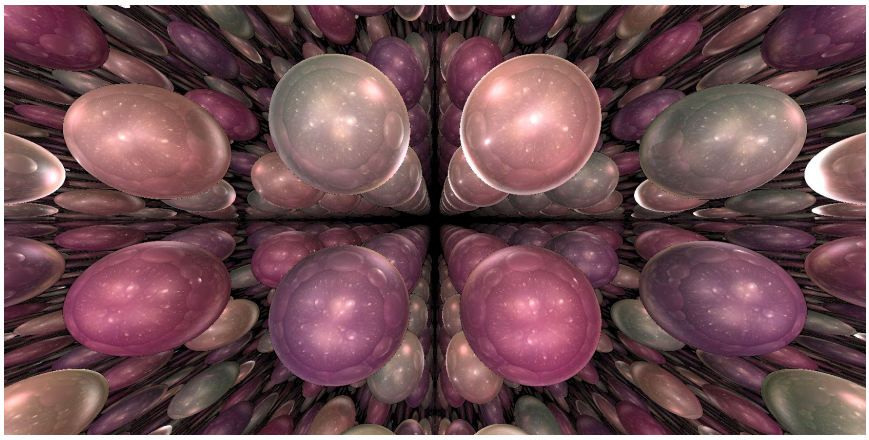


(B) Looking along an S^2 direction.

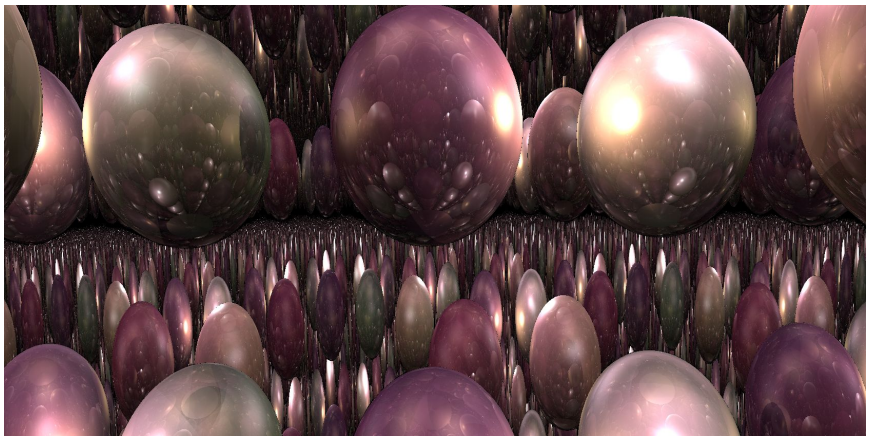


(C) Looking along the $\mathbb{E}$ direction.

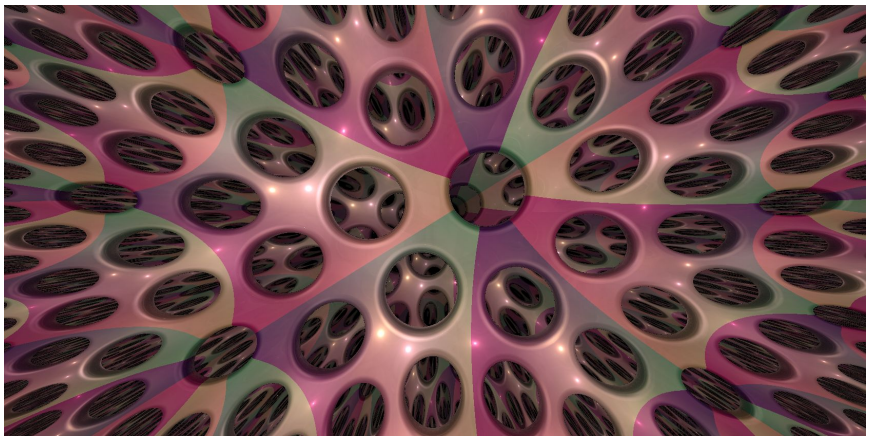
FIGURE 8.5. $S^2 \times \mathbb{E}$ Geometry. The Hopf manifold $S^2 \times S^1$.



(A) Looking along the $\mathbb{E}$ direction.



(B) Looking along an $\mathbb{H}^2$ direction.



(C) Looking along the $\mathbb{E}$ direction.

FIGURE 8.6. $\mathbb{H}^2 \times \mathbb{E}$ Geometry. The product of a torus with cone point of angle π, with a circle.

subgroup of rotations. These rotations are difficult to visualize in the Heisenberg model of Nil. See Figure 9.1.

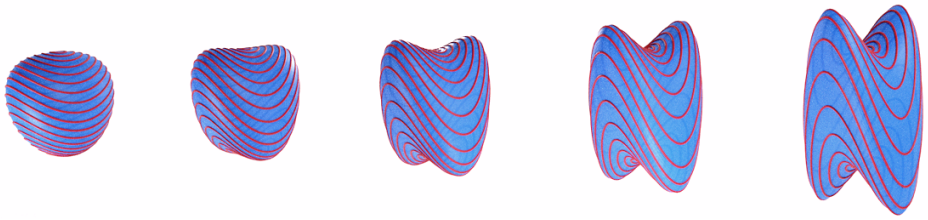


FIGURE 9.1. Balls of radius one to five in the Heisenberg model of Nil. The images have been rescaled to take up approximately the same space on the page. The red curves are invariant under the rotations fixing the origin.

9.2. **Rotation invariant model.** For our computations we use a "rotation invariant" model of Nil. The underlying space of the model is the affine subspace X of $\mathbb{R}^4$ defined by $w = 1$. The group law is as follows: the point $[x, y, z, 1]$ acts on X on the left as the matrix

$$\begin{bmatrix} 1 & 0 & 0 & x \\ 0 & 1 & 0 & y \\ -y/2 & x/2 & 1 & z \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

The origin o is the point $[0, 0, 0, 1]$. Its tangent space T_oX is identified with the linear subspace of $\mathbb{R}^4$ given by the equation $w = 0$. Our reference frame is $e = (\mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z)$ where $(\mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z, \mathbf{e}_w)$ is the standard basis of $\mathbb{R}^4$. The metric tensor at the point $p = [x, y, z, 1]$ is now given by

$$ds^2 = dx^2 + dy^2 + \left(dz - \frac{1}{2}(xdy - ydx)\right)^2.$$

The map

$$\begin{array}{ccc} H & \to & X \\ [x, y, z] & \mapsto & \left[x, y, z - \frac{1}{2}xy, 1\right] \end{array}$$

is an isometry between the Heisenberg model and the rotation invariant model. For every $\alpha \in \mathbb{R}$, we write R_α for the transformation with matrix

$$\begin{bmatrix} \cos\alpha & -\sin\alpha & 0 & 0 \\ \sin\alpha & \cos\alpha & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

One can check that R_α is an isometry of X, rotating by angle α around the z-axis. Let F be the transformation with matrix

$$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

This is another isometry of X, in this case it *flips* the z-axis, and satisfies $F \circ R_\alpha \circ F^{-1} = R_{-\alpha}$. These two kinds of isometries generate the stabilizer $K = \mathrm{O}(2)$ of o in $G = \mathrm{Isom}(X)$.

9.3. **Geodesic flow and parallel transport.** The solution of the geodesic flow in the Heisenberg model has been computed, for example in [Mol03]. We could convert the solution into our rotation invariant model X. Instead, we take this opportunity to illustrate Grayson's method (Sections 3.2.1 and 3.4.1) and calculate the geodesic flow and parallel-transport operator directly in X, as follows.

Let $\gamma\colon \mathbb{R} \to X$ be a geodesic in Nil and $T(t)\colon T_{\gamma(0)}X \to T_{\gamma(t)}X$ be the corresponding parallel-transport operator. We define two paths $u\colon \mathbb{R} \to T_oX$ and $Q\colon \mathbb{R} \to \mathrm{SO}(3)$ by the following relations

$$\begin{aligned} \dot\gamma(t) &= d_oL_{\gamma(t)}u(t), \\ T(t) \circ d_oL_{\gamma(0)} &= d_oL_{\gamma(t)}Q(t). \end{aligned}$$

Recall that the identification of parallel transport with the path Q in $\mathrm{SO}(3)$ is done via our reference frame $e = (\mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z)$ at the origin o. After some computation, Equations (3.2) and (3.6) respectively become

$$\begin{cases} \dot u_x = -u_z u_y \\ \dot u_y = u_z u_x \\ \dot u_z = 0 \end{cases}$$

and

$$\dot Q + BQ = 0 \quad \text{where} \quad B = \frac{1}{2}\begin{bmatrix} 0 & -u_z & -u_y \\ u_z & 0 & u_x \\ u_y & -u_x & 0 \end{bmatrix}.$$

For the initial condition $u(0) = [a\cos\alpha, a\sin\alpha, c, 0]$, where $a \in \mathbb{R}_+$ and $c \in \mathbb{R}$ satisfy $a^2 + c^2 = 1$, one gets

$$u(t) = [a\cos(ct+\alpha), a\sin(ct+\alpha), c, 0].$$

In order to get the expression for Q, we follow the strategy detailed in Section 3.4.1 and obtain

$$Q(t) = dR_\alpha e^{ctU_1} P e^{-\frac{1}{2}tU_2} P^{-1} dR_\alpha^{-1}, \quad \forall t \in \mathbb{R},$$

where

$$U_1 = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad U_2 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix},$$

and

$$dR_\alpha = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad P = \begin{bmatrix} a & 0 & -c \\ 0 & 1 & 0 \\ c & 0 & a \end{bmatrix}.$$

Note that $dR_\alpha \colon T_oX \to T_oX$ is the differential of the rotation R_α written in the reference frame $e = (\mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z)$.

Let us now move back to the original geodesic $\gamma \colon \mathbb{R} \to X$, which we write as

$$\gamma(t) = [x(t), y(t), z(t), 1].$$

Without loss of generality we can assume that $\gamma(0) = o$. Equation (3.1) becomes

$$\begin{cases} \dot{x} = u_x \\ \dot{y} = u_y \\ \dot{z} = u_z + \dfrac{1}{2}(xu_y - yu_x) \end{cases}.$$

Plugging in our solution for u, we finally get

$$\begin{cases} x(t) = \dfrac{2a}{c}\sin\left(\dfrac{ct}{2}\right)\cos\left(\dfrac{ct}{2} + \alpha\right) \\ y(t) = \dfrac{2a}{c}\sin\left(\dfrac{ct}{2}\right)\sin\left(\dfrac{ct}{2} + \alpha\right) \\ z(t) = ct + \dfrac{1}{2}\dfrac{a^2}{c^2}\big(ct - \sin(ct)\big) \end{cases} \quad \text{whenever } c \neq 0, \tag{9.1}$$

and otherwise

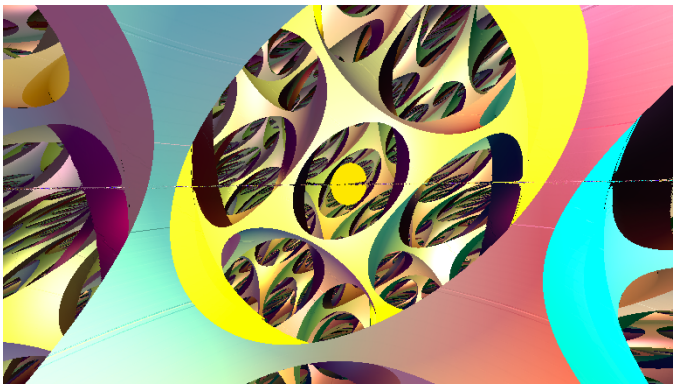
$$\begin{cases} x(t) = a\cos(\alpha)t \\ y(t) = a\sin(\alpha)t \\ z(t) = 0. \end{cases} \tag{9.2}$$

Remark 9.3. If c is very small but not zero, the above formulas are the source of significant numerical errors. This is due to the term

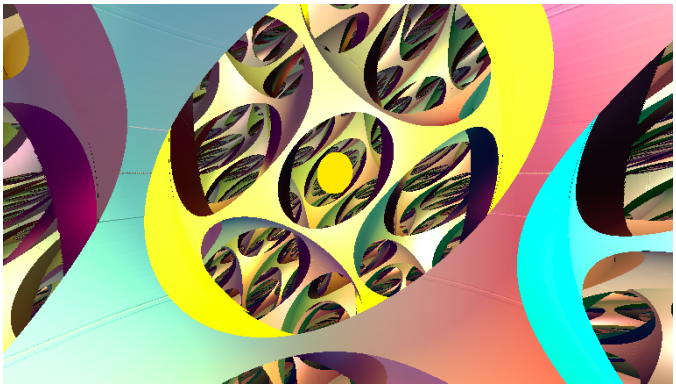
$$\frac{ct - \sin(ct)}{c^2},$$

see Section 2.4.1(2). In practice, this causes noise around the xy-plane, see Figure 9.2. To fix this issue, when ct is small, we replace the formula

given in Equation (9.1) by its asymptotic expansion of order seven around zero. ◊



(A) Using Equation (9.1). One observes noise around the xy-plane.



(B) Replacing Equation (9.1) by an asymptotic expansion in a neighborhood of the xy-plane.

FIGURE 9.2. Fixing the instability of the formula around $c = 0$. Both pictures represent the lattice of Nil given by the integer Heisenberg group. Here we choose a simple color scheme to highlight the noise.

9.4. **Distance to a vertical object.** Observe that Nil comes with a natural 1-Lipschitz projection $\pi\colon X \to \mathbb{E}^2$, sending $[x, y, z, 1]$ to $[x, y]$. In analogy with objects in the product geometries, we call the pre-image under π of any non-empty subset of $\mathbb{E}^2$ a *vertical* object. For example, any affine plane with equation $\alpha x + \beta y = \gamma$ is a vertical object.

Lemma 9.4. *Let S be a subset of $\mathbb{E}^2$ and $Z = \pi^{-1}(S)$ the associated vertical object. The distance from any point $p \in X$ to Z coincides with the distance between $\pi(p)$ and S in $\mathbb{E}^2$.*

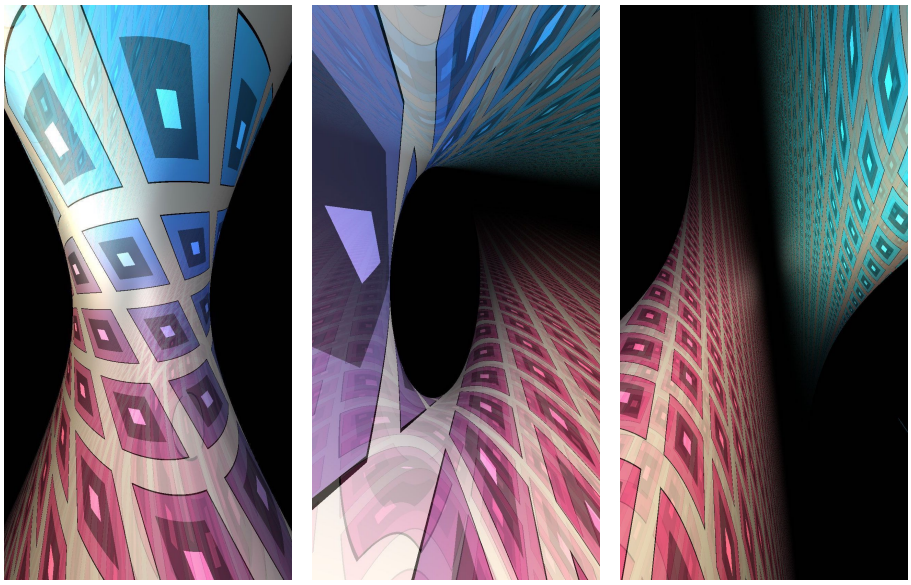
Proof. Any isometry of X preserves the fibers of the projection π and induces an isometry of $\mathbb{E}^2$. Hence, applying a translation, it suffices to prove the claim in the case that p is the origin o. Similarly, applying a rotation, we can assume that the projection $\pi(o)$ on S is a point q of the form $q = [x, 0]$, with $x \geqslant 0$. Since π is 1-Lipschitz, we have

$$\operatorname{dist}(\pi(o), S) \leqslant \operatorname{dist}(o, Z).$$

Let us explain the reverse inequality. We have seen previously that the map $\gamma\colon \mathbb{R} \to X$, mapping t to $[t, 0, 0, 1]$, is a geodesic of Nil. Hence the distance in Nil between o and the pre-image $\widetilde{q} = [x, 0, 0, 1]$ of q is at most x. Consequently

$$\operatorname{dist}(o, Z) \leqslant \operatorname{dist}(o, \widetilde{q}) \leqslant x \leqslant \operatorname{dist}(\pi(o), S). \qquad \square$$

Figure 9.3 shows a vertical half-space in Nil. We texture the boundary with squares of side length one in its euclidean metric. Note that here (and in Sections 10 and 11) we extend the notion of a half-space from that given at the start of Section 3.7: the boundary may not be totally geodesic. Figure 9.12c shows vertical solid cylinders.



(A) (B) (C)

FIGURE 9.3. Three views of a vertical half-space in Nil geometry. We see multiple reflections of the plane in itself, due to the spiraling of geodesics. Rendered with artificial (constant) light intensity, and fog.

9.5. **Exact distance and direction to a point.** Since X is homogeneous, we only need to compute the distance between any point $p \in X$ and the origin. In order to calculate lighting pairs, we need to compute the direction at the origin $v \in T_oX$ of the geodesics γ from o to p. Unfortunately, in Nil there is no closed-form expression for either of these two quantities. We compute both using the same numerical approach. Using the flip symmetry, we may assume that the coordinates $[x, y, z, 1]$ of p satisfy $z \geqslant 0$.

Assume first that the point $p = [x, y, z, 1]$ lies neither on the xy-plane nor on the z-axis. Let γ be a geodesic from o to p. That is, $\gamma(0) = o$ and $\gamma(t) = p$, for some $t \geqslant 0$. As in Section 9.3, we write $v = [a\cos\alpha, a\sin\alpha, c, 0]$ for its (unit) tangent vector at o. We deduce from Equation (9.1) that

$$z = \phi + \frac{\rho^2}{8\sin^2(\phi/2)}(\phi - \sin\phi),$$

where $\rho^2 = x^2 + y^2$ and $\phi = ct$. These quantities have the following useful geometric interpretation:

- ρ is the distance in $\mathbb{E}^2$ between $\pi(o)$ and $\pi(p)$, and
- ϕ is the angle described by the projection of γ in $\mathbb{E}^2$.

Computing the directions from o to p consists of solving a system with five unknowns (a, c, α, t, and ϕ) and five equations (the three given by Equation (9.1) along with the relations $a^2 + c^2 = 1$ and $\phi = ct$). Once ϕ has been found, it is an exercise to uniquely recover a, c, α and t by directly solving the equations. Hence there is a one-to-one correspondence between the geodesics joining o to p and the zeros of the function

$$\chi_{\rho,z}(\phi) = -z + \phi + \frac{\rho^2}{8\sin^2(\phi/2)}(\phi - \sin\phi),$$

see Figure 9.4.

A geodesic γ is minimizing if and only if the corresponding angle ϕ belongs to $(0, 2\pi)$. It turns out that $\chi_{\rho,z}$ is strictly convex on the interval $(2k\pi, 2k\pi+2\pi)$ for every integer $k \geqslant 0$. Moreover, it is increasing on $(0, 2\pi)$. In order to find the minimizing geodesic from o to p we numerically compute the unique zero of $\chi_{\rho,z}$ on $(0, 2\pi)$ using Newton's method.

For physically accurate lighting, we also need the lighting pairs, $\mathcal{L}_o(p)$, as defined in Section 5.6. Using binary search, we find a value of $\phi_0 \in (2\pi, 4\pi)$ where $\chi_{z,\rho}$ is positive and $d\chi_{z,\rho}/d\phi$ is negative. We then run Newton's method, starting from ϕ_0, producing a sequence $\{\phi_n\}$. Recall that $\chi_{z,\rho}$ is strictly convex on $(2\pi, 4\pi)$. Hence if the equation

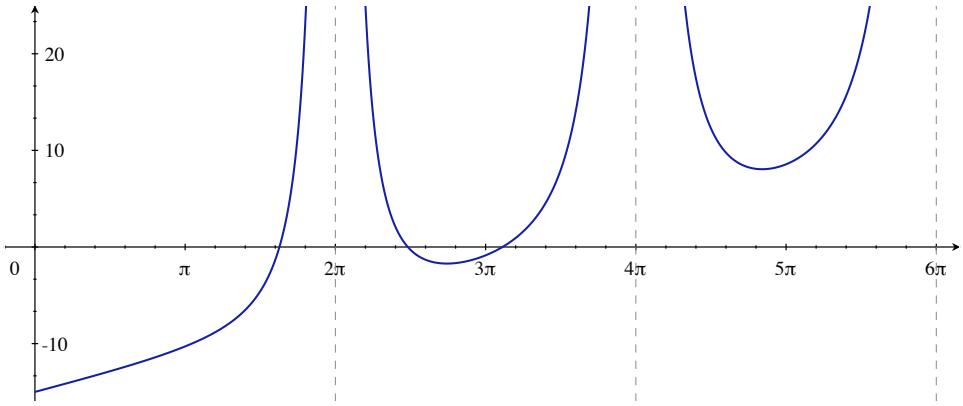


FIGURE 9.4. The graph of the function $\chi_{\rho,z}$ for $\rho = 2$ and $z = 15$. The function is not defined at $\phi = 2k\pi$ for $k \in \mathbb{Z}_{>0}$. In this case, there are exactly three geodesics joining the origin to any point p with coordinates $[\rho\cos\theta, \rho\sin\theta, z, 1]$.

$\chi_{z,\rho}(\phi) = 0$ admits a solution in this interval, then $\{\phi_n\}$ will converge toward the first such solution. Otherwise, either $\{\phi_n\}$ escapes the interval $(2\pi, 4\pi)$, or the sign of the derivative $d\chi_{z,\rho}/d\phi$ becomes positive. Either case is a halting condition for our algorithm. Repeating this procedure starting with a point for which $d\chi_{z,\rho}/d\phi$ is positive, we find the other solution in the interval. Depending on the level of precision that we want for lighting, we can repeat the procedure on the next intervals $(4\pi, 6\pi)$, $(6\pi, 8\pi)$, ...

Assume now that $p = [x, y, 0, 1]$ lies in the xy-plane. Then there is a unique geodesic γ joining o to p. It coincides with the euclidean geodesic of $\mathbb{R}^2$ between the same points. Hence its direction and length can be computed explicitly. Alternatively, using continuity, we can extend the definition of the previous function $\chi_{\rho,z}$ at $\phi = 0$ by letting $\chi_{\rho,z}(0) = -z$. In this way, this particular case is included in the previous discussion. Indeed the only zero of $\chi_{\rho,0}$ is $\phi = 0$.

If $p = [0, 0, z, 1]$ lies on the z-axis, then the path $\gamma(t) = [0, 0, t, 1]$ is a geodesic from o to p with initial direction $v = [0, 0, 1, 0]$ and length $t = z$. If $2n\pi \leqslant z < 2n\pi + 2\pi$, for some integer $n \geqslant 1$, then o and p are joined by n other rotation-invariant families of geodesics $\{\gamma_{1,\alpha}\}, \ldots, \{\gamma_{n,\alpha}\}$ where α runs over $[0, 2\pi)$. The kth of these has length

$$t_{k,\alpha} = 2k\pi\sqrt{\frac{z}{k\pi} - 1}$$

and direction at the origin

$$v_{k,\alpha} = \left[\sqrt{\frac{z-2k\pi}{z-k\pi}}\cos(\alpha), \sqrt{\frac{z-2k\pi}{z-k\pi}}\sin(\alpha), \sqrt{\frac{k\pi}{z-k\pi}}, 0\right].$$

9.6. **Distance underestimator for a ball.** As mentioned in Section 2.2, we don't necessarily need the exact distance to an object to perform ray-marching. A distance underestimator also works. A rough estimate using the solution of the geodesic flow shows the following.

Lemma 9.5. *Let $f\colon \mathbb{R}_+ \to \mathbb{R}_+$ be the continuous increasing map defined by*

$$f(d) = \begin{cases} d, \text{ if } d < \sqrt{6}, \\ \frac{4}{3}\left(1 + \frac{1}{12}d^2\right)^{3/2}, \text{ if } \sqrt{6} \leqslant d < 2\sqrt{6}, \\ \frac{1}{2\sqrt{3}}d^2, \text{ if } 2\sqrt{6} \leqslant d. \end{cases}$$

If $p = [x, y, z, 1]$ is a point at distance d from the origin o, then $\sqrt{x^2+y^2} \leqslant d$ and $|z| \leqslant f(d)$. □

As a consequence, for every $\psi \in (0,1)$, for every $m \geqslant 1$, we have

$$0 < \left[(1-\psi)\big(x^2+y^2\big)^{\frac{m}{2}} + \psi\big(f^{-1}(|z|)\big)^m\right]^{\frac{1}{m}} \leqslant \operatorname{dist}(o,p).$$

This allows us to build a distance underestimator $\sigma'\colon X \to \mathbb{R}$ to render a ball of radius r centered at o, as follows. Let

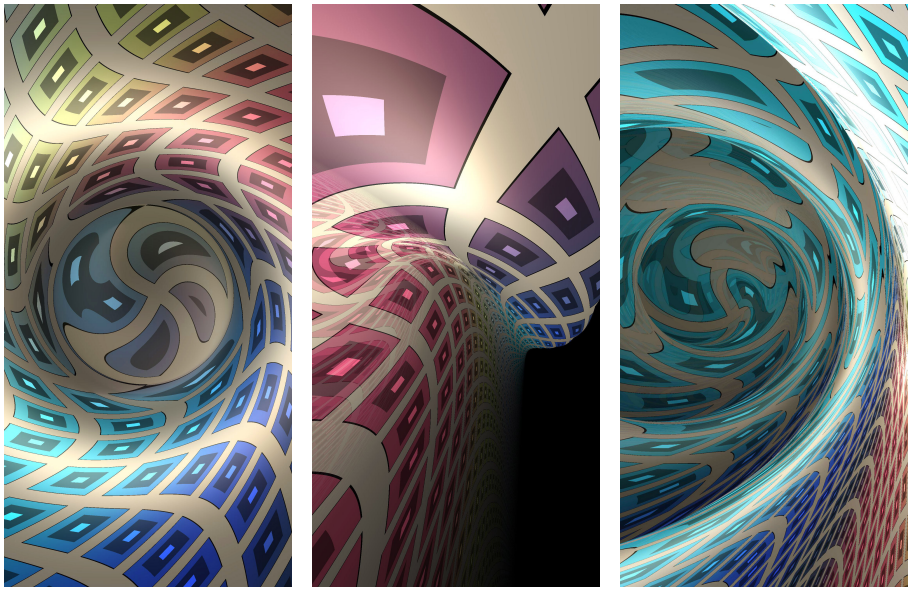
$$\sigma'(p) = \begin{cases} \sigma(p) - r, \text{ if } \sigma(p) > r + \eta \\ \operatorname{dist}(o,p) - r, \text{ otherwise} \end{cases}$$

where

$$\sigma(p) = \left[(1-\psi)\big(x^2+y^2\big)^{\frac{m}{2}} + \psi\big(f^{-1}(|z|)\big)^m\right]^{\frac{1}{m}}$$

and $\eta > 0$ is a constant that is much larger than the threshold ϵ used to stop the ray-marching algorithm. This is more efficient than the exact signed distance function: here, the rough (and inexpensive to calculate) estimate σ is used to handle points at a large distance from the ball. When the point p is close to the ball we replace this estimate by the exact distance computed numerically as explained in Section 9.5. We use this distance underestimator to render the balls in Figure 5.13 (a line of balls along the fiber direction), and Figure 9.12b (a lattice of balls in Nil).

9.7. **Creeping to horizontal half-spaces.** In the case of vertical objects, we can use the geometry of Nil to help us build signed distance functions. For "horizontal" objects, for example the $z \leqslant 0$ half-space, we do not have anything equivalent. Thus, it is difficult to come up with a signed distance function (or even a distance underestimator). However, we can still detect whether a point is in a half-space or not, and so we can use the same binary search algorithm as used to detect the boundary of a fundamental domain in Section 4.2.1. Figure 9.5 shows the $z \leqslant 0$ half-space in Nil geometry, with boundary textured by squares in the coordinate grid of side length one.



(A) $p = [0, 0, 8.5, 1]$ (B) $p = [5, 0, 3, 1]$ (C) $p = [0, 5, 15, 1]$

FIGURE 9.5. The $z \leqslant 0$ half-space in Nil geometry, viewed from the point p. Rendered with artificial (constant) light intensity, and fog.

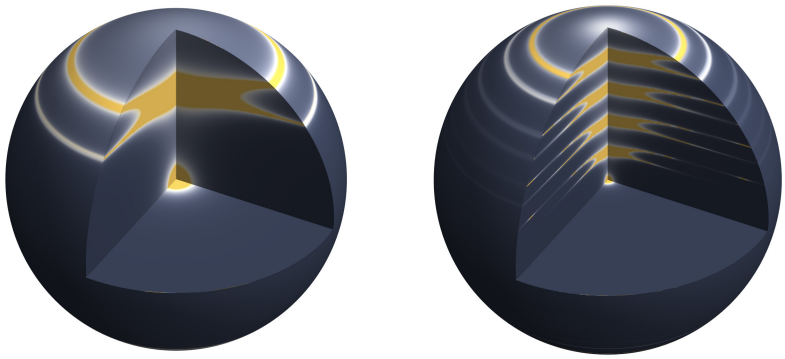
9.8. **Lighting.** We addressed the calculation of lighting pairs in Section 9.5. Here, we calculate the intensity $I(r, u)$ experienced from an isotropic light source traveling a distance r with initial tangent u. Recall that this is inversely proportional to the area density $\mathcal{A}(r, u)$. Here we calculate this area density directly by taking the derivative of the geodesic flow as in Equation (5.9).

A parameterization of the speed geodesic starting at the origin $o = \mathbf{e}_4$ with arc length parameter r in the direction $u = [a \cos \alpha, a \sin \alpha, c, 0] \in T_o\text{Nil}$ is given by Equation (9.1) for the generic case (when $c \neq 0$) and by

Equation (9.2) for geodesics in the xy-plane. Below we concern ourselves with the generic case. Let (L, z, α) be the cylindrical coordinates on $T_o\mathrm{Nil}$ with (L, z) the norm of the projections onto the xy plane and z axis respectively, and $\alpha \in [0, 2\pi)$ measured from the positive x axis. In these coordinates the point $ru \in T_o\mathrm{Nil}$ is expressed $(L, z, \alpha) = (ra, rc, \alpha)$. Thus, using Equation (5.13) we may calculate the area density in terms of the L, z and α derivatives of Equation (9.1).

$$\mathcal{A} = \frac{2r^2}{z^4}\left|\sin\frac{z}{2}\right|\left|L^2 z\cos\frac{z}{2} - 2r^2\sin\frac{z}{2}\right|. \tag{9.6}$$

See Figure 9.6. As with the computation of the geodesic flow in Section 9.3, to obtain correct lighting along the xy plane direction, one should use the asymptotic expansion of Equation (9.6) around $z = 0$.

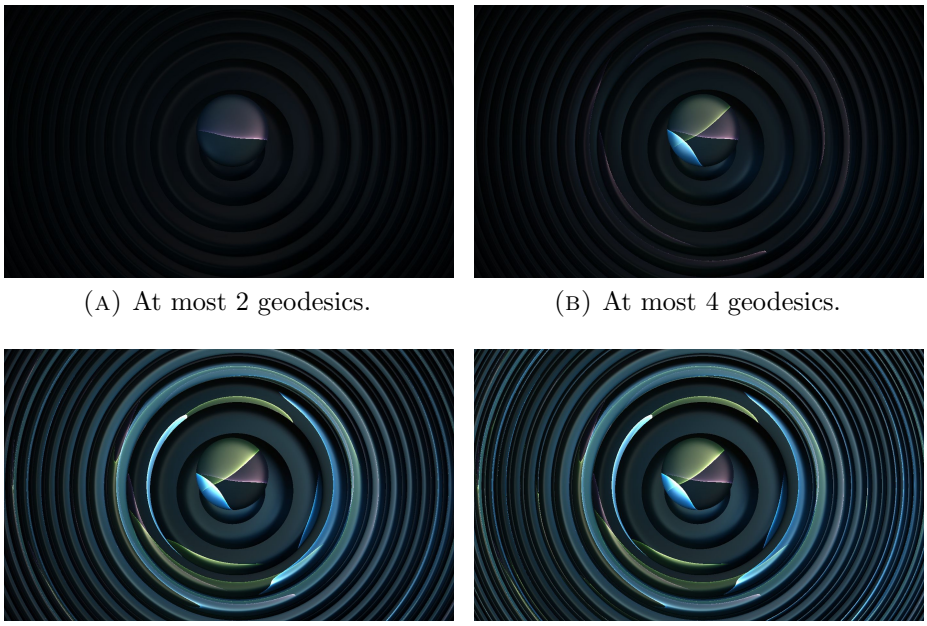


(A) Within a ball of radius 10. (B) Within a ball of radius 30.

FIGURE 9.6. The lighting intensity function $I(r, u)$ in Nil geometry.

In horizontal directions, the light intensity quickly drops away. Near the vertical axis, the intensity of a light source periodically blows up as geodesics reconverge. See Figures 9.7, 9.8, 9.9, and 9.10.

9.9. **Discrete subgroups and fundamental domains.** The compact Nil manifolds are circle bundles over euclidean two-orbifolds with non-zero Euler class [Sco83, Theorem 4.17]. The simplest example of a Nil manifold can also be seen as the suspension M of a regular two-torus T by a Dehn twist. The fundamental group Γ of M is a lattice in G. We explain here with a concrete example how to construct a fundamental domain D for the action on Γ on X.



(A) At most 2 geodesics. (B) At most 4 geodesics.

(C) At most 8 geodesics. (D) At most 16 geodesics.

FIGURE 9.7. A line of balls in Nil along the z-axis, lit by three light sources (cyan, yellow, and magenta) far behind the viewer, using correct light intensity. Compare with Figure 5.13, which is rendered with constant light intensity. As almost every point is reached by finitely many geodesics, one may render accurate lighting for any compact region of Nil by computing a sufficiently large number of possible directions.

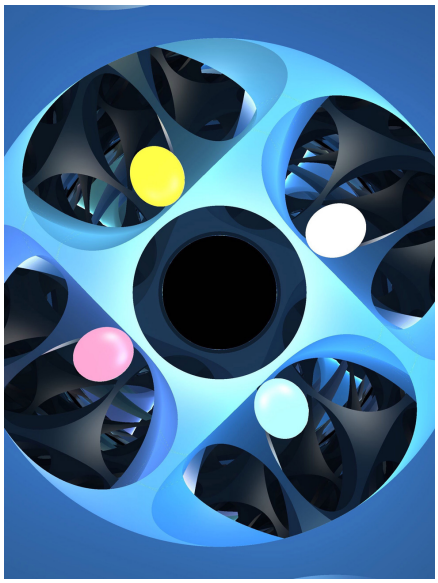
Let f be the Dehn-twist of the standard two-torus $T = \mathbb{R}^2/\mathbb{Z}^2$ with action given by the matrix

$$\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}.$$

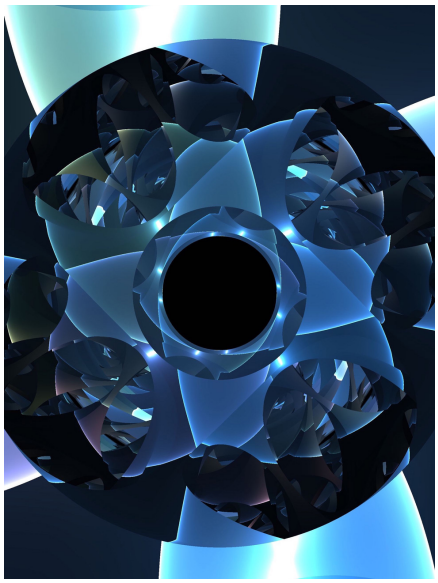
Consider the Dehn-twist torus bundle which is the mapping torus of T with monodromy f. Its fundamental group Γ has presentation

$$\Gamma = \langle A, B, C \mid [A, B] = C, [A, C] = 1, [B, C] = 1 \rangle.$$

Here A and C can be interpreted as the standard generators of $\pi_1(T) \cong \mathbb{Z}^2$. The conjugation by B is the automorphism of $\mathbb{Z}^2$ induced by f. Note that C is central, hence corresponds to the loop along which we are performing our Dehn twist. The group Γ is actually generated by A and B only. Nevertheless it is more convenient to keep three generators as they represent translations in three independent directions. The group Γ can be identified with the discrete Heisenberg group, that is



(A) Near the lights (one unit in front of viewer, in the z-direction).



(B) Far from the lights (seven units behind the viewer, in the z-direction).

FIGURE 9.8. Four lights (white, yellow, cyan, and magenta) illuminate a tiling of Nil in the style of Figure 2.1b. Far away, there are curves of high intensity light caused by the convergence of one-parameter families of geodesics. The scene does not cast shadows in these images.

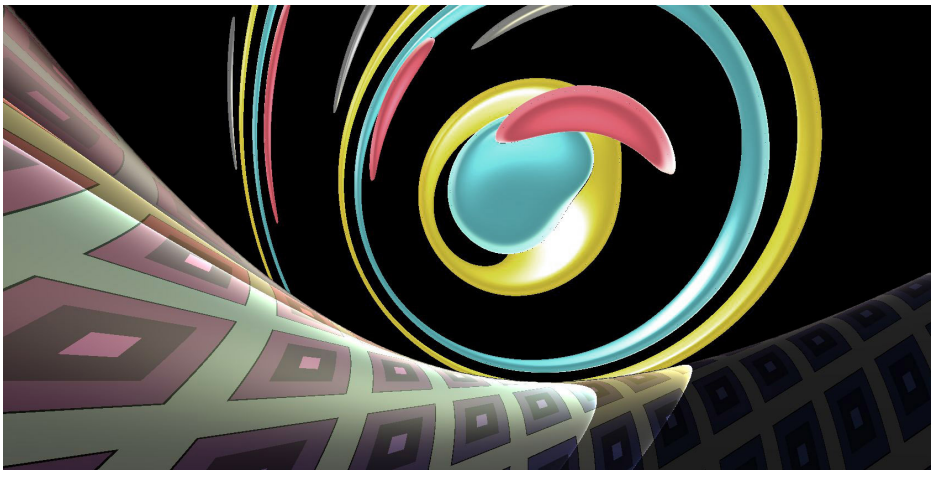
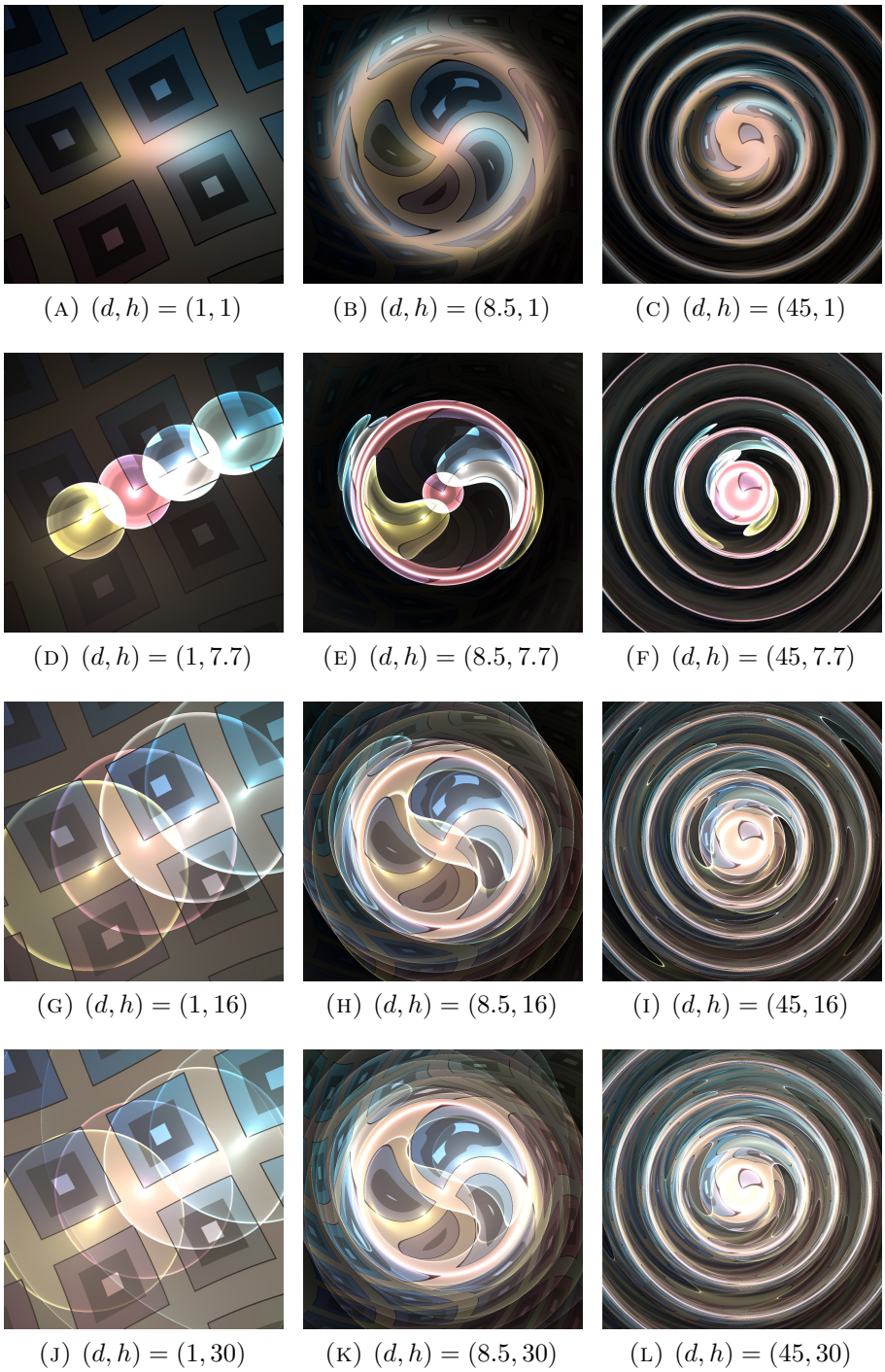


FIGURE 9.9. Sunset in Nil. When the light intensity blows up far away from the light source, it may illuminate distant parts of an otherwise dark object. Standing at such a location, the distant light sources appear large in the sky.



(A) $(d, h) = (1, 1)$ (B) $(d, h) = (8.5, 1)$ (C) $(d, h) = (45, 1)$

(D) $(d, h) = (1, 7.7)$ (E) $(d, h) = (8.5, 7.7)$ (F) $(d, h) = (45, 7.7)$

(G) $(d, h) = (1, 16)$ (H) $(d, h) = (8.5, 16)$ (I) $(d, h) = (45, 16)$

(J) $(d, h) = (1, 30)$ (K) $(d, h) = (8.5, 30)$ (L) $(d, h) = (45, 30)$

FIGURE 9.10. Four lights illuminate the $z \leqslant 0$ half-space in Nil. The viewer is above the plane at position $[0, 0, d, 1]$, and the light sources are positioned at $[k/2, 0, h, 1]$ for $k \in \{-1, 0, 1, 2\}$. We use correct lighting with up to three geodesics, and no fog.

the set of points with integer coordinates in the Heisenberg model of Nil. Concretely, A, B, and C are the elements of Nil whose coordinates in X are

$$A = [1, 0, 0, 1], \quad B = [0, 1, 0, 1], \quad \text{and} \quad C = [0, 0, 1, 1].$$

Observe that via the projection $\pi\colon X \to \mathbb{E}^2$, every element of Γ induces an isometry of $\mathbb{E}^2$: A and B correspond to translations along the x- and y-axis respectively, while C acts trivially on $\mathbb{E}^2$. It follows that the "cube"

$$D = [-1/2, 1/2]^3 \times \{1\}$$

is a fundamental domain for the action of Γ on X. Note that A, B, and C do not directly pair the square sides of the "cube". See Remark 9.7. Our rotation-invariant model X for Nil is also a projective model. The fundamental domain D can be see as the intersection of a collection of half-spaces $H_x^{\pm}$, $H_y^{\pm}$, $H_z^{\pm}$ as described in Section 4.1.2. Here

$$H_x^- = \{x \geqslant -1/2\} \quad \text{and} \quad H_x^+ = \{x \leqslant 1/2\}.$$

while $H_y^{\pm}$ and $H_z^{\pm}$ are defined in the same way. The teleporting algorithm has two main steps. Let $p = [x, y, z, 1]$ be a point in X.

(1) If p does not belong to H_x^- (respectively H_x^+, H_y^- H_y^+), then we move it by A (respectively A^{-1}, B, B^{-1}). After finitely many steps, the new point p will lie in

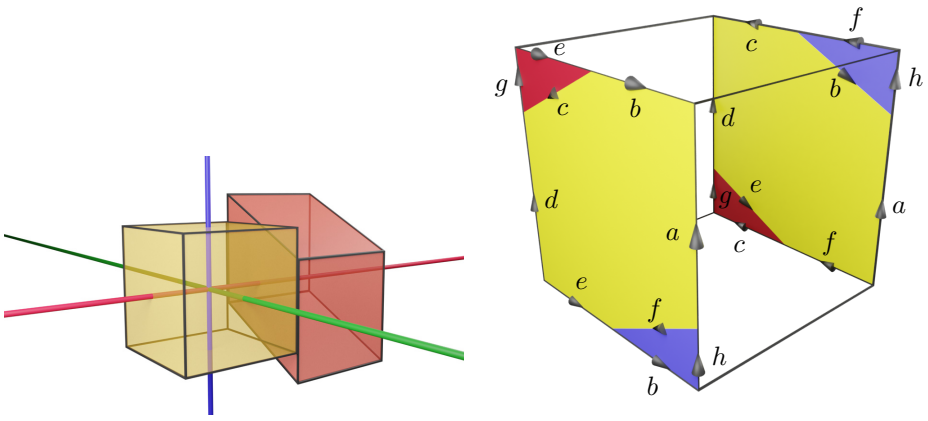
$$H_x^- \cap H_x^+ \cap H_y^- \cap H_y^+.$$

The isometries of $\mathbb{E}^2$ induced by A and B commute, so we don't pay attention to the order in which we perform these operations.

(2) Once this is done, if p does not belong to H_z^- (respectively H_z^+), then we move it by C (respectively C^{-1}). Note that C does not affect the xy-coordinates of p. Therefore, after this process, p lies in D.

Remark 9.7. Note that the collection of isometries $\{A^{\pm 1}, B^{\pm 1}, C^{\pm 1}\}$ does not provide a face pairing of our fundamental domain D in the sense of Section 4.1. Consider for example the square sides F_x^- and F_x^+ which are the intersections of D with the affine planes ∂H_x^- and ∂H_x^+ respectively. The generator A is a shear, not an affine translation in $\mathbb{R}^4$ along the x-axis. See Figure 9.11a. Thus it does not map F_x^- to F_x^+. In order to get a proper face pairing, one must subdivide the sides of D and increase the number of generators. This is illustrated on Figure 9.11b. We draw the one-skeleton of D and color the sides F_x^- (on the left) and F_x^+ (on the right). The yellow (respectively blue, red) face in F_x^- is mapped bijectively to the face with the same color in

F_x^+ via A (respectively AC, AC^{-1}). A similar subdivision can be found



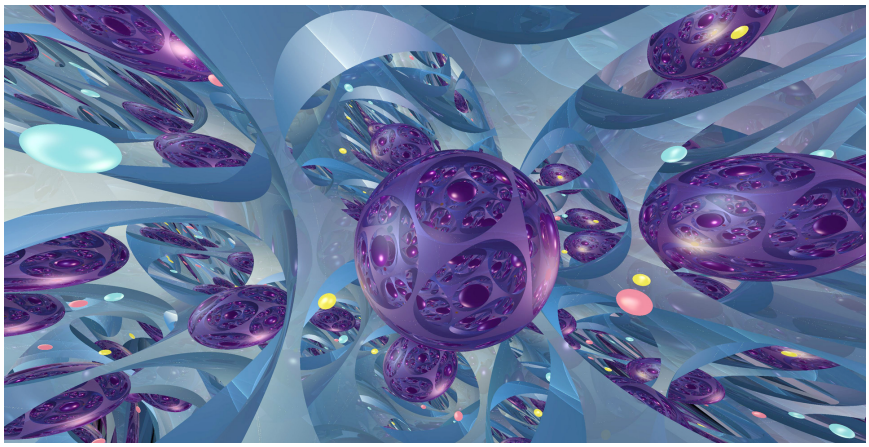
(A) The fundamental domain D (yellow) and its image (red) under the shear A. The red, green, and blue lines correspond to the x-, y-, and z-axes in our model of Nil.

(B) The yellow, blue, and red faces are in one-to-one correspondence via A, AC, and AC^{-1}. The decorations indicate the edge identifications induced by A and C only.

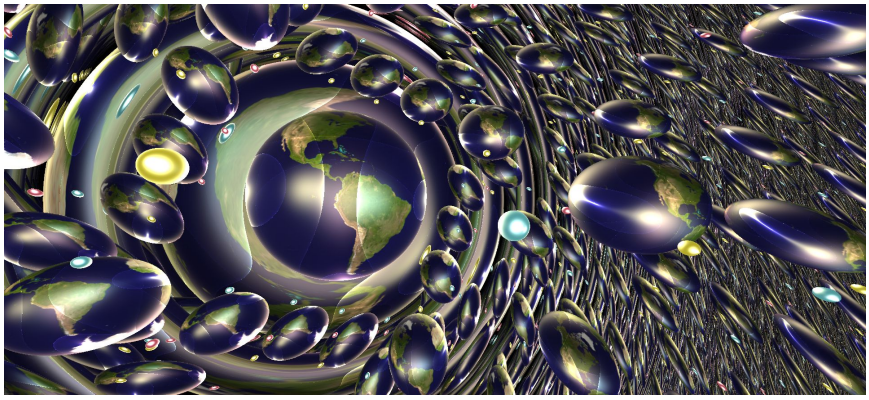
FIGURE 9.11. Face pairing in Nil.

for the sides $F_y^- = D \cap \partial H_y^-$ and $F_y^+ = D \cap \partial H_y^+$. (No subdivision of the horizontal faces of D is needed as C is an affine translation along the z-axis.) As explained in Remark 4.3 and illustrated by the above algorithm, when using a fundamental domain defined as the intersection of projective half-spaces, we do not need a proper face pairing to implement teleportation. ◊

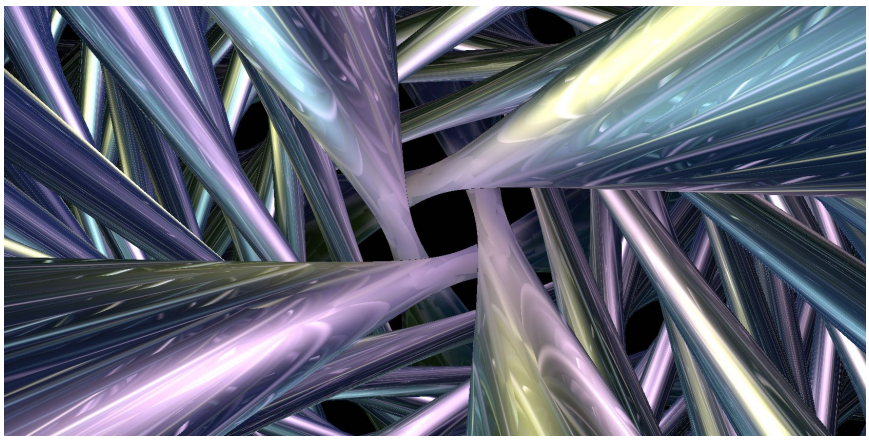
In Figure 9.12, we show the in-space view for various scenes in Nil geometry. Figure 9.12a shows the Dehn-twist torus bundle with monodromy f as in Section 9.9, with a fundamental domain drawn in the style of Figure 2.1b. Figure 9.12b shows a lattice of spheres, textured as the Earth, lit by a corresponding lattice of light sources. Figure 9.12c shows solid cylinders (which we implement as vertical objects) around fibers of Nil. Compare with Figure 8.2.



(A) The Dehn-twist torus bundle with monodromy f.



(B) Lattice of balls.



(C) Fibers.

FIGURE 9.12. Nil Geometry.

10. $\widetilde{\mathrm{SL}}(2,\mathbb{R})$

10.1. **Model.** We identify the space $\mathcal{M}_{2,2}(\mathbb{R})$ of 2×2-matrices with $\mathbb{R}^4$ via the basis $E = (E_0, E_1, E_2, E_3)$ given by

$$E_0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad E_1 = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix},$$
$$E_2 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad E_3 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}.$$

The quadratic form $k = -\det$ is diagonal in this basis: given any point $p = [p_0, p_1, p_2, p_3]$ in $\mathbb{R}^4$, we have

$$k(p) = -p_0^2 - p_1^2 + p_2^2 + p_3^2.$$

In particular $\mathrm{GL}(2,\mathbb{R})$ and $\mathrm{SL}(2,\mathbb{R})$ correspond to the subsets

$$\mathcal{Q}_0 = \{p \in \mathbb{R}^4 \mid k(p) \neq 0\} \quad \text{and} \quad \mathcal{Q} = \{p \in \mathbb{R}^4 \mid k(p) = -1\}$$

of $\mathbb{R}^4$. We choose for the origin the point $o = [1, 0, 0, 0]$. This corresponds to the identity. The group law can be rewritten as follows: given a point $p = [p_0, p_1, p_2, p_3]$ in $\mathcal{Q}_0$, the corresponding element of $\mathrm{GL}(2,\mathbb{R})$ acts on $\mathcal{Q}_0$ as the matrix

$$\begin{bmatrix} p_0 & -p_1 & p_2 & p_3 \\ p_1 & p_0 & p_3 & -p_2 \\ p_2 & p_3 & p_0 & -p_1 \\ p_3 & -p_2 & p_1 & p_0 \end{bmatrix}.$$

We endow $\mathcal{Q}_0$ with an $\mathrm{GL}(2,\mathbb{R})$-invariant riemannian metric:

$$ds^2 = \frac{4\beta_0(p)}{k(p)^2}\left(dp_0^2 + dp_1^2 + dp_2^2 + dp_3^2\right) - \frac{4\beta_1(p)}{k(p)^2}(dp_0 dp_2 - dp_1 dp_3) - \frac{4\beta_2(p)}{k(p)^2}(dp_0 dp_3 + dp_1 dp_2),$$

where

$$\begin{cases} \beta_0(p) = p_0^2 + p_1^2 + p_2^2 + p_3^3 \\ \beta_1(p) = p_0 p_2 - p_1 p_3 \\ \beta_2(p) = p_0 p_3 + p_1 p_2. \end{cases}$$

It turns out that the level sets of k are totally geodesic subspaces of $\mathcal{Q}_0$. The stabilizer $K < G$ of the origin $o \in \mathcal{Q}$ is generated by:

- *rotations* R_α of angle α, with matrix
$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \cos\alpha & -\sin\alpha \\ 0 & 0 & \sin\alpha & \cos\alpha \end{bmatrix}, \text{ and}$$
- the *flip* F, with matrix
$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

Observe that $F \circ R_\alpha \circ F^{-1} = R_{-\alpha}$, so K is isomorphic to $\mathrm{O}(2)$.

The space we are really interested in is not $\mathrm{SL}(2,\mathbb{R})$, but its *universal cover*. Topologically, the latter is a line bundle over $\mathbb{H}^2$. The identification goes as follows. Consider the adjoint representation of $\mathrm{SL}(2,\mathbb{R})$ on its Lie algebra
$$\mathfrak{sl}_2 = \{M \in \mathcal{M}_{2,2}(\mathbb{R}) \mid \mathrm{Tr}(M) = 0\}.$$
This action preserves the Killing quadratic form
$$\mathfrak{K}(M) = \frac{1}{2}\mathrm{Tr}(M^2)$$
which has signature $(2,1)$. Hence it induces an action by isometries on the hyperboloid model of $\mathbb{H}^2$. In our context, the Lie algebra $\mathfrak{sl}_2$ is isomorphic to the linear space $T_o\mathcal{Q} \subset \mathbb{R}^4$ spanned by
$$e_x = -E_3, \quad e_y = E_2, \quad \text{and} \quad e_z = E_1.$$
The Killing form is diagonal in this basis: if $M = xe_x + ye_y + ze_z$, then $\mathfrak{K}(M) = x^2 + y^2 - z^2$. So we choose for the hyperboloid model of $\mathbb{H}^2$ the set $\mathcal{H}$ as defined in Section 8.1:
$$\mathcal{H} = \big\{[x,y,z] \in T_o\mathcal{Q} \mid x^2 + y^2 - z^2 = -1 \text{ and } z > 0\big\}.$$
We define a 1-Lipschitz, $\mathrm{SL}(2,\mathbb{R})$-equivariant projection $\pi\colon \mathrm{SL}(2,\mathbb{R}) \to \mathbb{H}^2$ by sending the origin o to the point $[0,0,1] \in \mathcal{H}$. (The scaling factor four in the metric on $\mathcal{Q}_0$ was precisely chosen so that the best Lipschitz constant for π is one.) The fiber of the point $q = [x,y,z]$ is a circle parametrized as follows.
$$\pi^{-1}(q) = \{S_w\varsigma(q) \mid w \in [0, 4\pi)\},$$
where $\varsigma\colon \mathcal{H} \to \mathcal{Q}$ is the section given by
$$\varsigma(q) = \left[\sqrt{\frac{z+1}{2}}, 0, \frac{x}{\sqrt{2(z+1)}}, \frac{y}{\sqrt{2(z+1)}}\right]$$

and S_w is the transformation of $\mathcal{Q}$ with matrix

$$\begin{bmatrix} \cos\left(\frac{w}{2}\right) & -\sin\left(\frac{w}{2}\right) & 0 & 0 \\ \sin\left(\frac{w}{2}\right) & \cos\left(\frac{w}{2}\right) & 0 & 0 \\ 0 & 0 & \cos\left(\frac{w}{2}\right) & \sin\left(\frac{w}{2}\right) \\ 0 & 0 & -\sin\left(\frac{w}{2}\right) & \cos\left(\frac{w}{2}\right) \end{bmatrix}.$$

Note that S_w translates points along the fiber by an angle $w/2$, not w. This accounts for the fact that the map $\mathrm{SL}(2,\mathbb{R}) \to \mathrm{SO}(2,1)$ is a two-sheeted cover. Finally one observes that the projection

$$\begin{array}{rccc} \lambda\colon & \mathcal{H}\times\mathbb{R} & \to & \mathcal{Q} \\ & (q,w) & \mapsto & S_w\varsigma(q) \end{array}$$

is a covering map, providing an identification between $\widetilde{\mathrm{SL}}(2,\mathbb{R})$ and our model space $X = \mathcal{H}\times\mathbb{R}$. We call the factor $\mathcal{H}$ (respectively $\mathbb{R}$) the *horizontal* (respectively *vertical* or *fiber*) component of X.

Remark 10.1. In practice, we adopt a slightly different point of view. We store a point $p \in X$ as a pair $(g,w) \in \mathrm{SL}(2,\mathbb{R})\times\mathbb{R}$ where g is the image of p by the covering map λ and w is the fiber component of p. This representation is redundant, but allows us to go quickly back and forth between $\mathrm{SL}(2,\mathbb{R})$ and its universal cover. ◊

We choose as a base point of X the point $\widetilde{o} = [0,0,1,0]$ which is a pre-image of o. The covering map λ induces an isomorphism between the stabilizer of $\widetilde{o}$ and the stabilizer of o, that is $\mathrm{O}(2)$.

- We choose a lift $\widetilde{R}_\alpha$ of R_α with the following properties. It fixes the fiber component, and acts on the horizontal component as the usual rotation of $\mathbb{H}^2$ by angle α centered at $\pi(o)$. Beware that R_α is a rotation of our model space $\mathcal{Q}$ of $\mathrm{SL}(2,\mathbb{R})$ which is distinct from the element of $\mathrm{SL}(2,\mathbb{R})$ representing a rotation of $\mathbb{H}^2$.
- The map $\widetilde{F}$ sending $p = [x,y,z,w]$ of X to $p' = [y,x,z,-w]$ is a lift of F.

10.2. **Geodesic flow and parallel transport in** $\mathrm{SL}(2,\mathbb{R})$**.** The solution of the geodesic flow has been computed in [DESS09]. We follow a slightly more geometric approach.

Since the covering map is a local isometry, the geodesics of X are lifts of geodesics in $\mathcal{Q}$. Hence we first integrate the geodesic flow in $\mathcal{Q}$ using Grayson's method. We endow the tangent space $T_{\widetilde{o}}X$ with the reference frame $\widetilde{e} = (\widetilde{e}_x, \widetilde{e}_y, \widetilde{e}_w)$, where

$$\widetilde{e}_x = \frac{\partial}{\partial x}, \quad \widetilde{e}_y = \frac{\partial}{\partial y}, \quad \text{and} \quad \widetilde{e}_w = \frac{\partial}{\partial w}.$$

We write $e = (e_x, e_y, e_w)$ for its image under $d_{\tilde{o}}\lambda \colon T_{\tilde{o}}X \to T_o\mathcal{Q}$. (Note that e_x and e_y coincide with the previous definition.) It follows from our choice of metric that e is an orthonormal basis of $T_o\mathcal{Q}$.

Let $\gamma \colon \mathbb{R} \to \mathcal{Q}$ be a geodesic in $\mathrm{SL}(2,\mathbb{R})$ and let $T(t) \colon T_{\gamma(0)}\mathcal{Q} \to T_{\gamma(t)}\mathcal{Q}$ be the corresponding parallel-transport operator. As in Sections 3.2.1 and 3.4.1, we define paths $u \colon \mathbb{R} \to T_o\mathcal{Q}$ and $Q \colon \mathbb{R} \to \mathrm{SO}(3)$ by the relations

$$\begin{aligned} \dot{\gamma}(t) &= d_o L_{\gamma(t)} u(t), \text{ and} \\ T(t) \circ d_o L_{\gamma(0)} &= d_o L_{\gamma(t)} Q(t). \end{aligned}$$

After some computation, Equations (3.2) and (3.6) can be written relative to the basis e as follows

$$\begin{cases} \dot{u}_x = 2u_y u_w \\ \dot{u}_y = -2u_x u_w \\ \dot{u}_w = 0 \end{cases}$$

and

$$\dot{Q} + BQ = 0, \quad \text{where} \quad B = \frac{1}{2}\begin{bmatrix} 0 & 3u_w & u_y \\ -3u_w & 0 & -u_x \\ -u_y & u_x & 0 \end{bmatrix}.$$

For the initial condition $u(0) = a\cos(\alpha)e_x + a\sin(\alpha)e_y + ce_w$, where $a \in \mathbb{R}_+$ and $c \in \mathbb{R}$ satisfy $a^2 + c^2 = 1$, one gets

$$u(t) = a\cos(\alpha - 2ct)e_x + a\sin(\alpha - 2ct)e_y + ce_w.$$

In order to calculate the expression for Q, we follow the strategy detailed above and obtain

$$Q(t) = dR_\alpha e^{-2ctU_1} P e^{\frac{1}{2}tU_2} P^{-1} dR_\alpha^{-1}, \quad \forall t \in \mathbb{R},$$

where

$$U_1 = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad U_2 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix},$$

and

$$dR_\alpha = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad P = \begin{bmatrix} a & 0 & -c \\ 0 & 1 & 0 \\ c & 0 & a \end{bmatrix}.$$

Note that $dR_\alpha \colon T_o\mathcal{Q} \to T_o\mathcal{Q}$ is the differential at o of the rotation R_α, written in the frame $e = (e_x, e_y, e_w)$.

Let us now move back to the original geodesic $\gamma\colon \mathbb{R} \to X$. Equation (3.1) becomes $\dot\gamma(t) = A(t)\gamma(t)$, where

$$A(t) = \frac{1}{2}\begin{bmatrix} 0 & -u_w & u_x & u_y \\ u_w & 0 & -u_y & u_x \\ u_x & -u_y & 0 & u_w \\ u_y & u_x & -u_w & 0 \end{bmatrix}.$$

Using a change of variables, one can reformulate the previous equation into a first-order differential system with *constant* coefficients that we integrate with standard methods. We obtain that the geodesic γ, such that $\gamma(0) = o$ and $\dot\gamma(0) = a\cos(\alpha)e_x + a\sin(\alpha)e_y + ce_w$, decomposes (up to a rotation) as a product of two one-parameter subgroups:

$$\gamma(t) = R_\alpha(\eta(t) * \xi(t)). \tag{10.2}$$

As before, R_α is the rotation of $\mathcal{Q}$ by angle α and $*$ is group multiplication in $\mathrm{SL}(2,\mathbb{R})$. The *spin* factor $\xi\colon \mathbb{R} \to \mathrm{SL}(2,\mathbb{R})$ represents a rotation of $\mathbb{H}^2$ fixing the origin $\pi(o) \in \mathcal{H}$. It can be written in $\mathcal{Q}$ as

$$\xi(t) = [\cos(ct), \sin(ct), 0, 0].$$

The *translation* factor $\eta\colon \mathbb{R} \to \mathrm{SL}(2,\mathbb{R})$ can have three forms, corresponding to the three types of isometries of $\mathbb{H}^2$. For simplicity we let $\kappa = \sqrt{|c^2 - a^2|}$.

- If $c > a$, then η is an *elliptic* transformation, given in $\mathcal{Q}$ by
$$\eta(t) = \left[\cos\left(\frac{\kappa t}{2}\right), -\frac{c}{\kappa}\sin\left(\frac{\kappa t}{2}\right), \frac{a}{\kappa}\sin\left(\frac{\kappa t}{2}\right), 0\right].$$
- If $c = a$, then η is a *parabolic* transformation, given in $\mathcal{Q}$ by
$$\eta(t) = \left[1, -\frac{t}{2\sqrt{2}}, \frac{t}{2\sqrt{2}}, 0\right].$$
- If $c < a$, then η is a *hyperbolic* transformation, given in $\mathcal{Q}$ by
$$\eta(t) = \left[\cosh\left(\frac{\kappa t}{2}\right), -\frac{c}{\kappa}\sinh\left(\frac{\kappa t}{2}\right), \frac{a}{\kappa}\sinh\left(\frac{\kappa t}{2}\right), 0\right].$$

10.3. **Passing to the universal cover.** Let us now consider the geodesic $\widetilde\gamma$ in the universal cover $X = \mathcal{H} \times \mathbb{R}$ starting at $\widetilde o$ with initial velocity $a\cos(\alpha)\widetilde e_x + a\sin(\alpha)\widetilde e_y + c\widetilde e_w$. This is a lift of the geodesic γ computed above. The horizontal component of $\widetilde\gamma$ is obtained as the image of γ under the projection $\pi\colon \mathrm{SL}(2,\mathbb{R}) \to \mathbb{H}^2$. Note that the spin factor $\xi(t)$ fixes the base point $\pi(o) \in \mathcal{H}$. Moreover, the rotation R_α of $\mathcal{Q}$ induces (via the projection π) the rotation r_α of $\mathbb{H}^2$ by angle α centered at $\pi(o)$. Consequently, $\pi \circ \gamma(t)$ is the image under r_α of one

of the following points, depending on whether $c > a$, $c = a$, or $c < a$ respectively:

$$\begin{bmatrix} \frac{2a}{\kappa}\sin\left(\frac{\kappa t}{2}\right)\cos\left(\frac{\kappa t}{2}\right) \\ -\frac{2ac}{\kappa^2}\sin^2\left(\frac{\kappa t}{2}\right) \\ 1+\frac{2a^2}{\kappa^2}\sin^2\left(\frac{\kappa t}{2}\right) \end{bmatrix}, \quad \begin{bmatrix} \frac{\sqrt{2}}{2}t \\ -\frac{1}{4}t^2 \\ 1+\frac{1}{4}t^2 \end{bmatrix}, \quad \begin{bmatrix} \frac{2a}{\kappa}\sinh\left(\frac{\kappa t}{2}\right)\cosh\left(\frac{\kappa t}{2}\right) \\ -\frac{2ac}{\kappa^2}\sinh^2\left(\frac{\kappa t}{2}\right) \\ 1+\frac{2a^2}{\kappa^2}\sinh^2\left(\frac{\kappa t}{2}\right) \end{bmatrix}.$$

These are parametrizations of orbits under the one-parameter subgroups above. Their images are a circle, a horocycle, and an equidistant curve to a geodesic, respectively.

In order to compute the fiber component of $\widetilde{\gamma}$ it is convenient to introduce *cylindrical coordinates* on $\mathcal{Q}$. Given $\rho \in \mathbb{R}_+$, $\theta \in [0, 2\pi)$ and $w \in [0, 4\pi)$, the point of $\mathcal{Q}$ with cylindrical coordinates $[\rho, \theta, w]$ is

$$\begin{bmatrix} \cosh\left(\frac{\rho}{2}\right)\cos\left(\frac{w}{2}\right) \\ \cosh\left(\frac{\rho}{2}\right)\sin\left(\frac{w}{2}\right) \\ \sinh\left(\frac{\rho}{2}\right)\cos\left(\theta-\frac{w}{2}\right) \\ \sinh\left(\frac{\rho}{2}\right)\sin\left(\theta-\frac{w}{2}\right) \end{bmatrix}.$$

This choice has been made so that the projection π from $\mathrm{SL}(2,\mathbb{R})$ (in cylindrical coordinates) to $\mathbb{H}^2$ (in polar coordinates) is given by $[\rho, \theta, w] \mapsto [\rho, \theta]$. In view of the expression for γ and its projection onto $\mathcal{H}$, we may calculate an expression for the fiber component $w(t)$ of $\widetilde{\gamma}(t)$. This calculation is greatly simplified by the use of polar coordinates. We obtain

$$w(t) = 2ct + 2\phi(t)$$

where $\phi(t)$ is characterized by

$$\tan\phi(t) = \begin{cases} -\dfrac{c}{\kappa}\tan\left(\dfrac{\kappa t}{2}\right), \text{ if } c > a \\ -\dfrac{t}{2\sqrt{2}}, \text{ if } c = a \\ -\dfrac{c}{\kappa}\tanh\left(\dfrac{\kappa t}{2}\right), \text{ if } c < a \end{cases}$$

Observe that if $c \leqslant a$, then $\phi(t) \in (-\pi/2, \pi/2)$. Therefore its value can be computed from the above equation using the standard arctan function. On the other hand, if $c > a$, then the geodesic $\widetilde{\gamma}$ spirals, and the value of $\phi(t)$ needs to be adjusted by the correct multiple of 2π.

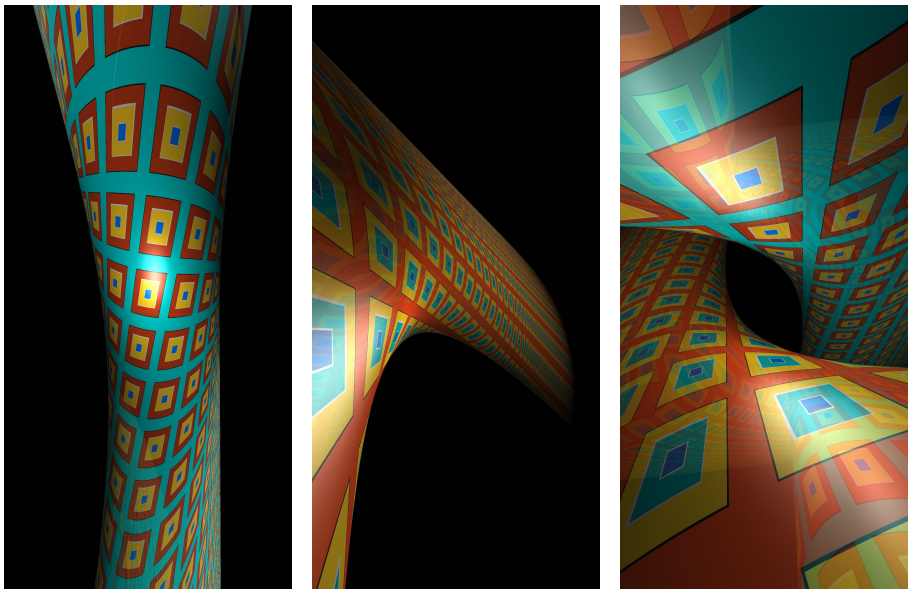
Note that the covering map $\lambda\colon X \to \mathcal{Q}$ is a local isometry. Hence the parallel transport operator in X can be obtained by lifting the parallel transport operator in $\mathcal{Q}$. In view of Grayson's method, this

operator is encoded by a local path $\widetilde{Q}\colon \mathbb{R} \to \mathrm{SO}(3)$, see Section 3.4.1. The identification relies on a choice of a preferred frame $\widetilde{e}$ in the tangent space $T_{\widetilde{o}}X$ at the origin. By construction λ is equivariant with respect to the projection $\widetilde{\mathrm{SL}}(2,\mathbb{R}) \to \mathrm{SL}(2,\mathbb{R})$. Moreover it maps $\widetilde{e}$ to our preferred frame e in $T_o\mathcal{Q}$. Thus $\widetilde{Q}$ and Q actually coincide.

10.4. **Distance to a vertical object.** Exactly as in Nil, we say that an object $Z \subset X$ is *vertical* if it is the pre-image of the projection $\pi \circ \lambda\colon X \to \mathbb{H}^2$ of a non-empty subset S of $\mathbb{H}^2$. In this situation, for any point $p \in X$ we have

$$\mathrm{dist}_X(p, Z) = \mathrm{dist}_{\mathbb{H}^2}(\pi \circ \lambda(p), S).$$

Figure 10.1 shows pre-images of a half-space with geodesic boundary in $\mathbb{H}^2$. The boundary of each is patterned with a square grid following the induced euclidean metric on the plane. The grid has side-length $1/2$.



(A) One half-space. (B) One half-space. (C) Two half-spaces.

FIGURE 10.1. Vertical half-spaces in $\widetilde{\mathrm{SL}}(2,\mathbb{R})$ geometry.

10.5. **Exact distance and direction to a point.** The strategy to compute the distance and direction from the origin to an arbitrary point p with cylindrical coordinates $[\rho, \theta, w]$ is similar to the strategy used in Nil. Because of the flip symmetry, we may assume that $w \geqslant 0$. First

assume that $\rho > 0$. Using the solution of the geodesic flow, we observe that the geodesics $\widetilde{\gamma}$ joining $\widetilde{o}$ to p are in one-to-one correspondence with the zeros of a function

$$\phi \mapsto \chi_{\rho,w}(\phi). \tag{10.3}$$

We define this function in Figure 10.2; see Figure 10.3 for its graph.

$$\chi_{\rho,w}(\phi) = \begin{cases} -\frac{1}{2}w + \phi - 2\tan\phi \frac{\cosh(\rho/2)}{\sqrt{\sinh^2(\rho/2) - \tan^2\phi}} \operatorname{arctanh}\left(\frac{\sqrt{\sinh^2(\rho/2) - \tan^2\phi}}{\cosh(\rho/2)}\right), \\ \qquad\qquad \text{if } \phi > -\frac{\pi}{2} \text{ and } |\tan\phi| < \sinh(\rho/2) \\ -\frac{1}{2}w + \phi - 2\tan\phi, \\ \qquad\qquad \text{if } \phi > -\frac{\pi}{2} \text{ and } |\tan\phi| = \sinh(\rho/2) \\ -\frac{1}{2}w + \phi - 2\tan\phi \frac{\cosh(\rho/2)}{\sqrt{\tan^2\phi - \sinh^2(\rho/2)}} \left(\arctan\left(\frac{\sqrt{\tan^2\phi - \sinh^2(\rho/2)}}{\cosh(\rho/2)}\right) - \operatorname{sign}(\tan\phi)\left\lfloor \frac{1}{2} - \frac{\phi}{\pi} \right\rfloor \pi\right), \\ \qquad\qquad \text{if } |\tan\phi| > \sinh(\rho/2) \text{ and } \pi \neq -\frac{\pi}{2} \mod \pi \\ -\frac{1}{2}w + \phi - 2\cosh(\rho/2), \\ \qquad\qquad \text{if } \phi = -\frac{\pi}{2} \mod \pi \end{cases}$$

FIGURE 10.2. The map $\chi_{\rho,w}$. The first regime corresponds to geodesics with a hyperbolic translation factor, the second to geodesics with a parabolic translation factor, and the third and fourth to geodesics with an elliptic translation factor. In Figure 10.3, these are drawn in red, green, and blue respectively.

Observe that along a given geodesic $\widetilde{\gamma}$, the angle ϕ is a decreasing function of the time parameter t. Said differently, when $\widetilde{\gamma}$ is moving up in the fiber direction, then its projection in $\mathbb{H}^2$ turns clockwise. Hence the domain of $\chi_{\rho,w}$ is contained in $\mathbb{R}_-$. Moreover $\chi_{\rho,w}$ is decreasing around $\phi = 0$.

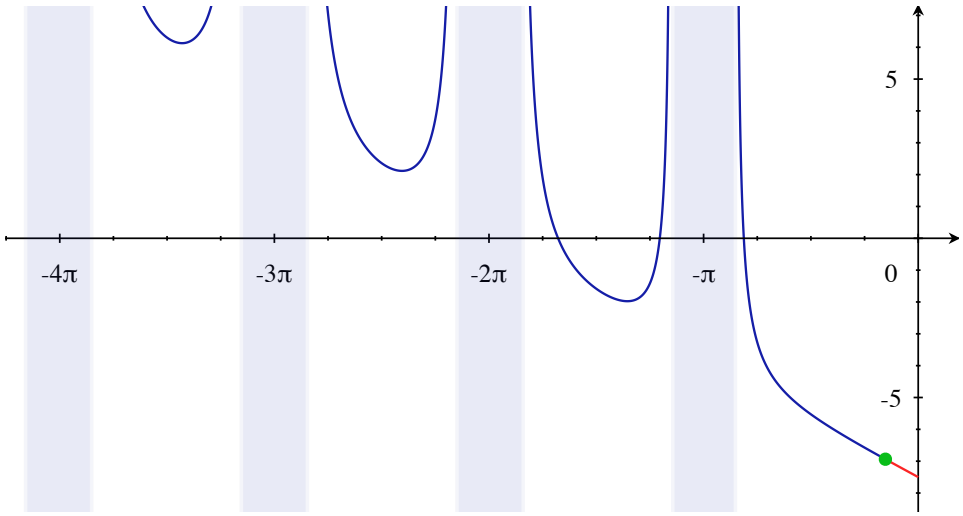


FIGURE 10.3. The graph of the function $\chi_{\rho,w}$ for $\rho = 1$ and $w = 15$. The green dot corresponds to a geodesic whose translation part is parabolic. It separates the geodesics whose translation part are elliptic (dark blue) from those that are hyperbolic (red). The light blue strips indicate the values of ϕ for which $\chi_{\rho,w}$ is not defined. There are exactly three geodesics joining the origin to any point p with cylindrical coordinates $[1, \theta, 15]$.

As in Section 9.5, we compute the zeros of $\chi_{\rho,w}$ using Newton's method, and thus calculate the lighting pairs $\mathcal{L}_{\widetilde{o}}(p)$.

Assume now that $\rho = 0$. The path $\gamma(t) = [0, 0, 1, t]$ is a geodesic from $\widetilde{o}$ to p with initial direction $v = \widetilde{e}_w$ and length $t = w$. If $2n\pi \leqslant w < 2n\pi + 2\pi$, for some integer $n \geqslant 1$, then $\widetilde{o}$ and p are joined by n other rotation-invariant families of geodesics $\{\gamma_{1,\alpha}\}, \ldots, \{\gamma_{n,\alpha}\}$, where α runs over $[0, 2\pi)$. Each geodesic in the kth family has length

$$t_{k,\alpha} = 2k\pi\sqrt{\frac{1}{2}\Big(\frac{w}{2k\pi} + 1\Big)^2 - 1}.$$

Moreover, the initial direction at the origin is characterized by

$$\begin{aligned} d\widetilde{R}_\alpha^{-1} v_{k,\alpha} &= v_{k,0} \\ &= \sqrt{\frac{(w + 2k\pi)^2 - (4k\pi)^2}{2(w + 2k\pi)^2 - (4k\pi)^2}}\widetilde{e}_x + \frac{w + 2k\pi}{\sqrt{2(w + 2k\pi)^2 - (4k\pi)^2}}\widetilde{e}_w. \end{aligned}$$

10.6. **Distance underestimator for a ball.** As we explained in Section 10.1, $X \cong \widetilde{\mathrm{SL}}(2,\mathbb{R})$ is a (metrically) twisted line bundle over $\mathbb{H}^2$. As a subset of $\mathbb{R}^4$, our model for $\widetilde{\mathrm{SL}}(2,\mathbb{R})$ is identical to our model for $Y = \mathbb{H}^2 \times \mathbb{E}$ (see Section 8). This gives an identification (of course, not an isometry) between X and Y, which we use to approximate distances in X as follows.

Lemma 10.4. *For every point $p \in X$, we have*

$$\mathrm{dist}_X(o,p) \leqslant \mathrm{dist}_Y(o,p) \leqslant 2\mathrm{dist}_X(o,p).$$

Proof. Consider an arc length parametrized geodesic $\gamma\colon [0,\ell] \to X$ joining o to p. We write $L_Y(\gamma)$ for its length in Y. A computation shows that $L_Y(\gamma) \leqslant 2\ell$. Hence $\mathrm{dist}_Y(o,p) \leqslant L_Y(\gamma) \leqslant 2\mathrm{dist}_X(o,p)$. Second, a similar calculation shows that the arc length parametrized geodesic γ' of Y joining o to p is still parametrized by arc length when viewed as a path in X. Consequently $\mathrm{dist}_X(o,p) \leqslant L_X(\gamma') \leqslant \mathrm{dist}_Y(o,p)$. □

Remark 10.5. Note that the proof here relies on the fact that these geodesics begin at the origin, o. The result does not hold for general geodesics. ◊

As in Nil, we use this observation to construct a distance underestimator $\sigma'\colon X \to \mathbb{R}$ to render a ball of radius r centered at o, as follows. Let

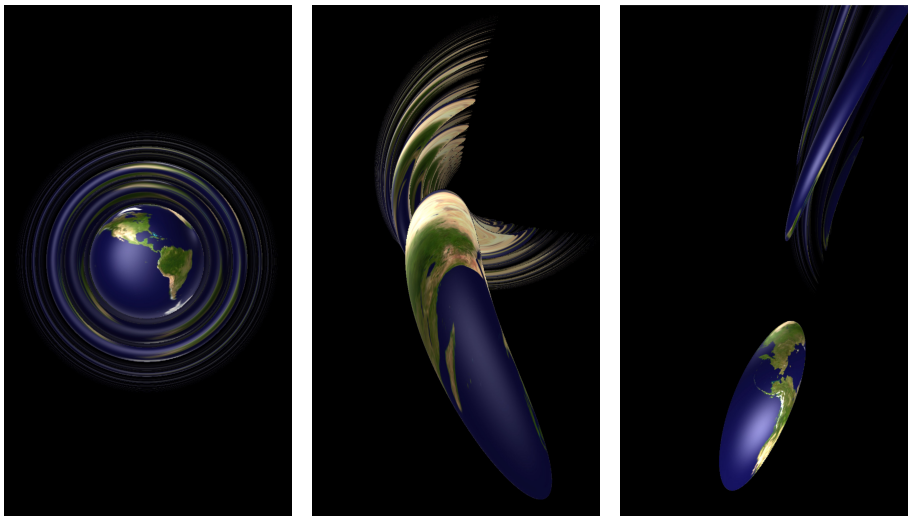
$$\sigma'(p) = \begin{cases} \sigma(p) - r, \text{ if } \sigma(p) > r + \eta \\ 2\sigma(p) - r, \text{ if } \sigma(p) < 2(r - \eta) \\ \mathrm{dist}(o,p) - r, \text{ otherwise,} \end{cases}$$

where

$$\sigma(p) = \frac{1}{2}\sqrt{\mathrm{arccosh}^2(z^2 - x^2 - y^2) + w^2}$$

is half the distance from the origin to p in Y, and $\eta > 0$ is a constant that is much larger than the threshold ϵ used to stop the ray-marching algorithm. In the last case of σ', the exact distance is computed numerically as explained in Section 10.5. We use this distance underestimator to render the balls in Figure 10.4. Compare with Figure 5.13, which shows a line of balls in Nil.

10.7. **Creeping to horizontal half-spaces.** As for Nil in Section 9.7, we can use a version of creeping to draw pictures of "horizontal" half-spaces. For example, in Figure 10.5 we draw the half-space $w \leqslant 0$, the boundary $\mathbb{H}^2$ patterned with equidistant curves to a geodesic (in white).



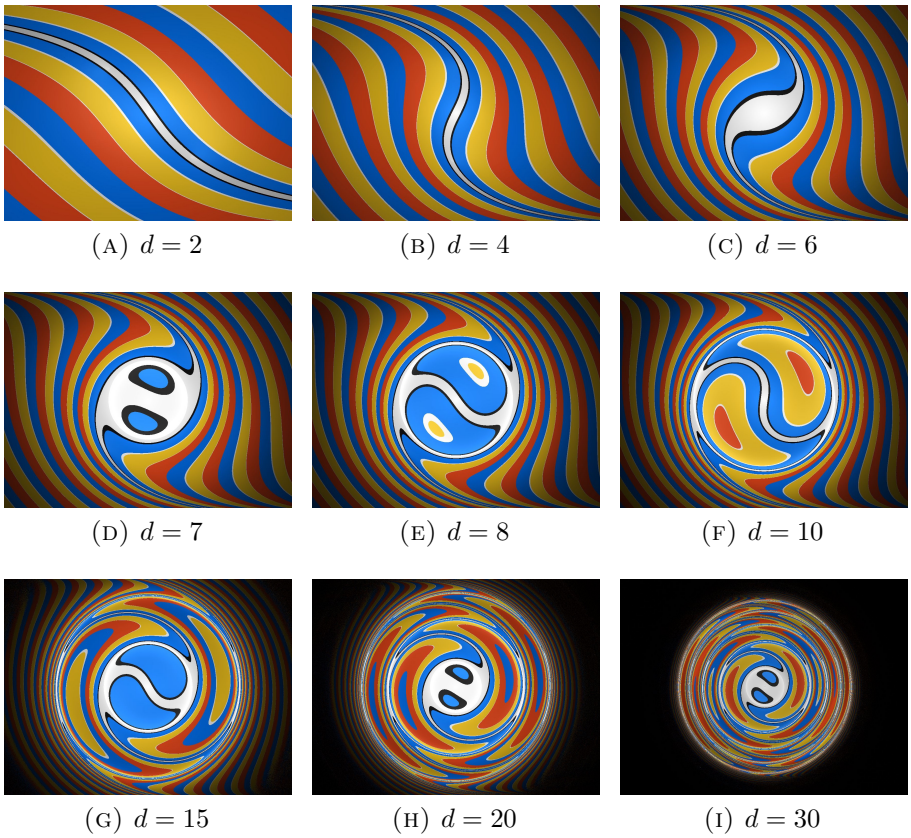
(A) Looking along the fiber. (B) Looking near the fiber direction. (C) A view further from the fiber direction.

FIGURE 10.4. A line of unit balls spaced every 2π along the fiber direction in $\widetilde{\mathrm{SL}}(2,\mathbb{R})$. (Equivalently, a single ball of radius one in $\mathrm{SO}(2,1)$.)

10.8. **Lighting.** We addressed the computation of lighting pairs in Section 10.5. Here, we calculate the intensity $I(r,u)$ experienced from an isotropic light source at distance r and in the direction u. By Equation (5.7), this is inversely proportional to the area density $\mathcal{A}(r,u)$. We calculate this directly by taking the derivative of the geodesic flow as in Equation (5.9).

As a first simplification, note that as the covering map $\lambda\colon X \to \mathcal{Q}$ is a local isometry and $\mathcal{A}(r,u)$ is a local quantity, we may treat $d\lambda_{\tilde{o}}\colon T_{\tilde{o}}X \to T_o\mathcal{Q}$ as an identification and work directly in $\mathcal{Q} = \mathrm{SL}(2,\mathbb{R})$. Let u be the unit vector $u = [a\cos\alpha, a\sin\alpha, c] \in T_o\mathcal{Q}$ expressed in the basis (e_x, e_y, e_w). Recall that Equation (10.2) gives a parameterization of the unit speed geodesic $\gamma(t)$ in direction u as the product of two one-parameter subgroups of $\mathcal{Q} = \mathrm{SL}(2,\mathbb{R})$ followed by a rotation of angle α about the fiber direction. These one-parameter subgroups, and hence the geodesic flow, come in three regimes determined by whether $|c/a|$ is greater than, equal to, or less than one. Below we concern ourselves with the two generic cases.

Let $[\rho,\alpha,w]$ be the cylindrical coordinates on $T_o\mathcal{Q}$ with (ρ,w) the norm of the projections onto the xy-plane and w-axis respectively, and $\alpha \in [0,2\pi)$ measured from the positive x-axis. In these coordinates,



(A) $d = 2$ (B) $d = 4$ (C) $d = 6$

(D) $d = 7$ (E) $d = 8$ (F) $d = 10$

(G) $d = 15$ (H) $d = 20$ (I) $d = 30$

FIGURE 10.5. Horizontal half-space with boundary the hyperbolic plane. The plane is colored with a geodesic (white) and equidistant curves (primary colors). The observer is at $p = [0, 0, 1, d]$. The figures are rendered with a single light source at height 3 above $\mathbb{H}^2$, and distance fog.

the point $ru \in T_o\mathcal{Q}$ is expressed as $[\rho, \alpha, w] = [ra, \alpha, rc]$. Using Equation (5.13), we may calculate the area density in terms of the ρ, α, and w derivatives of the geodesic flow. Here one may deal with the two regimes (in these coordinates, $|\rho| > |w|$ and $|\rho| < |w|$) separately, or unify them into a single computation with complex trigonometric functions. This follows from the particularly nice form of the one-parameter subgroups in Equation (10.2). In either case, even after much simplification, the resulting formula for area density is rather complicated. We describe it below.

Let $K = \sqrt{|\rho^2 - w^2|}$ and let $f_1 \dots f_6$ denote the polynomials in ρ, w, and K:

$$\begin{aligned} f_1 &= 17\rho^6 + 7\rho^4 w^2 + 16\rho^2 w^4 + 32w^6 \\ f_2 &= 48\rho^2 w^2(\rho^2 + w^2) \\ f_3 &= 3\rho^4(5\rho^2 + 3w^2) \\ f_4 &= \rho^6 - 2\rho^2 w^2 - w^4 - \rho^4(w^2 + 1) \\ f_5 &= \rho^6 + 2\rho^2 w^2 + w^4 - \rho^4(w^2 - 1) \\ f_6 &= 2\rho^2(\rho^2 + w^2)K. \end{aligned}$$

To combine the two regimes, we let $(S(x), C(x))$ denote $(\sin(x), \cos(x))$ when $|w| > |\rho|$, and $(\sinh(x), \cosh(x))$ for $|w| < |\rho|$. Finally, let g_1 and g_2 be the functions

$$\begin{aligned} g_1(\rho, w) =& f_1(\rho, w) - f_2(\rho, w)C(K) + f_3(\rho, w)C(2K) \\ g_2(\rho, w) =& f_4(\rho, w) + f_5(\rho, w)C(K) \pm f_6(\rho, w)S(K), \end{aligned}$$

where the $\pm$ in g_2 is positive for $|w| > |\rho|$ and negative when $|w| < |\rho|$. With this notation, the area density is given by

$$\mathcal{A}(r, u) = \frac{\sqrt{\rho^2 + w^2}}{2K^6} \left| S\left(\frac{K}{2}\right) \right| \sqrt{\left| g_1(\rho, w) g_2(\rho, w) \right|}. \tag{10.6}$$

See Figure 10.6. As with the computation of the geodesic flow in Section 10.2, one should use the asymptotic expansion of Equation (10.6) to obtain correct lighting along the null cone $|w| = |\rho|$.
Figure 10.6 shows the intensity variation, as seen in the tangent space to a point.

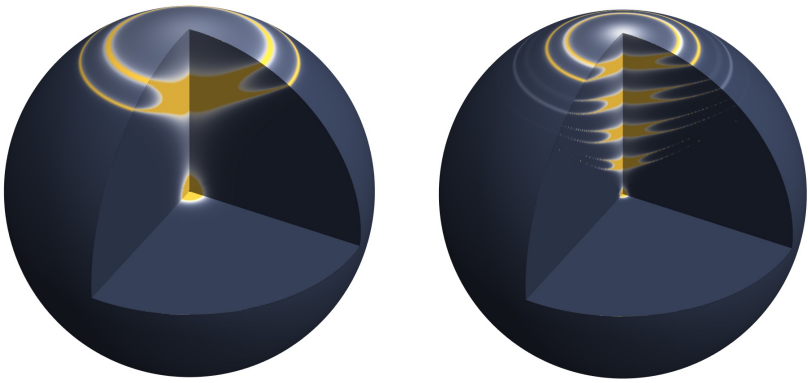
10.9. **Discrete subgroups and fundamental domains.** The manifolds with $\widetilde{\mathrm{SL}}(2, \mathbb{R})$ geometry are classified in [Sco83, Theorem 4.15]. The main examples are unit tangent bundles of hyperbolic surfaces and two-dimensional orbifolds.

Our model $\mathcal{Q}$ of $\mathrm{SL}(2, \mathbb{R})$ is a projective model, in the sense that it induces a faithful representation $\mathrm{Isom}(\mathcal{Q}) \to \mathrm{PGL}(4, \mathbb{R})$. This is not the case for X however. Nevertheless, we can adapt the strategy described in Section 4.1.2 to produce an efficient fundamental domain. We explain this strategy with an example.

Let Γ be the fundamental group of a genus two surface Σ:

$$\Gamma = \langle A_1, A_2, B_1, B_2 \mid [A_1, B_1][A_2, B_2] = 1 \rangle.$$

A choice of hyperbolic metric on Σ induces a representation $\Gamma \to \mathrm{SL}(2, \mathbb{R})$. For our example, we choose this metric so that a fundamental



(A) Within a ball of radius 10. (B) Within a ball of radius 30.

FIGURE 10.6. The lighting intensity function $I(r, u)$ in $\widetilde{\mathrm{SL}}(2, \mathbb{R})$ geometry

domain U for the action of Γ on $\mathbb{H}^2$ is a regular octagon centered at the origin, see Figure 10.7. The generators of Γ can now be written as points of $\mathcal{Q}$:

$$A_1 = \left[\frac{\sqrt{2}}{2} + 1, -\frac{\sqrt{2}}{2} - 1, -\sqrt{2}\sqrt{\sqrt{2}+1}, 0\right],$$
$$A_2 = \left[\frac{\sqrt{2}}{2} + 1, -\frac{\sqrt{2}}{2} - 1, \sqrt{2}\sqrt{\sqrt{2}+1}, 0\right],$$
$$B_1 = \left[\frac{\sqrt{2}}{2} + 1, \frac{\sqrt{2}}{2} + 1, \sqrt{\sqrt{2}+1}, -\sqrt{\sqrt{2}+1}\right],$$
$$B_2 = \left[\frac{\sqrt{2}}{2} + 1, \frac{\sqrt{2}}{2} + 1, -\sqrt{\sqrt{2}+1}, \sqrt{\sqrt{2}+1}\right].$$

The pre-image $\widetilde{\Gamma}$ of Γ by the covering map $\lambda\colon X \to \mathcal{Q}$ is now a lattice in $\widetilde{\mathrm{SL}}(2, \mathbb{R})$, viewed as a subset of G, the isometries of $X = \widetilde{\mathrm{SL}}(2, \mathbb{R})$. We choose lifts $\widetilde{A}_1$, $\widetilde{A}_2$, $\widetilde{B}_1$, and $\widetilde{B}_2$ of the previous generators so that their fiber components are respectively $-\pi/2$, $-\pi/2$, $\pi/2$, and $\pi/2$. For convenience, we define a new element $\widetilde{C}$ that is the translation by 2π along the fiber direction. One checks that $\widetilde{C}^{-2} = [\widetilde{A}_1, \widetilde{B}_1][\widetilde{A}_2, \widetilde{B}_2]$ in $\widetilde{\Gamma}$. Note also that $\widetilde{C}$ commutes with $\widetilde{A}_1$, $\widetilde{A}_2$, $\widetilde{B}_1$, and $\widetilde{B}_2$. A fundamental domain for the action of Γ on X is the subset $D = U \times [-\pi, \pi]$ of $X = \mathcal{H} \times \mathbb{R}$. However, our model X is not well suited to checking easily whether a point belongs to D or not.

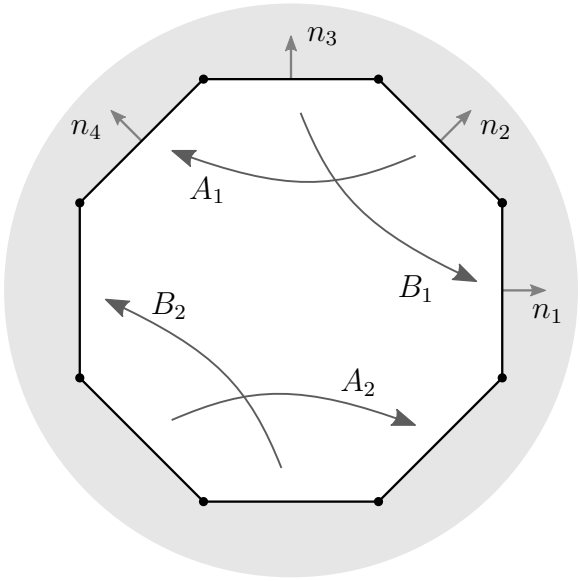


FIGURE 10.7. A sketch of the fundamental domain in $\mathbb{H}^2$. The gray disc is the Klein model of the hyperbolic plane. The white octagon is the fundamental domain for the action on $\mathbb{H}^2$ of the fundamental group Γ of a genus-two surface.

To solve this problem, we consider the isometry $h\colon \mathcal{H} \to \mathcal{K}$ between the hyperboloid model $\mathcal{H} \subset \mathbb{R}^3$ and the Klein model $\mathcal{K} \subset \mathbb{R}^2$ of $\mathbb{H}^2$. The isometry h extends to a bijection

$$\begin{array}{ccc} \mathcal{H} \times \mathbb{R} & \to & \mathcal{K} \times \mathbb{R} \\ (q, w) & \mapsto & (h(q), w). \end{array}$$

This provides yet another model $X' = \mathcal{K} \times \mathbb{R}$ for $\widetilde{\mathrm{SL}}(2, \mathbb{R})$. The image of D under this identification is $D' = U' \times [-\pi, \pi]$ where U' is now an octagon in $\mathcal{K}$ whose sides are straight lines. We define the following normal vectors in $\mathbb{R}^3$

$$\begin{aligned} n_1 &= [1, 0, 0], & n_3 &= [0, 1, 0], \\ n_2 &= \left[\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}, 0\right], & n_4 &= \left[-\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}, 0\right], \\ n_5 &= [0, 0, 1], \end{aligned}$$

see Figure 10.7. To each index $k \in \{1, 2, 3, 4\}$ we associate two half-spaces

$$H_k^- = \{v \in \mathbb{R}^3 \colon \langle v, n_k \rangle \geqslant -\delta\}, \quad \text{and} \quad H_k^+ = \{v \in \mathbb{R}^3 \colon \langle v, n_k \rangle \leqslant \delta\},$$

where $\langle \cdot, \cdot \rangle$ is the standard dot product in $\mathbb{R}^3$ and $\delta = \sqrt{2}\sqrt{\sqrt{2}-1}$. We choose δ so that D' is the intersection of these half-spaces. Similarly, we let

$$H_5^- = \{v \in \mathbb{R}^3 \colon \langle v, n_5 \rangle \geqslant -\pi\}, \quad \text{and} \quad H_5^+ = \{v \in \mathbb{R}^3 \colon \langle v, n_5 \rangle \leqslant \pi\}.$$

The teleporting algorithm has two main steps. Let $p = (q, w)$ be a point in our new model $\mathcal{K} \times \mathbb{R}$ of $\widetilde{\mathrm{SL}}(2, \mathbb{R})$.

(1) If q does not belong to H_1^+ (respectively H_2^+, H_3^+, H_4^+, H_1^-, H_2^-, H_3^-, H_4^-), then we move p by $\widetilde{B}_1^{-1}$ (respectively $\widetilde{A}_1$, $\widetilde{B}_1$, $\widetilde{A}_1^{-1}$, $\widetilde{B}_2^{-1}$, $\widetilde{A}_2$, $\widetilde{B}_2$, $\widetilde{A}_2^{-1}$). Observe that U' is also a Dirichlet domain for the action of Γ on $\mathbb{H}^2$. More precisely, $H_1^+ \cap H_3^+$ is the set of points in $\mathcal{K}$ which are closer to the origin o than their translates by $B_1^{\pm 1}$. Hence the translation by $\widetilde{B}_1^{-1}$ moves the projection q of p to $\mathcal{K}$ closer to o. It follows that after finitely many steps, we can ensure that q belongs to U'. Since we always reduce the distance from o to p, the order in which we perform the algorithm does not matter.

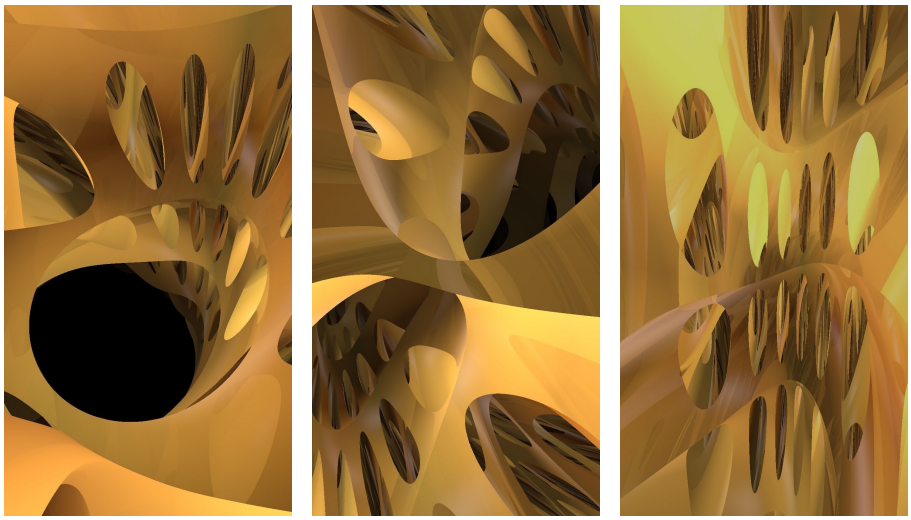
(2) Once this is done, if p does not belong to H_5^- (respectively H_5^+), then we move it by $\widetilde{C}$ (respectively $\widetilde{C}^{-1}$). Note that $\widetilde{C}$ does not affect the horizontal component q of p. Therefore, after this process p lies in the fundamental domain D'.

Figure 10.8 shows some views within the unit tangent bundle to a genus two surface, as described in this section. The fundamental domain is a very tall octagonal prism. To better illustrate the geometry, our scene is the complement of three spheres stacked vertically within this domain.

In Figure 10.9, we show the in-space view for various scenes in $\widetilde{\mathrm{SL}}(2, \mathbb{R})$ geometry. Figure 10.9a shows the same scene as Figure 10.8, with a globe added at the center of each of the three spheres. Figure 10.9b shows a lattice of globes in the unit tangent bundle for a sphere with cone points $\pi/3, \pi/3$, and $2\pi/3$. Figure 10.9c shows solid cylinders (which we implement as vertical objects) around fibers of $\widetilde{\mathrm{SL}}(2, \mathbb{R})$. The lighting in these images is based on a continuously varying direction field rather than point light sources.

11. Sol

11.1. **Model.** As with Nil and $\widetilde{\mathrm{SL}}(2, \mathbb{R})$, Sol is a Lie group. The underlying space of our model is the affine subspace X of $\mathbb{R}^4$ defined by $w = 1$. The group law is as follows: the point $[x, y, z, 1]$ acts on X on



(A) Looking along the fiber.

(B) Looking near the fiber direction.

(C) A view further from the fiber direction.

FIGURE 10.8. The unit tangent bundle of a genus two surface.

the left as the matrix

$$\begin{bmatrix} e^z & 0 & 0 & x \\ 0 & e^{-z} & 0 & y \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

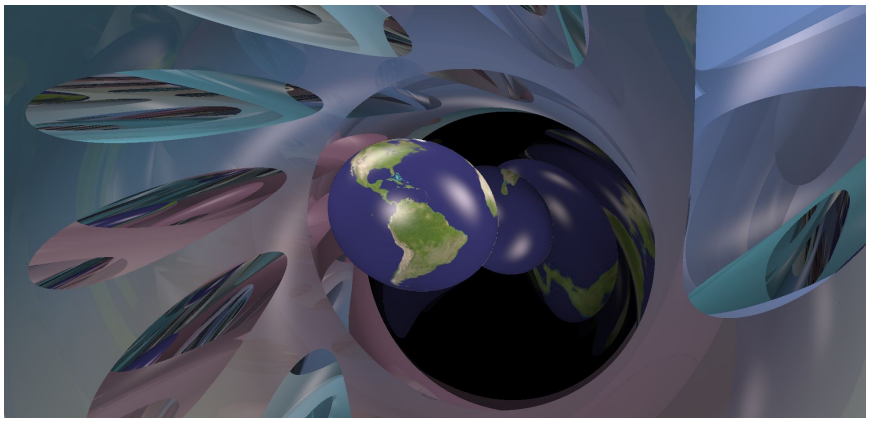
The origin o is the point $[0,0,0,1]$. Its tangent space T_oX is identified with the linear subspace of $\mathbb{R}^4$ given by the equation $w = 0$. The metric tensor at an arbitrary point $p = [x, y, z, 1]$ is

$$ds^2 = e^{-2z}dx^2 + e^{2z}dy^2 + dz^2. \tag{11.1}$$

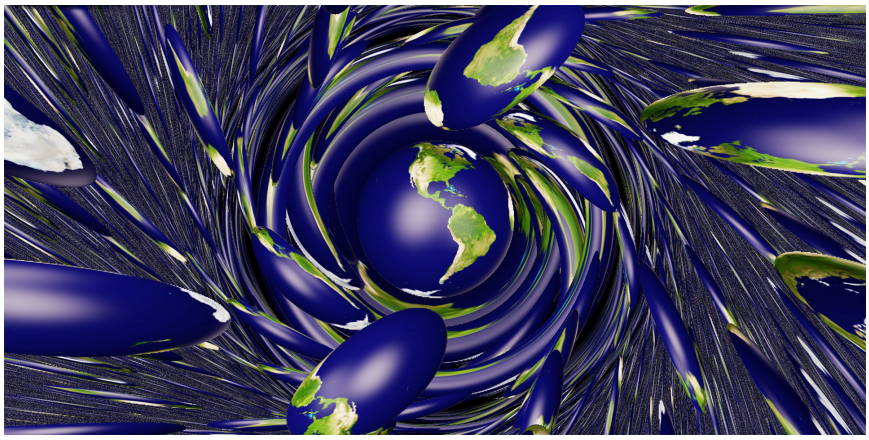
With this metric, the action of Sol on itself is an action by isometries. The stabilizer K of the origin o is isomorphic to the dihedral group of order eight, D_8, which is generated by two symmetries acting on X as the matrices

$$S_1 = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{and} \quad S_2 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

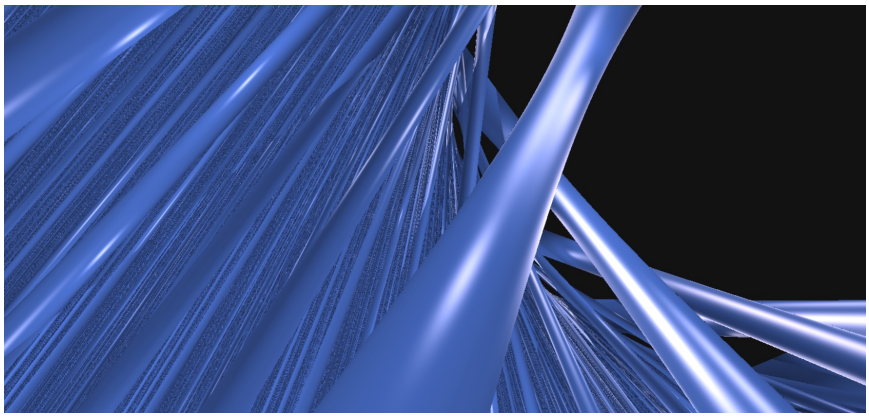
respectively. These symmetries can be observed in the balls of Sol, see Figure 11.1.



(A) A globe within the unit tangent bundle of a genus two surface.



(B) A lattice of globes.



(C) Lifts of the fibers in $UT\mathbb{H}^2$.

FIGURE 10.9. $\widetilde{\mathrm{SL}}(2,\mathbb{R})$ Geometry.

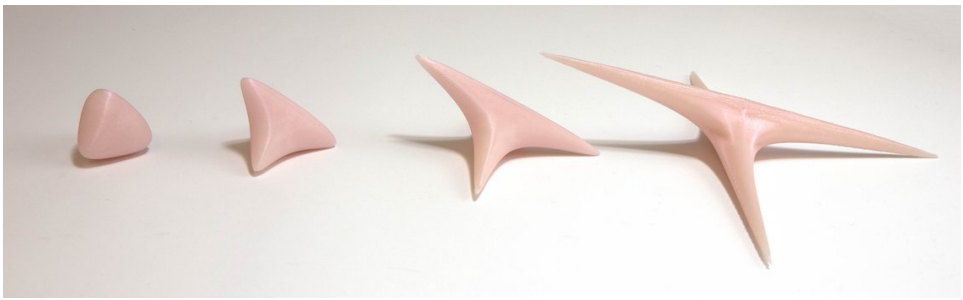


FIGURE 11.1. 3D-printed models of the balls of radius one to four in Sol. We scaled the ball of radius r down by a factor of r in order to keep the physical sizes reasonable. Photograph by Edmund Harriss.

11.2. **Geodesic flow and parallel transport.** As for Nil and $\widetilde{\mathrm{SL}}(2,\mathbb{R})$ in Sections 9.3 and 10.2, we can use Grayson's method to study the geodesic flow and parallel transport. Let $\gamma\colon \mathbb{R} \to X$ be a geodesic, and let be $T(t)\colon T_{\gamma(0)}X \to T_{\gamma(t)}X$ be the corresponding parallel-transport operator. Following Sections 3.2.1 and 3.4.1, we define two paths $u\colon \mathbb{R} \to T_oX$ and $Q\colon \mathbb{R} \to \mathrm{SO}(3)$ by the relations

$$\begin{aligned} \dot\gamma(t) &= d_oL_{\gamma(t)}u(t), \text{ and} \\ T(t)\circ d_oL_{\gamma(0)} &= d_oL_{\gamma(t)}Q(t). \end{aligned}$$

After some computation, Equations (3.2) and (3.6) respectively become

$$\begin{cases} \dot u_x = u_xu_z \\ \dot u_y = -u_yu_z \\ \dot u_z = u_y^2 - u_x^2 \end{cases}$$

and

$$\dot Q + BQ = 0, \quad \text{where} \quad B = \begin{bmatrix} 0 & 0 & u_x \\ 0 & 0 & -u_y \\ -u_x & u_y & 0 \end{bmatrix}.$$

The path u, as well as the geodesic γ, can be computed explicitly [Tro98].[3] Assume that γ starts at the origin o, so that the initial condition is $u(0) = \dot\gamma(0) = [a,b,c,0]$. Because of the symmetries of Sol, we can assume without loss of generality that $a \geqslant 0$ and $b \geqslant 0$. We distinguish three cases.

[3] A commonly cited reference for solving the geodesic flow in Sol is [BS07]. However, the authors do not conduct the computation to the final stage – see their Theorem 4.1(1). Moreover, the formulas given in Theorem 4.1(2) have some errors.

Case $a = 0$. Here the solution for u is

$$u(t) = \left[0, \frac{b}{\cosh t + c \sinh t}, \frac{c + \tanh t}{1 + c \tanh t}, 0\right].$$

It follows that

$$\gamma(t) = \left[0, \frac{b \tanh t}{1 + c \tanh t}, \ln(\cosh t + c \sinh t), 1\right].$$

In particular, γ stays in the plane $\{x = 0\}$. This plane is totally geodesic and isometric to $\mathbb{H}^2$.

Case $b = 0$. Here u and γ can be deduced from the previous case, via a conjugation by the symmetry S_2 fixing the origin. That is,

$$u(t) = \left[\frac{a}{\cosh t - c \sinh t}, 0, \frac{c - \tanh t}{1 - c \tanh t}, 0\right]$$

and

$$\gamma(t) = \left[\frac{a \tanh t}{1 - c \tanh t}, 0, -\ln(\cosh t - c \sinh t), 1\right].$$

Note that γ stays in the plane $\{y = 0\}$, which is also a totally geodesic, isometrically embedded copy of $\mathbb{H}^2$.

Case $ab \neq 0$. We first define some auxiliary parameters. Let

$$k = \sqrt{\frac{1 - 2ab}{1 + 2ab}} \quad \text{and} \quad k' = 2\sqrt{\frac{ab}{1 + 2ab}}.$$

The associated *complete elliptic integrals* of the first and second kind are respectively

$$K(k) = \int_0^{\frac{\pi}{2}} \frac{d\theta}{\sqrt{1 - k^2 \sin^2 \theta}} \quad \text{and} \quad E(k) = \int_0^{\frac{\pi}{2}} \sqrt{1 - k^2 \sin^2 \theta} d\theta.$$

We denote by sn and cn the *Jacobi elliptic sine* and *cosine* functions with elliptic modulus k. We write dn for the *delta amplitude* and ζ for the *Jacobi zeta function*, also with elliptic modulus k. For an in-depth study of elliptic functions, we refer the reader to [Jac29, OM49, Law89]. Recall that sn and cn are $4K(k)$-periodic. Let

$$\mu = \sqrt{1 + 2ab}.$$

We also fix $\alpha \in [0, 4K(k))$ such that

$$\operatorname{sn} \alpha = -\frac{c}{\sqrt{1 - 2ab}} \quad \text{and} \quad \operatorname{cn} \alpha = \frac{a - b}{\sqrt{1 - 2ab}}.$$

Setting $s = \mu t + \alpha$, we now have

$$u(t) = \left[\sqrt{ab}\, \frac{k \operatorname{cn} s + \operatorname{dn} s}{k'}, \sqrt{ab}\, \frac{k'}{k \operatorname{cn} s + \operatorname{dn} s}, -k\mu \operatorname{sn} s, 0\right].$$

In order to write the solution for γ, we let

$$L = \frac{E(k)}{k'K(k)} - \frac{k'}{2}.$$

We finally get

$$\gamma(t) = \begin{bmatrix} \sqrt{\frac{b}{a}}\left(\frac{1}{k'}\big(\zeta(s) - \zeta(\alpha)\big) + \frac{k}{k'}\big(\operatorname{sn} s - \operatorname{sn}\alpha\big) + \big(s-\alpha\big)L\right) \\ \sqrt{\frac{a}{b}}\left(\frac{1}{k'}\big(\zeta(s) - \zeta(\alpha)\big) - \frac{k}{k'}\big(\operatorname{sn} s - \operatorname{sn}\alpha\big) + \big(s-\alpha\big)L\right) \\ \frac{1}{2}\ln\left(\frac{b}{a}\right) + \operatorname{arcsinh}\left(\frac{k}{k'}\operatorname{cn} s\right) \\ 1 \end{bmatrix}.$$

In practice, we use a mixed approach, as follows.

- When we need to flow for a long time (for example when all objects in the scene are very far away from the camera), then we use the explicit formula above. However, if the initial direction $\dot\gamma(0)$ is close to one of the hyperbolic planes, this formula suffers from many numerical errors. This is an example of the kind of error described in Section 2.4.1(2). In this case, we replace the exact solution by its asymptotic expansion of order two.
- When we need to flow for a short time, the above method again seems to suffer from significant numerical errors. This happens when during the ray-marching algorithm some object is very close, or when updating the position and facing of the observer between two frames. In this situation, we numerically integrate the geodesic flow and the parallel transport equations using the Runge–Kutta method of order two.

Remark. Since Jacobi elliptic and zeta functions are not available in the OpenGL library, we implemented them directly, using the AGM algorithm [Bul65, OLBC10, Abr66].

11.3. **Distance to coordinate half-spaces.** Given $\alpha \in \mathbb{R}$, we write $H_z^+(\alpha) = \{z \geqslant \alpha\}$ and $H_z^-(\alpha) = \{z \leqslant \alpha\}$. Note that the boundary $\{z = \alpha\}$ of these half-spaces is isometric to a euclidean plane, but is not convex as a subspace of X. Recall that we write $\operatorname{sdf}(\cdot, S)$ for the signed distance function for the scene S.

Lemma 11.2. *Fix a real number α. For every point $p = [x, y, z, 1]$ in X, we have*

$$\operatorname{sdf}\big(p, H_z^-(\alpha)\big) = z - \alpha \quad \textit{and} \quad \operatorname{sdf}\big(p, H_z^+(\alpha)\big) = \alpha - z.$$

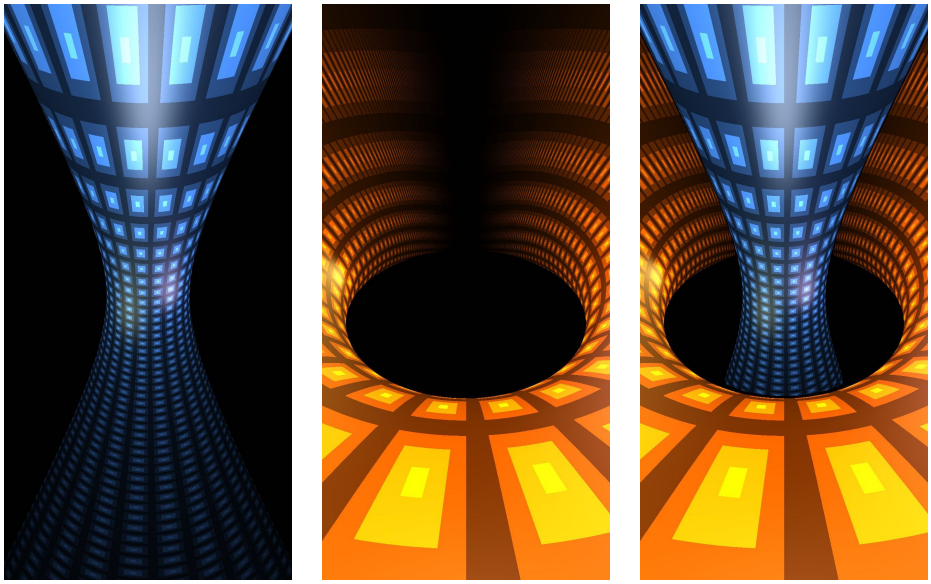
Proof. Observe that the collections $\{H_z^+(\alpha) \mid \alpha \in \mathbb{R}\}$ and $\{H_z^-(\alpha) \mid \alpha \in \mathbb{R}\}$ are both invariant under the action of Sol on itself. Thus without loss of generality, we can assume that p is the origin o. Similarly, the symmetry S_2 fixes the origin and permutes $H_z^+(\alpha)$ and $H_z^-(\alpha)$. Hence it suffices to prove the statement for $H_z^+(\alpha)$. Suppose that $\alpha \geqslant 0$ (the other case works in the same way). The path $\gamma(t) = [0, 0, t, 1]$ is a geodesic starting at the origin and hitting $H_z^+(\alpha)$ at time $t = \alpha$. Hence we have $\operatorname{dist}(o, H_z^+(\alpha)) \leqslant \alpha$.

Let us prove the other inequality. Consider a point $q \in H_z^+(\alpha)$ and a minimizing arc length parametrized geodesic $\gamma\colon [0, \ell] \to X$ from o to a point q. If we write the path γ as $\gamma(t) = [x(t), y(t), z(t), 1]$, then from the metric given in Equation (11.1) we get that $|\dot{z}(t)| \leqslant 1$, because γ is arc length parametrized. Consequently, we have

$$\operatorname{dist}(o, q) \geqslant \ell \geqslant z(\ell) \geqslant \alpha.$$

This inequality holds for every point $q \in H_z^+(\alpha)$, hence the result. □

In Figure 11.2, we use these signed distance functions to draw horizontal half-spaces, patterned with square tilings.



(A) $H_z^+(1)$ (B) $H_z^-(-1)$ (C) $H_z^+(1) \cup H_z^-(-1)$

FIGURE 11.2. Wide-angle views of horizontal half-spaces in Sol geometry. The boundaries of these half-spaces are tiled by squares of side length 1/5.

We can similarly compute the exact distance function to a half-space bounded by a hyperbolic plane in Sol. Let $H_x^+(\alpha) = \{x \geqslant \alpha\}$ and $H_x^-(\alpha) = \{x \leqslant \alpha\}$.

Lemma 11.3. *Fix a real number α. For every point $p = [x, y, z, 1]$ in X, we have*

$$\operatorname{sdf}\big(p, H_x^+(\alpha)\big) = -\operatorname{sdf}\big(p, H_x^-(\alpha)\big) = \operatorname{arcsinh}\big((\alpha - x)e^{-z}\big).$$

Proof. Assume first that $p = o$ is the origin. We write the proof for $H_x^+(\alpha)$ with $\alpha > 0$. The other cases work in the same way. We claim that the distance from o to $H_x^+(\alpha)$ is also the distance in the hyperbolic plane $U = \{y = 0\}$ from o to the half plane $U^+(\alpha) = \{x \geqslant \alpha \text{ and } y = 0\}$. We have

$$\operatorname{dist}_X\big(o, H_x^+(\alpha)\big) \leqslant \operatorname{dist}_U\big(o, U^+(\alpha)\big).$$

In order to prove the converse inequality, it suffices to show that the projection $X \to U$ sending $[x, y, z, 1]$ to $[x, 0, z, 1]$ is 1-Lipschitz. To see this, take two points q and q', and a geodesic $\gamma\colon [0, T] \to X$ joining them. We write $\gamma(t) = [x(t), y(t), z(t), 1]$. From the metric given in Equation (11.1) we get

$$\begin{aligned}\operatorname{dist}(q, q') = L(\gamma) &= \int_0^T \sqrt{e^{-2z}\dot{x}^2 + e^{2z}\dot{y}^2 + \dot{z}^2}\,dt \\ &\geqslant \int_0^T \sqrt{e^{-2z}\dot{x}^2 + \dot{z}^2}\,dt \\ &= L(\pi \circ \gamma)\end{aligned}$$

where $L(\gamma)$ and $L(\pi \circ \gamma)$ stands for the length in X of γ and $\pi \circ \gamma$ respectively. Thus, $\operatorname{dist}(q, q') \geqslant \operatorname{dist}(\pi(q), \pi(q'))$.

We now compute $\operatorname{dist}_U(o, U^+(\alpha))$. Recall that U is isometric to the hyperbolic plane $\mathbb{H}^2$. More precisely $[x, z]$ is a horocycle-based coordinate system of $\mathbb{H}^2$: the distance between $p_1 = [x_1, 0, z_1, 1]$ and $p_2 = [x_2, 0, z_2, 1]$ is characterized by

$$\cosh \operatorname{dist}(p_1, p_2) = \cosh(z_1 - z_2) + \frac{1}{2}e^{-(z_1+z_2)}(x_1 - x_2)^2.$$

One checks that the projection of o onto $U^+(\alpha)$ is the point

$$\left[\alpha, 0, \frac{1}{2}\ln(1 + \alpha^2), 1\right]$$

and

$$\operatorname{dist}_U\big(o, U^+(\alpha)\big) = \operatorname{arcsinh}(\alpha).$$

Assume now that $p = [x, y, z, 1]$ is an arbitrary point. There is a unique element L of Sol sending o to p. Observe that L^{-1} maps $H_x^+(\alpha)$

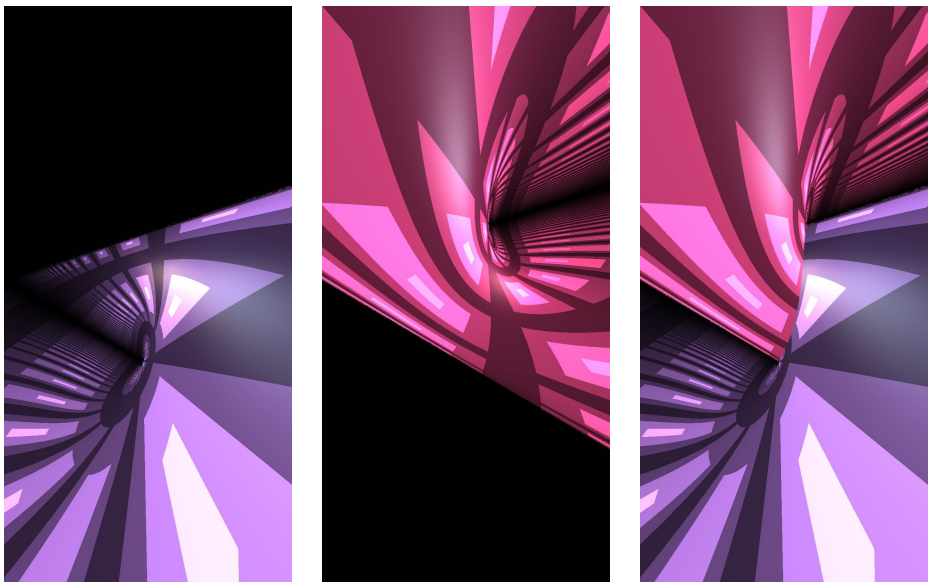
to $H_x^+(\alpha')$ where $\alpha' = (\alpha - x)e^{-z}$. The result then follows from the previous discussion. □

We can define the half-spaces $H_y^\pm(\alpha)$ as we did for $H_x^\pm(\alpha)$. Using the fact that the isometry S_2 fixing the origin sends $[x, y, z, 1]$ to $[y, x, -z, 1]$ we get the following statement.

Lemma 11.4. *Fix a real number α. For every point $p = [x, y, z, 1]$ in X, we have*

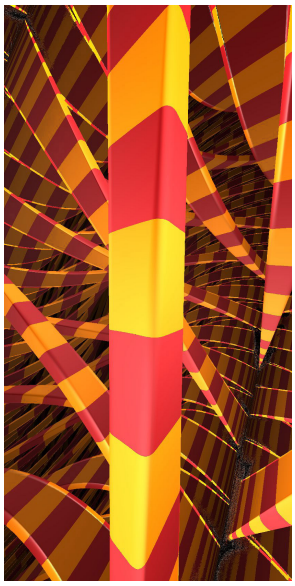
$$\mathrm{sdf}\big(p, H_y^+(\alpha)\big) = -\mathrm{sdf}\big(p, H_y^-(\alpha)\big) = \mathrm{arcsinh}\left((\alpha - y)e^z\right)$$

In Figure 11.3, we use these signed distance functions to draw half-spaces with hyperbolic boundary, patterned with square tilings. Combining these signed distance functions with boolean operations, we can make tubes around vertical geodesics with square cross-sections. See Figure 11.4a.

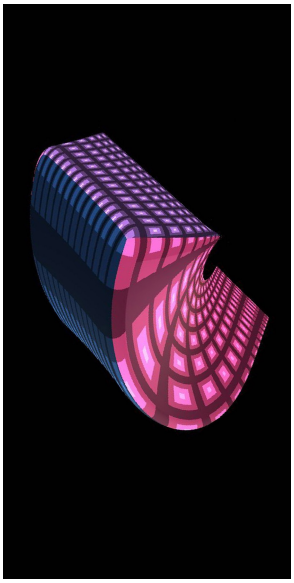


(A) $H_x^+(1)$ (B) $H_y^-(-1)$ (C) $H_x^+(1) \cup H_y^-(-1)$

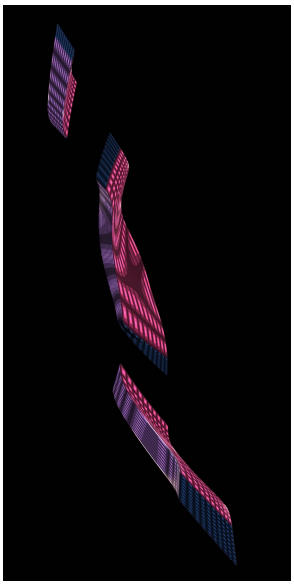
FIGURE 11.3. Wide-angle view of half-spaces with hyperbolic plane boundary in Sol geometry. The boundaries are tiled by quadrilaterals formed from a family geodesics parallel to the z-axis and the families of orthogonal horocycles. The horocycles are evenly spaced, with distance one between neighbors.



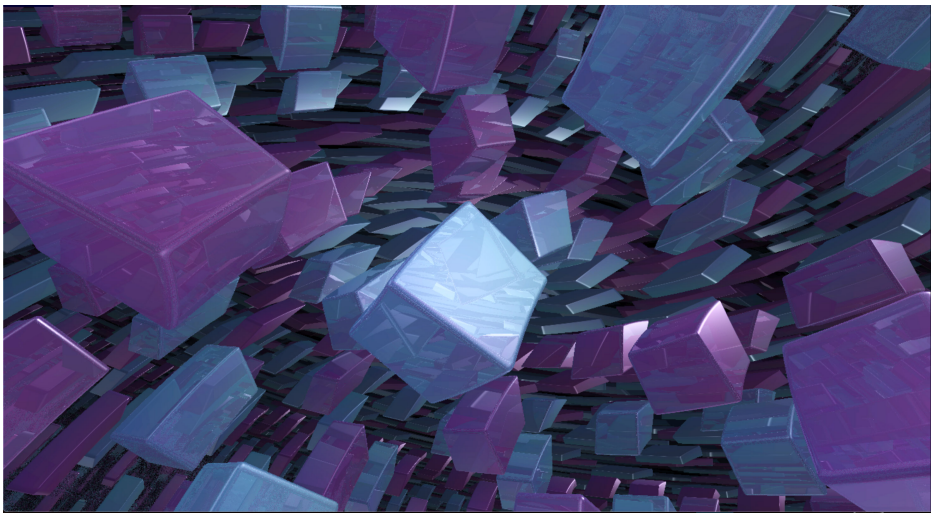
(A) Tubes around vertical geodesics with square cross-sections.



(B) A cube with side-length 3.



(C) The same (single) cube as in Figure 11.4b, viewed from a distance.



(D) A lattice of cubes, dense enough that anomalies seen in Figure 11.4c are mostly hidden from view.

FIGURE 11.4. Scenes made from half-spaces with boolean operations.

11.4. **Distance to horizontal axis-aligned solid cylinders.** Following the same strategy as in Section 11.3, we compute the signed distance function for certain solid cylinders. Let $c_x\colon \mathbb{R} \to X$ be the curve given by $c_x(t) = [t, 0, 0, 1]$. Note that c_x is *not* a geodesic of X, but it is a one-parameter subgroup of Sol.

Lemma 11.5. *For every point $p = [x, y, z, 1]$ in X, we have*

$$\cosh \operatorname{dist}(p, c_x) = \cosh z + \frac{1}{2} e^z y^2.$$

Proof. Since c_x is invariant under translations along the x-axis (which are isometries of X), we can assume that p has the form $p = [0, y, z, 1]$. Following the argument given in the proof of Lemma 11.3, we observe that $\operatorname{dist}(p, c_x) = \operatorname{dist}(p, o)$. Using the distance formula in the hyperbolic plane $\{x = 0\}$, we get the result. □

Let $C_x(r)$ be the solid cylinder of radius r around c_x. That is, $C_x(r)$ is the set of point $q \in X$ such that $\operatorname{dist}(q, c_x) \leqslant r$. It follows from Lemma 11.5 that the signed distance function $\sigma\colon X \to \mathbb{R}$ for $C_x(r)$ is

$$\sigma(p) = \operatorname{arccosh}\left(\cosh z + \frac{1}{2} e^z y^2\right) - r.$$

Similarly, we define the solid cylinder of radius r around the curve c_y given by $c_y(t) = [0, t, 0, 1]$. The signed distance function $\sigma\colon X \to \mathbb{R}$ for $C_y(r)$ is

$$\sigma(p) = \operatorname{arccosh}\left(\cosh z + \frac{1}{2} e^{-z} x^2\right) - r.$$

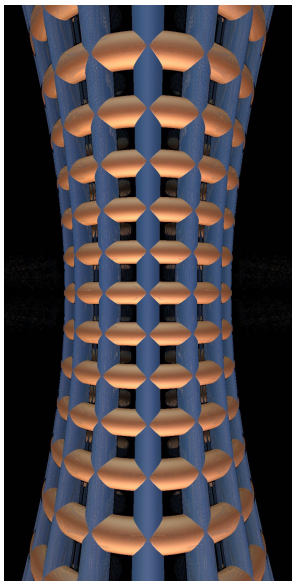
Using the elements of Sol, we can translate the solid cylinders $C_x(r)$ and $C_y(r)$ to get signed distance functions for solid cylinders around any translate of the x- and y-axes. See Figure 11.5a.

11.5. **Approximating balls and more general solid cylinders.** Given a point $p = [x, y, z, 1]$, we approximate its distance to the origin with the function

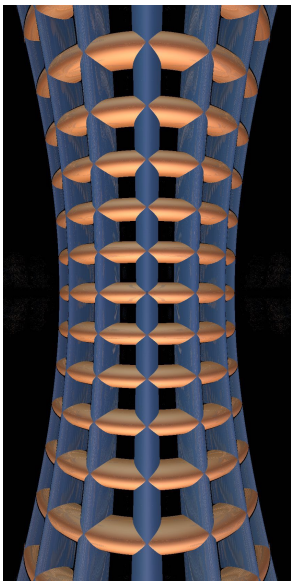
$$\sigma(p) = \sqrt{e^{-2z} x^2 + e^{2z} y^2 + z^2},$$

rescaled by homotheties of the domain and co-domain. This can be used to render decent "pseudo-balls", see Figures 11.6 and 11.7b. It is not currently clear to us whether this function can be used to build a distance underestimator for correct balls.

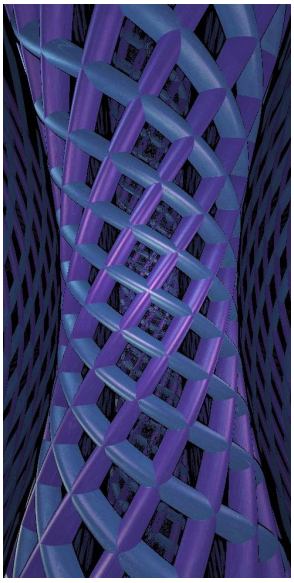
We can similarly produce "solid pseudo-cylinders," approximating the distance from a point to an orbit of a one-parameter subgroup transverse to a plane (the horizontal plane or either hyperbolic plane). The idea is to move a point under the one-parameter subgroup to put it in the



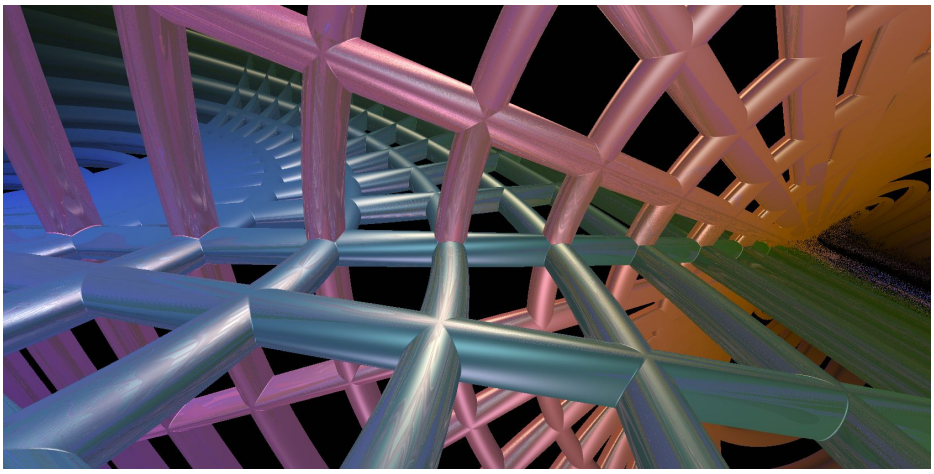
(A) Around translates of the x- and y-axes (exact sdfs).



(B) Around translates of the x- and y-axes (approximations).



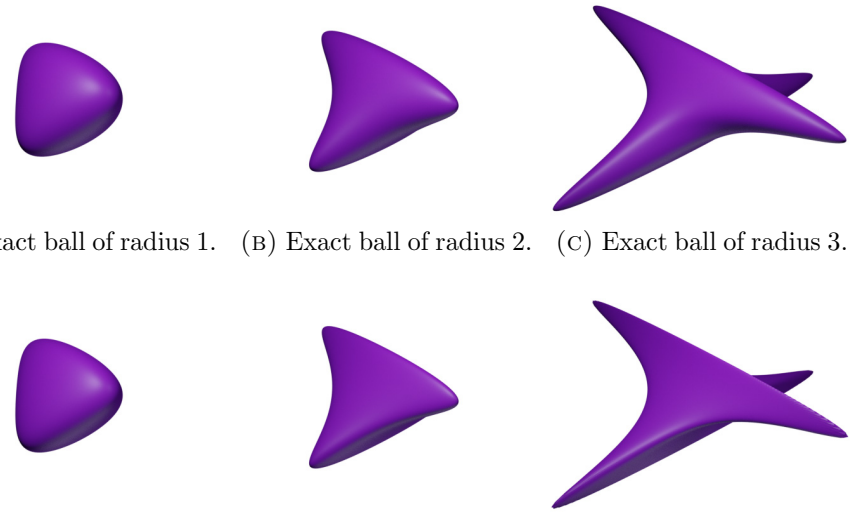
(C) Around geodesics in horizontal planes.



(D) Around horocyclic coordinate lines.

FIGURE 11.5. Solid cylinders.

plane, and then calculate a signed distance function there. If distances are difficult to calculate (either theoretically or practically) even when restricted to the plane, then we can cheat further by measuring, say, euclidean distance in the model space.



(A) Exact ball of radius 1. (B) Exact ball of radius 2. (C) Exact ball of radius 3.

(D) Level set $\sigma = 0.6$. (E) Level set $\sigma = 1.34$. (F) Level set $\sigma = 2.16$.

FIGURE 11.6. Extrinsic comparison of exact and pseudo-balls. The objects have been rescaled so that they all have approximately the same size.

Figure 11.5b shows solid pseudo-cylinders around the translates of the x- and y-axes. This compares well with the exact solid cylinders shown in Figure 11.5a. In Figure 11.5c we draw solid pseudo-cylinders around the geodesics $x = \pm y$ and their translates. Note that these are the only geodesics contained in the xy-plane. In Figure 11.5d we reproduce the two hyperbolic planes of Figure 11.3, represented by grids of solid cylinders. The horocycles in each grid are drawn with exact signed distance functions; for the geodesics we use solid pseudo-cylinders.

11.6. **Direction to a point.** Although it is certainly possible to do so, we did not try to numerically compute the exact direction of geodesics joining two given points in Sol. Recall that this data is only needed to compute lighting pairs for physically correct illumination as in Section 5. As we explained in Section 5.9, we choose instead a more-or-less arbitrary, continuously varying direction field: if s is a point of the scene S and q is the position of the light, then we run all the computations in the Phong model as if the direction from s to q were given by the straight line between s and q in the ambient space $\mathbb{R}^4$ containing our model X.

11.7. **Discrete subgroups and fundamental domains.** The classification of Sol manifolds is given in [Sco83, Theorem 4.17]. Every Sol manifold is a surface bundle over a one-dimensional orbifold. In

particular, Sol can be seen as the universal cover of the suspension M of a regular two-torus T by an Anosov homeomorphism.

The fundamental group Γ of M provides a lattice in X. We explain here with a concrete example how to construct a fundamental domain D for the action of Γ on X.

To avoid any confusion, we denote by $[u_1, u_2]$ the coordinates of a point in the universal cover $\mathbb{R}^2$ of the two-torus T. The fundamental group $\pi_1(T) \cong \mathbb{Z}^2$ acts on $\mathbb{R}^2$ by integer translations. Let f be the Anosov homeomorphism of T acting on $\mathbb{R}^2$ as the matrix

$$\begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix}.$$

Let M be the mapping torus of T with monodromy f. Its fundamental group Γ is given by the presentation

$$\Gamma = \langle A_1, A_2, B \mid [A_1, A_2] = 1, BA_1B^{-1} = A_1^2A_2, BA_2B^{-1} = A_1A_2 \rangle.$$

Here A_1 and A_2 are the standard generators of $\mathbb{Z}^2$, while the conjugation by B is the automorphism of $\mathbb{Z}^2$ induced by f. As in Nil, Γ is generated by A_1 and B. Nevertheless, is it more convenient to keep three generators, as they correspond to translations in three independent directions.

We identify the universal cover $\widetilde{M}$ of M with $\mathbb{R}^3$, equipped with coordinates $[u_1, u_2, u_3]$. Here the set $\{u_3 = 0\}$ corresponds to a copy of $\widetilde{T}$ inside $\widetilde{M}$. The generators A_1 and A_2 act by translation along u_1 and u_2, while B translates along u_3 and applies f to the orthogonal plane.

The next step is to identify X with $\widetilde{M}$. Let b be the point of Sol whose coordinates in X are $b = [0, 0, \tau, 1]$. (The value of $\tau > 0$ will be determined later.) We require that under our identification, the translation by b in X becomes the action of B on $\widetilde{M}$. Observe that b dilates the x-axis while contracting the y-axis. Thus we need to identify the x-direction (respectively y-direction) of X with the expanding (respectively contracting) direction of f.

The matrix defining f has two eigenvalues, namely ϕ^2 and ϕ^{-2}, where $\phi = (1 + \sqrt{5})/2$ is the golden ratio. The corresponding eigenvectors are

$$v_+ = [\phi, 1], \quad \text{and} \quad v_- = [-1, \phi].$$

We now define a homeomorphism $h\colon X \to \widetilde{M}$ as the restriction to X of the linear map $\mathbb{R}^4 \to \mathbb{R}^3$ given by the matrix

$$\begin{bmatrix} \phi & -1 & 0 & 0 \\ 1 & \phi & 0 & 0 \\ 0 & 0 & \tau^{-1} & 0 \end{bmatrix},$$

where we now set $\tau = 2\ln\phi$. In addition, we write a_1 and a_2 for the elements of Sol whose coordinates in X are

$$a_1 = \left[\frac{\phi}{\phi+2}, -\frac{1}{\phi+2}, 0, 1\right] \quad \text{and} \quad a_2 = \left[\frac{1}{\phi+2}, \frac{\phi}{\phi+2}, 0, 1\right].$$

It follows from our construction that the map h conjugates the translation by a_1 (respectively a_2, b) in X to the action of A_1 (respectively A_2, B) on $\widetilde{M}$. A fundamental domain D for the action of Γ on X is the image under h^{-1} of the cube $[-1/2, 1/2]^3 \subset \widetilde{M}$. That is,

$$D = \left\{ \left[\frac{u_1\phi + u_2}{\phi+2}, \frac{-u_1 + u_2\phi}{\phi+2}, u_3\tau, 1\right] \middle| \, u_1, u_2, u_3 \in [-1/2, 1/2] \right\}.$$

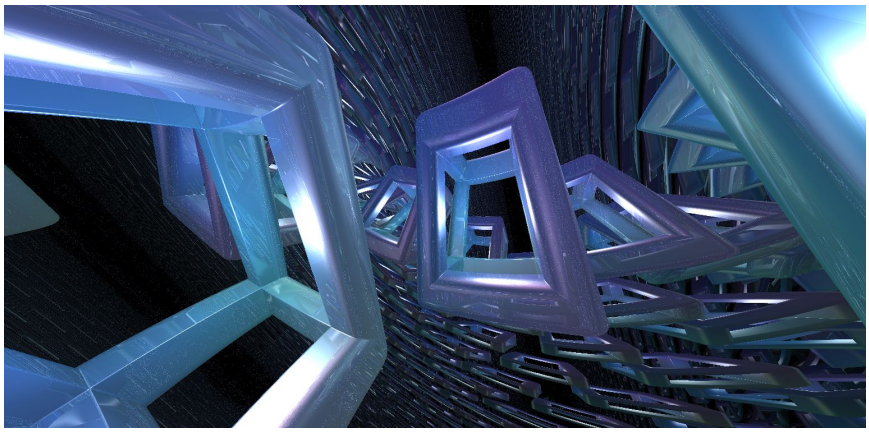
Our model X for Sol is also a projective model. The fundamental domain D can be seen as the intersection of a collection of half-spaces $H_1^{\pm}$, $H_2^{\pm}$, $H_3^{\pm}$ as described in Section 4.1.2. Here

$$H_1^- = \left\{ \left[\frac{u_1\phi + u_2}{\phi+2}, \frac{-u_1 + u_2\phi}{\phi+2}, u_3\tau, 1\right] \middle| \, u_1 \geqslant -1/2, \ u_2, u_3 \in \mathbb{R} \right\},$$
$$H_1^+ = \left\{ \left[\frac{u_1\phi + u_2}{\phi+2}, \frac{-u_1 + u_2\phi}{\phi+2}, u_3\tau, 1\right] \middle| \, u_1 \leqslant 1/2, \ u_2, u_3 \in \mathbb{R} \right\}.$$

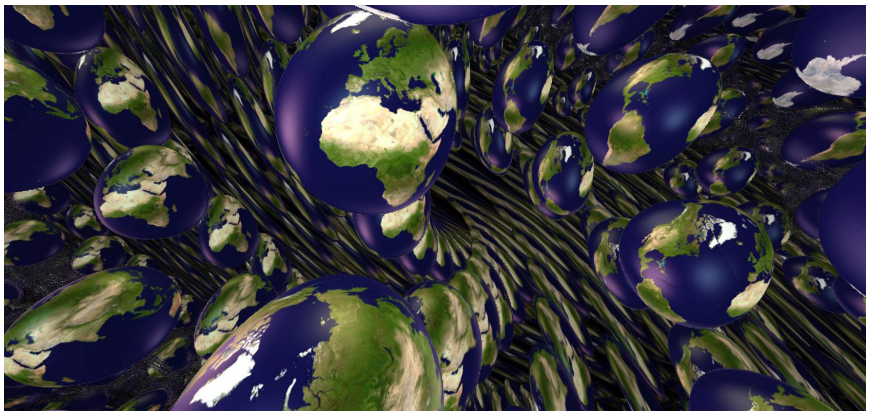
The half-spaces $H_2^{\pm}$, $H_3^{\pm}$ are defined in a similar way. The teleporting algorithm has two main steps. Let $p = [x, y, z, 1]$ be a point in X.

(1) If p does not belong to H_3^- (respectively H_3^+), then we move it by b (respectively b^{-1}). After finitely many steps, the new point p lies in $H_3^- \cap H_3^+$.
(2) Once this is done, if p does not belong to H_1^- (respectively H_1^+, H_2^-, H_2^+), then we translate it by a_1 (respectively a_1^{-1}, a_2, a_2^{-1}). Note that this does not change the z-coordinate of p. Since a_1 and a_2 commute, we don't pay attention to the order in which we perform these operations. After finitely many steps, the new point p belongs to D.

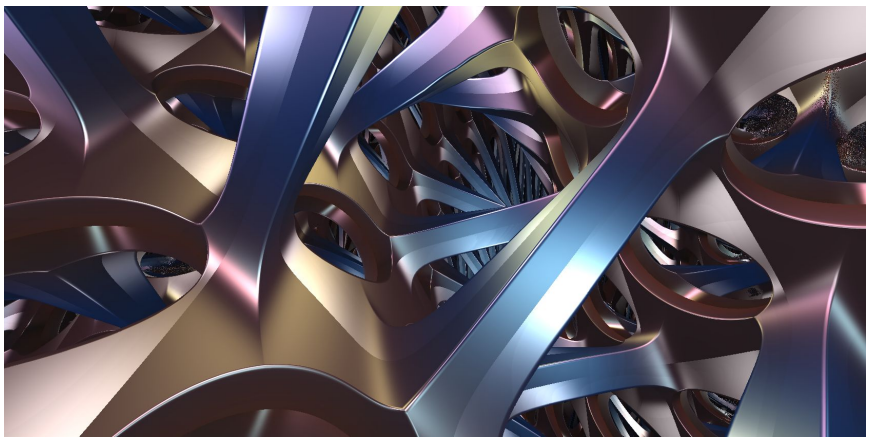
In Figure 11.7a, we draw a lattice of cubes in a neighborhood of the xy-plane. The center of each cube is at a vertex of the tiling of the plane $\widetilde{T}$ corresponding to the action of the subgroup of Γ generated by A_1 and A_2. Figure 11.7b shows the inside view of an Anosov torus bundle, with a ball textured as the Earth for the scene. Figure 11.7c shows the same manifold, with the complement of three solid pseudo-cylinders around the curves $[t\phi, -t, 0, 1]$, $[t, t\phi, 0, 1]$, and $[0, 0, t, 1]$ as the scene.



(A) Cubes on the xy-plane.



(B) An Anosov torus bundle.



(C) An Anosov torus bundle.

FIGURE 11.7. Sol Geometry.

12. Future directions

12.1. **Virtual reality.** As mentioned in Section 3.6, there are serious problems that must be addressed before we can use stereoscopic vision to give the user depth cues in a virtual reality experience.

12.2. **Sol.** Some elements of our work are still incomplete for Sol geometry, namely correct lighting, and correct signed distance functions (or even distance underestimators) for balls. One of the difficulties is that we do not yet have an efficient method to compute the lengths and directions of the geodesics from the origin o to an arbitrary point p.

For Nil and $\widetilde{\mathrm{SL}}(2,\mathbb{R})$, we used the rotation-invariance of our model to build a one-to-one correspondence between those geodesics and the zeros of a function $\phi \to \chi(\phi)$ (depending on p), see Sections 9.5 and 10.5. Since χ is convex on each interval I where it is defined, Newton's method very efficiently computes its zeros. In particular, any value $\phi_0 \in I$ where $\chi(\phi_0) > 0$ can serve as a seed for the algorithm.

The lack of rotation invariance in Sol makes it much harder to implement similar ideas. One could use a multi-variable Newton's method to find the geodesics from o to p. It is however not obvious where to start the procedure. A deeper analysis of the solutions of the geodesic flow is needed here.

12.3. **Directed distance underestimators.** For certain scenes it can be difficult to produce the corresponding signed distance function, or even a distance underestimator. An example is the xy-plane in the Nil geometry (Section 9.7). However, when we are ray-marching along a geodesic γ, we do not in fact need to know the distance from any point $p \in X$ to the scene, but only the distance to the closest point of the scene lying on γ. This leads us to the following definitions.

Definition 12.1. Given a scene $S \subset X$, the associated *directed signed distance function* $\sigma\colon TX \to \mathbb{R}$ is a map characterized as follows. Let $v \in T_pX$ be a tangent vector at p. Let γ be the geodesic starting at p in the direction v.

- If p does not belong to S, then $\sigma(v)$ is the distance from p to the closest point of S on γ.
- If p is in S, then $-\sigma(v)$ is the distance from p to the closest point of $X \setminus S$ on γ. ◊

Such a function is a priori also very hard to obtain. Indeed it means that we can compute the intersection of any geodesic with our scene; this is precisely the data required for ray-tracing. Nevertheless, as in

Section 2.2, we can perform ray-marching using an underestimator that takes as its input a tangent vector to a ray.

Definition 12.2. A *directed distance underestimator* for the scene S is a map $\sigma' \colon TX \to \mathbb{R}$ such that

(1) The signs of $\sigma'(v)$ and $\sigma(v)$ are the same for all points $v \in TX$,
(2) $|\sigma'(v)| \leqslant |\sigma(v)|$ for all $v \in TX$, and
(3) If $\{v_1, v_2, \ldots\}$ is a sequence of points in TX such that $\sigma'(v_n)$ converges to zero, then so does $\sigma(v_n)$. ♢

Ray-marching with such a directed distance underestimator will produce the same pictures as ray-marching with an undirected signed distance function. These, in some sense, bridge the gap between ray-tracing and undirected ray-marching. With directed distance underestimators, we expect to expand the collection of scenes that we can render.

Directed distance underestimators may also help improve efficiency. When using a standard signed distance function (or distance underestimator), the length of the steps becomes very small as a geodesic ray passes very close to the scene without hitting it. If the maximal number of steps for the algorithm is not large enough, this creates background-colored halos around objects. With a directed distance underestimator, we can hope that the length of the steps in this situation will be larger, thus making the algorithm converge faster.

12.4. **Non-maximal homogeneous riemannian geometries.** Recall that the transitive action of a Lie group G on a manifold X determines a homogeneous geometry. To be a Thurston geometry, a homogeneous geometry must satisfy four additional restrictions, see Section 1.1. The first two of these conditions, having X simply connected and G act with compact point stabilizer, define a *riemannian homogeneous space*. (For a complete classification of three-dimensional riemannian homogeneous spaces, see [Pat96].) These two conditions greatly simplify calculations of the geodesic flow, parallel transport, and more. The second two conditions restrict to those needed for geometrization. However, we do not need these conditions anywhere in our ray-marching algorithms. There are many interesting geometries satisfying only the first two conditions that could be visualized in a similar fashion. Celińska-Kopczyńska and Kopczyński have begun to investigate visualizations of one-parameter spaces of metrics of this kind on the three-sphere.

12.5. **Homogeneous pseudo-riemannian & lorentzian geometries.** Generalizing riemannian geometry, a *pseudo-riemannian manifold* is a manifold M together with a choice of (not necessarily positive

definite) nondegenerate bilinear form on each tangent space. When the bilinear form is not positive definite, the existence of null vectors (nonzero $v \in T_pM$ with $\langle v, v \rangle = 0$) makes these spaces difficult to interpret visually (although see [Ega17] for a literary interpretation). However, there is one class of pseudo-riemannian manifolds for which there is a clear interpretation of what the intrinsic view looks like: lorentzian manifolds. These have bilinear forms of signature $(n-1, 1)$ and are the basic models of space-time in relativistic physics.

In relativity, light travels along the null geodesics (geodesics with null tangents) in a lorentzian manifold, and so the intrinsic view may be computed by ray-marching starting with the lightcone of null vectors in the tangent space of the viewer. In the real world, we see light that travels along null geodesics in a lorentzian four-manifold. Arguably, then, it is more natural to consider the inside view of a lorentzian four-manifold rather than of a riemannian three-manifold.

The natural starting place is flat space-time: the Minkowski space $\mathbb{R}^{3,1}$. Ray-marching along lightcones in this geometry provides a method of simulating the inside view in special relativity. Previous visualization work in special relativity includes [SSM07, MWM$^+$10, MGW10, SCTK16]. Generalizing to homogeneous space-times of constant curvature, one could produce intrinsic simulations of de Sitter and anti-de Sitter space-time. Many of the methods described in Section 3 can be adapted to this setting. All three of these have natural projective models in $\mathbb{R}^5$, and explicit descriptions for their null geodesics and isometry groups are well known (see for example, [Sok16] and [KOP02]).

Beyond these, the classification of general lorentzian homogenous four-manifolds has been completed [CZ14], although it is more complex than the case of riemannian three-manifolds discussed above. In all such manifolds we may use analogs of the techniques introduced in Section 3 to simplify computations.

12.6. **Non homogeneous geometries.** Giving up on symmetry, there are many non-homogeneous riemannian and lorentzian manifolds for which intrinsic views may prove useful. Examples include watching a three-manifold evolve under the Ricci flow, analyzing collapsing space-times, or space-times with singularities (black holes). In most cases, the lack of symmetry forces us to use numeric solutions for the geodesic flow. However, there are also interesting non-homogeneous spaces with exactly solvable geodesic flow. These include the matrix group $\mathrm{SL}(2, \mathbb{R})$ with the metric it inherits from the 2×2 matrices $\mathcal{M}_{2,2}(\mathbb{R}) \cong \mathbb{R}^4$ as a hypersurface. However, these spaces all present considerable difficulties for the methods outlined in Section 3, and will require more work.

Appendix A. Comparison between methods to integrate the geodesic flow

In this work, whenever possible we have avoided numerical methods for following geodesics and have instead exploited explicit solutions of the geodesic flow. This allows us to quickly and accurately ray-march long distances, and thus render scenes with distant objects [CMST20a, CMST20b]. To support our choice, we ran some numerical experiments. We explain our protocol below.

Remark A.1. We do not claim to give a comprehensive and rigorous comparison of the various methods to integrate the geodesic flow. The computations here are made in Python (using Numpy long double floats) on a standard desktop computer. We do not use the GPU, and no parallel computing is involved. Nevertheless, we can use these experiments to compare the relative efficiency of the algorithms. ♢

A.1. **Experimental protocol.** Let (G, X) be one of the Thurston geometries. We fix an integer $N \in \mathbb{N}$ and a time $t \in \mathbb{R}_+$. We compare four methods: using exact formulas, Euler's method, and the Runge–Kutta methods of order two and four. For the numerical methods, we also compare different step sizes Δt.

We first generate a list V of N unit tangent vectors at the origin $o \in X$, chosen uniformly and independently at random. For each experiment $\mathcal{E}$ in each of the Tables 7, 8, 9, and 10, we fix a method and (for the numerical methods) a time-step. We then do the following computations.

- For each direction $v \in V$, we flow from o for time t and record the final position. This yields a list $Q_{\mathcal{E}}$ of N points. We also record the time needed to compute $Q_{\mathcal{E}}$.
- Next, for each $q_{\mathcal{E}} \in Q_{\mathcal{E}}$, we measure the error of $q_{\mathcal{E}}$ with respect to the exact flow. We discuss our choice of error measurements in Section A.2.
- Finally, we compute the maximal and mean errors for the set $Q_{\mathcal{E}}$.

A.2. **Measuring errors.** We calculate two different measures of error. Fixing notation, let $v \in V$ be one element in our collection of random tangent vectors and $q_{\mathcal{E}} \in Q_{\mathcal{E}}$ be the point obtained by following the geodesic flow starting at o in the direction of v in for time t in the experiment $\mathcal{E}$.

A.2.1. *Distance error.* We compute the coordinates of the point q obtained by following the geodesic flow starting at o in the direction of v in for time t using the *exact formulas*.

Definition A.2. The *distance error* is the distance in the metric of X between $q_{\mathcal{E}}$ and q. ◊

Remark A.3. One should worry about how accurate our computer's implementation of the exact formulas is. As mentioned in Remark A.1, we use NumPy for all of our calculations here, and long doubles, giving us around 19 decimal digits of accuracy. While we have not looked into the actual implementations of the functions we use, we would certainly hope that these implementations lose at most one or two digits of accuracy on each operation. Of course the results of these functions then need to be combined, which compounds the errors. Without using interval arithmetic, it is hard to say how accurate our final results are. However, as we will see in our experiments, with small values of t and small step size Δt, our exact calculation matches Runge–Kutta of order four ($\Delta t = 0.01$) up to a distance error of around 10^{-9} at worst. This provides evidence that our implementation of the exact formulas are at least this accurate in comparison with the true values, since the exact and Runge–Kutta methods take very different routes to their results. ◊

The distance error is natural, but the results are sometimes difficult to interpret because of our lack of intuition in those geometries. Moreover, our eyes place far more importance in which direction one sees an object in, over how far away it is. If we have an error in distance, then perhaps at worst the effect of fog is slightly incorrect. An error in direction could cause us to see objects in the wrong place, or distorted in some way. To better measure this, we introduce our second error measurement.

A.2.2. *Angle error.* We compute the tangent vector v' so that the *exact* geodesic flow starting from o in the direction of v' hits the point $q_{\mathcal{E}}$. When there are multiple such tangent vectors v', we choose the one which is closest (in angle) to v.

Definition A.4. The *angle error* is the angle between v and v'. ◊

Following our goal of producing accurate images in Section 1.2, it is reasonable to require that each pixel of our screen be colored according to an object that should be visible through that pixel. Therefore, given the resolution of our screen and a desired field of view, one can calculate a maximum acceptable angle error, as follows.

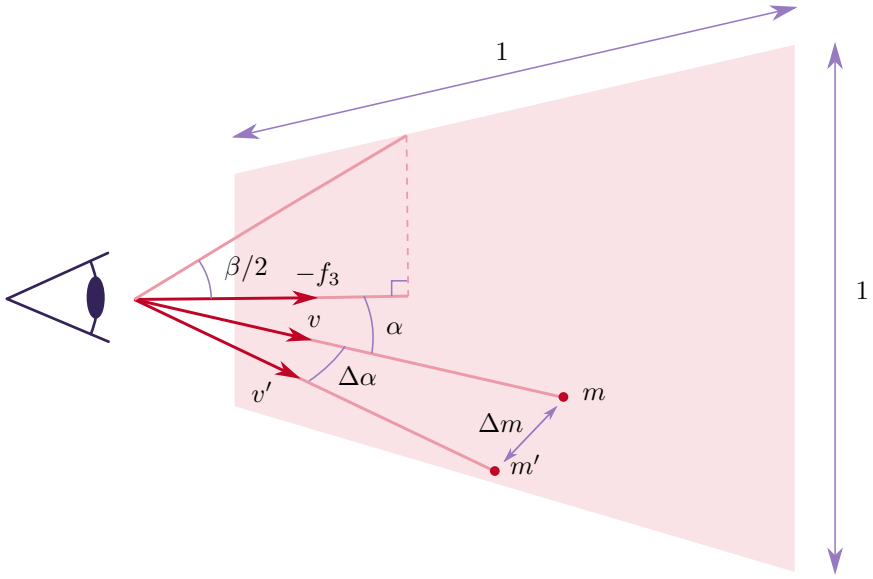


FIGURE A.1. Relation between angle error and resolution.

Let m and m' be the locations on our screen corresponding to the directions v and v'. See Figure A.1. Suppose that the width of the screen is one unit. The distance between m and m' can be estimated as follows.

Assume that the field of view is β. Let α be the angle between v and the vector $-f_3$ pointing forwards, and let $\Delta\alpha$ be the angle between v and v'. Then the distance $\mathrm{dist}(m, m')$ is at most

$$\frac{|\tan(\alpha + \Delta\alpha) - \tan(\alpha)|}{2\tan(\beta/2)}.$$

For a fixed angle error $\Delta\alpha$, this quantity is the largest when m is on the border of the screen, that is when $\alpha = \beta/2$. Hence the worst error Δm for m is related to $\Delta\alpha$ by

$$\tan(\Delta\alpha) = \frac{\Delta m \sin\beta}{1 + 2\Delta m \sin^2(\beta/2)}$$

For the picture on the screen to be accurate, we need Δm to be less than half the width of a pixel. For example, fixing the field of view at $\beta = 100°$, the maximum acceptable angle error (in degrees) is

- $\Delta\alpha \approx$ `3e-02` to produce a 1000×1000 pixel image,
- $\Delta\alpha \approx$ `6e-03` to produce a 5000×5000 pixel image.

Remark A.5. When following the geodesic flow for a time t which is smaller than the injectivity radius of the geometry X, there is only one

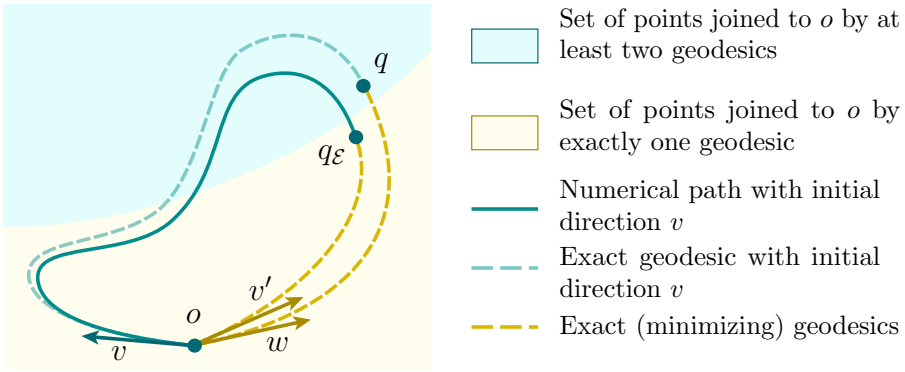


FIGURE A.2. A schematic picture of an exceptional sample.

exact geodesic joining o to q_ε. Here there is no choice in the definition of v'. For longer flow times, a new phenomenon arises. As usual, we numerically compute the path starting at o in the direction v and reach the point q_ε. This path approximates the exact geodesic ray starting at o in the direction v, which reaches the point q. See Figure A.2. Suppose that multiple exact geodesics join o to q, so that in addition to the direction v, we can also reach q along a geodesic with starting direction w. Suppose also that due to numerical errors, only one exact geodesic joins o to q_ε. The vector v' is then the only possible initial direction pointing from o to q_ε. It will be close to one of v and w, but it may be close to the wrong one: w. Thus, the angle between v and v' can be very large. However, the visual effect of this error will be indistinguishable from similar inaccuracies with the same distance error.

This situation is very rare: our numerical path must approximate a non-minimizing geodesic, with endpoint q_ε close to the boundary of one of the sets

$$X_n = \{x \in X \mid x \text{ is joined to } o \text{ by } n \text{ geodesics}\}$$

For the Thurston geometries, these boundaries form a zero-measure set.

In our results, we indicate the number of points for which the angle between v and v' is more than a large threshold, for example 40°. These cases correspond to the situation described above. We compute the maximal and mean errors excluding these exceptional samples. ◊

A.3. **Results.** We carried out our protocol for Nil and $\mathrm{SL}(2,\mathbb{R})$. We computed the distance and angle errors using the numerical methods described in Sections 9.5 and 10.5 respectively. See Tables 7, 8, 9, and 10. We made sure that the errors coming from use of Newton's method to calculate v' are negligible compared to the results. We ran

the experiments for time $t = 6$ and $t = 10$. Note that 6 is less than the injectivity radius (2π for both Nil and $\widetilde{\mathrm{SL}}(2, \mathbb{R})$).

Method	Δt	**Time needed (in s.)**	**Maximal distance error**	**Mean distance error**	**Number of directions off ($\Delta\alpha > 60°$)**	**Maximal angle error (in °)**	**Mean angle error (in °)**
Exact flow	–	0.2	–	–	–	–	–
Euler	0.1	12.0	8.4e-01	5.2e-01	0	8.7e+00	4.4e+00
Euler	0.01	115.3	8.6e-02	5.2e-02	0	9.4e-01	4.8e-01
Runge–Kutta 2	0.1	17.8	9.3e-03	5.0e-03	0	1.7e-01	3.1e-02
Runge–Kutta 2	0.01	176.0	9.6e-05	5.1e-05	0	1.7e-03	3.0e-04
Runge–Kutta 4	0.1	32.4	8.2e-06	4.2e-06	0	8.1e-05	1.8e-05
Runge–Kutta 4	0.01	323.3	8.2e-10	4.2e-10	0	2.7e-08	1.2e-09

TABLE 7. Integrating the geodesic flow in Nil. Computation made with $N = 10,000$ and $t = 6$.

Method	Δt	**Time needed (in s.)**	**Maximal distance error**	**Mean distance error**	**Number of directions off ($\Delta\alpha > 60°$)**	**Maximal angle error (in °)**	**Mean angle error (in °)**
Exact flow	–	0.3	–	–	–	–	–
Euler	0.1	20.4	3.8e+00	2.2e+00	690	6.0e+01	1.4e+01
Euler	0.01	191.8	4.3e-01	2.2e-01	71	5.3e+01	2.4e+00
Runge–Kutta 2	0.1	30.0	3.1e-02	1.4e-02	186	2.5e+00	7.5e-02
Runge–Kutta 2	0.01	297.0	3.3e-04	1.4e-04	19	1.5e-01	9.8e-04
Runge–Kutta 4	0.1	54.7	2.8e-05	1.2e-05	0	2.0e-02	5.7e-05
Runge–Kutta 4	0.01	540.1	2.8e-09	1.2e-09	0	3.5e-06	4.5e-09

TABLE 8. Integrating the geodesic flow in Nil. Computation made with $N = 10,000$ and $t = 10$.

A.4. **Discussion.** The maximal angle errors for Euler's method do not produce accurate 1000×1000 pixel images with field of view 100°, even for small flow time ($t = 6$) and small time step ($\Delta t = 0.01$). The

Method	Δt	**Time needed (in s.)**	**Maximal distance error**	**Mean distance error**	**Number of directions off ($\Delta\alpha > 40°$)**	**Maximal angle error (in °)**	**Mean angle error (in °)**
Exact flow	–	0.8	–	–	–	–	–
Euler	0.1	33.2	3.7e+00	2.7e+00	0	5.2e+01	5.4e+00
Euler	0.01	325.1	6.6e-01	3.9e-01	0	2.7e+00	7.1e-01
Runge–Kutta 2	0.1	49.0	4.8e-02	2.8e-02	0	1.0e+00	1.3e-01
Runge–Kutta 2	0.01	485.5	6.6e-04	3.6e-04	0	9.4e-03	1.2e-03
Runge–Kutta 4	0.1	83.9	4.1e-05	2.4e-05	0	2.0e-03	1.8e-04
Runge–Kutta 4	0.01	817.5	4.7e-09	2.3e-09	0	1.9e-07	1.3e-08

TABLE 9. Integrating the geodesic flow in $\widetilde{\mathrm{SL}}(2,\mathbb{R})$. Computation made with $N = 10,000$ and $t = 6$.

Method	Δt	**Time needed (in s.)**	**Maximal distance error**	**Mean distance error**	**Number of directions off ($\Delta\alpha > 40°$)**	**Maximal angle error (in °)**	**Mean angle error (in °)**
Exact flow	–	0.7	–	–	–	–	–
Euler	0.1	53.7	1.1e+01	7.9e+00	594	4.0e+01	8.3e+00
Euler	0.01	523.3	6.2e+00	3.6e+00	74	3.7e+01	2.2e+00
Runge–Kutta 2	0.1	78.5	9.0e-01	3.3e-01	180	4.0e+01	5.9e-01
Runge–Kutta 2	0.01	775.1	1.3e-02	4.7e-03	25	3.4e-01	2.9e-03
Runge–Kutta 4	0.1	133.1	8.6e-04	3.6e-04	0	2.8e-01	9.1e-04
Runge–Kutta 4	0.01	1,316.9	7.4e-08	3.1e-08	0	1.1e-04	7.5e-08

TABLE 10. Integrating the geodesic flow in $\widetilde{\mathrm{SL}}(2,\mathbb{R})$. Computation made with $N = 10,000$ and $t = 10$.

Runge–Kutta method of order two is accurate enough for small distance only (and sometimes only for smaller step like $\Delta t = 0.01$). For medium distances ($t = 10$), in $\widetilde{\mathrm{SL}}(2,\mathbb{R})$ only the Runge–Kutta method of order four with step $\Delta t = 0.01$ meets our criterion. The Runge–Kutta method of order four is also the only one that does not produce exceptional points in the sense of Remark A.5.

In terms of the time needed to run the computations, the exact method is superior to the numerical ones. Lookup tables may be precomputed to avoid long calculation times, although one should then also worry about inaccuracies introduced by interpolation.

References

[Abr66] *Handbook of mathematical functions, with formulas, graphs and mathematical tables*, Edited by Milton Abramowitz and Irene A. Stegun. Fifth printing, with corrections. National Bureau of Standards Applied Mathematics Series, Vol. 55, National Bureau of Standards, Washington, D.C., (for sale by the Superintendent of Documents, U.S. Government Printing Office, Washington, D.C., 20402), 1966. MR 0208798 [115]

[BBC72] T.W. Bradley, C.J. Bradley, and A.P. Cracknell, *The mathematical theory of symmetry in solids: Representation theory for point groups and space groups*, Clarendon Press, 1972. [61]

[Ber15] Pierre Berger, *Espaces Imaginaires*, `http://espaces-imaginaires.fr`, 2015. [9]

[BH99] Martin R. Bridson and André Haefliger, *Metric spaces of non-positive curvature*, Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences], vol. 319, Springer-Verlag, Berlin, 1999. MR 1744486 [61]

[BLV15] Pierre Berger, Alex Laier, and Luiz Velho, *An image-space algorithm for immersive views in 3-manifolds and orbifolds*, Vis. Comput. **31** (2015), no. 1, 93–104. [9, 28]

[Bon09] Francis Bonahon, *Low-dimensional geometry*, Student Mathematical Library, vol. 49, American Mathematical Society, Providence, RI; Institute for Advanced Study (IAS), Princeton, NJ, 2009, From Euclidean surfaces to hyperbolic knots, IAS/Park City Mathematical Subseries. MR 2522946 [37]

[BS07] Attila Bölcskei and Brigitta Szilágyi, *Frenet formulas and geodesics in Sol geometry*, Beiträge Algebra Geom. **48** (2007), no. 2, 411–421. MR 2364799 [113]

[Bul65] Roland Bulirsch, *Numerical calculation of elliptic integrals and elliptic functions*, Numer. Math. **7** (1965), 78–90. MR 175284 [115]

[CDGW] Marc Culler, Nathan M. Dunfield, Matthias Goerner, and Jeffrey R. Weeks, *SnapPy, a computer program for studying the geometry and topology of* 3*-manifolds*, Available at `http://snappy.computop.org` (06/16/2020, version 2.8). [8, 37, 65]

[CHK00] Daryl Cooper, Craig D. Hodgson, and Steven P. Kerckhoff, *Chapter 2. orbifolds*, MSJ Memoirs, vol. Volume 5, The Mathematical Society of Japan, Tokyo, Japan, 2000. [37]

[CMST20a] Rémi Coulon, Elisabetta Matsumoto, Henry Segerman, and Steve Trettel, *Non-euclidean virtual reality III: Nil*, Proceedings of Bridges 2020: Mathematics, Art, Music, Architecture, Education, Culture (Phoenix, Arizona), Tessellations Publishing, 2020, pp. 153–160. [4, 10, 25, 129]

[CMST20b] ———, *Non-euclidean virtual reality IV: Sol*, Proceedings of Bridges 2020: Mathematics, Art, Music, Architecture, Education, Culture (Phoenix, Arizona), Tessellations Publishing, 2020, pp. 161–168. [4, 10, 25, 129]

[CMST20c] Rémi Coulon, Elisabetta A. Matsumoto, Henry Segerman, and Steve Trettel, *Non-euclidean VR*, `https://github.com/henryseg/non-euclidean_VR`, 2020. [4]

[CS19] Matei P. Coiculescu and Richard Evan Schwartz, *The spheres of Sol*, 2019, arXiv:1911.04003. [19]

[CZ14] Giovanni Calvaruso and Amirhesam Zaeim, *Four-dimensional homogeneous lorentzian manifolds*, Monatshefte für Mathematik **174** (2014). [128]

[DESS09] Blaženka Divjak, Zlatko Erjavec, Barnabás Szabolcs, and Brigitta Szilágyi, *Geodesics and geodesic spheres in* $\widetilde{\mathrm{SL}(2,\mathbb{R})}$ *geometry*, Math. Commun. **14** (2009), no. 2, 413–424. MR 2743187 [96]

[Ega17] Greg Egan, *Dichronauts*, Night Shade Books, 2017. [128]

[FWW02] William Floyd, Brian Weber, and Jeffrey Weeks, *The Achilles' heel of* $\mathrm{O}(3,1)$*?*, Experiment. Math. **11** (2002), no. 1, 91–97. MR 1960304 [16]

[Gen16] Andrew Liang Li Geng, *5-dimensional geometries I: the general classification*, 2016, arXiv:1605.07545. [6]

[Gol] William Goldman, *Geometric structures on manifolds*, `http://www.math.umd.edu/~wmg/gstom.pdf`. [27]

[GPE17] Bor Gregorcic, Gorazd Planinsic, and Eugenia Etkina, *Doing science by waving hands: Talk, symbiotic gesture, and interaction with digital content as resources in student inquiry*, Physical Review Physics Education Research **13** (2017), no. 2, 020104. [4]

[Gra83] Matthew Aaron Grayson, *Geometry and growth in three dimensions*, ProQuest LLC, Ann Arbor, MI, 1983, Thesis (Ph.D.)–Princeton University. MR 2632777 [18]

[HHMS17a] Vi Hart, Andrea Hawksley, Elisabetta Matsumoto, and Henry Segerman, *Non-euclidean virtual reality I: Explorations of* $\mathbb{H}^3$, Proceedings of Bridges 2017: Mathematics, Art, Music, Architecture, Education, Culture (Phoenix, Arizona), Tessellations Publishing, 2017, Available online at `http://archive.bridgesmathart.org/2017/bridges2017-33.pdf`, pp. 33–40. [8]

[HHMS17b] ———, *Non-euclidean virtual reality II: Explorations of* $\mathbb{H}^2 \times \mathbb{E}$, Proceedings of Bridges 2017: Mathematics, Art, Music, Architecture, Education, Culture (Phoenix, Arizona), Tessellations Publishing, 2017, Available online at `http://archive.bridgesmathart.org/2017/bridges2017-41.pdf`, pp. 41–48. [8, 24]

[Hil02] J. A. Hillman, *Four-manifolds, geometries and knots*, Geometry & Topology Monographs, vol. 5, Geometry & Topology Publications, Coventry, 2002. MR 1943724 [6]

[HSK89] J. C. Hart, D. J. Sandin, and L. H. Kauffman, *Ray tracing deterministic 3-d fractals*, SIGGRAPH Comput. Graph. **23** (1989), no. 3, 289–296. [10]

[Jac29] C.G.J. Jacobi, *Fundamenta nova theoriae functionum ellipticarum*, Regiomonti, 1829. [114]

[JGMR17] Mina C Johnson-Glenberg and Colleen Megowan-Romanowicz, *Embodied science and mixed reality: How gesture and motion capture affect physics education*, Cognitive Research: Principles and Implications **2** (2017), no. 1, 24. [4]

[KCK19] Eryk Kopczyński and Dorota Celińska-Kopczyńska, *HyperRogue: Thurston Geometries*, `http://zenorogue.blogspot.com/2019/09/hyperrogue-112-thurston-geometries-free.html`, 2019. [9]

[KCK20] ———, *Real-time visualization in non-isotropic geometries*, 2020, arXiv:2002.09533. [9, 28]

[KOP02] Yoon-bai Kim, Chae Young Oh, and Namil Park, *Classical geometry of de Sitter space-time: An Introductory review.* [128]

[Law89] Derek F. Lawden, *Elliptic functions and applications*, Applied Mathematical Sciences, vol. 80, Springer-Verlag, New York, 1989. MR 1007595 [114]

[LTWJ16] Robb Lindgren, Michael Tscholl, Shuai Wang, and Emily Johnson, *Enhancing learning and engagement through embodied interaction within a mixed reality simulation*, Computers & Education **95** (2016), 174–187. [4]

[Lum19] Jean-Pierre Luminet, *An illustrated history of black hole imaging : Personal recollections (1972-2002)*, 2019, arXiv:1902.11196. [8]

[Mag19] MagmaMcFry, *SolvView*, `https://github.com/MagmaMcFry/SolvView`, 2019. [9]

[MGW10] T. Müller, S. Grottel, and D. Weiskopf, *Special relativistic visualization by local ray tracing*, IEEE Transactions on Visualization and Computer Graphics **16** (2010), no. 6, 1243–1250. [128]

[MLP+14] Tamara Munzner, Stuart Levy, Mark Phillips, Celeste Fowler, Charlie Gunn, Nathaniel Thurston, Daniel Krech, Scott Wisdom, Daeron Meyer, and Tim Rowley, *Geomview: An Interactive 3D Viewing Program for Unix*, `http://www.geomview.org`, 1991–2014. [8]

[Mol97] Emil Molnár, *The projective interpretation of the eight* 3*-dimensional homogeneous geometries*, Beiträge Algebra Geom. **38** (1997), no. 2, 261–288. MR 1473106 [30]

[Mol03] ———, *On Nil geometry*, Period. Polytech. Mech. Engrg. **47** (2003), no. 1, 41–49. MR 2045762 [79]

[MWM+10] D. McGrath, M. Wegener, T. J. McIntyre, C. Savage, and M. Williamson, *Student experiences of virtual reality: A case study in learning special relativity*, American Journal of Physics **78** (2010), 862–868. [128]

[NdSVa] Tiago Novello, Vinicius da Silva, and Luiz Velho, *Design and Visualization of Riemannian Metrics*, `https://www.visgraf.impa.br/ray-vr/?page_id=378`. [9]

[NdSVb] ———, *Ray Tracing in Nil, Sol, and SL2(R) Geometries*, `https://www.visgraf.impa.br/ray-vr/?page_id=252`. [9]

[NdSV20] Tiago Novello, Vinícius da Silva, and Luiz Velho, *How to see the eight thurston geometries*, 2020, arXiv:2005.12772. [9]

[NS17] Roice Nelson and Henry Segerman, *Visualizing hyperbolic honeycombs*, Journal of Mathematics and the Arts **11** (2017), no. 1, 4–39. [65]

[NSW18] Roice Nelson, Henry Segerman, and Michael Woodard, *hypVR-Ray*, `https://github.com/mtwoodard/hypVR-Ray`, 2018. [8]

[OLBC10] Frank W. J. Olver, Daniel W. Lozier, Ronald F. Boisvert, and Charles W. Clark (eds.), *NIST handbook of mathematical functions*, U.S. Department of Commerce, National Institute of Standards and Technology, Washington, DC; Cambridge University Press, Cambridge, 2010, With 1 CD-ROM (Windows, Macintosh and UNIX). MR 2723248 [115]

[OM49] Fritz Oberhettinger and Wilhelm Magnus, *Anwendung der elliptischen Funktionen in Physik und Technik*, Springer-Verlag, Berlin, 1949. MR 0031129 [114]

[Pat96] Victor Patrangenaru, *Classifying* 3- *and* 4*-dimensional homogeneous riemannian manifolds by cartan triples.*, Pacific J. Math. **173** (1996), no. 2, 511–532. [7, 127]

[Per02] Grisha Perelman, *The entropy formula for the Ricci flow and its geometric applications*, 2002, arXiv:0211159. [4]

[Per03a] ———, *Finite extinction time for the solutions to the Ricci flow on certain three-manifolds*, 2003, arXiv:0307245. [4]

[Per03b] ———, *Ricci flow with surgery on three-manifolds*, 2003, arXiv:0303109. [4]

[PG92] Mark Phillips and Charlie Gunn, *Visualizing hyperbolic space: Unusual uses of 4x4 matrices*, Proceedings of the 1992 Symposium on Interactive 3D Graphics (New York, NY, USA), I3D '92, Association for Computing Machinery, 1992, p. 209–214. [8]

[Pho75] Bui Tuong Phong, *Illumination for computer generated pictures*, Commun. ACM **18** (1975), no. 6, 311–317. [12, 40]

[Quia] Inigo Quilez, *Distance functions*, `https://iquilezles.org/www/articles/distfunctions/distfunctions.htm`. [11, 26]

[Quib] ———, *Soft shadows in raymarched SDFs*, `https://www.iquilezles.org/www/articles/rmshadows/rmshadows.htm`. [42]

[Sco83] Peter Scott, *The geometries of 3-manifolds*, The Bulletin of the London Mathematical Society **15** (1983), no. 5, 401–487 (English). [63, 75, 87, 107, 122]

[SCTK16] Zachary W. Sherin, Ryan Cheu, Philip Tan, and Gerd Kortemeyer, *Visualizing relativity: The openrelativity project*, American Journal of Physics **84** (2016), 369–374. [128]

[Sok16] Leszek M. Sokolowski, *The bizarre anti-de Sitter spacetime*, `https://arxiv.org/pdf/1611.01118.pdf`, 2016. [128]

[SSM07] C. M. Savage, A. Searle, and L. McCalman, *Real time relativity: Exploratory learning of special relativity*, American Journal of Physics **75** (2007), 791–798. [128]

[Thu97] William P. Thurston, *Three-dimensional geometry and topology. Vol. 1*, Princeton Mathematical Series, vol. 35, Princeton University Press, Princeton, NJ, 1997, Edited by Silvio Levy. MR 1435975 [37]

[Thu98] ———, *How to see* 3*-manifolds*, vol. 15, 1998, Topology of the Universe Conference (Cleveland, OH, 1997), pp. 2545–2571. MR 1649658 [8, 37, 38]

[Tre18] Steve Trettel, *Life in hyperbolic space*, `http://www.stevejtrettel.site/LifeInHyperbolic.pdf`, 2018. [21]

[Tro98] Marc Troyanov, *L'horizon de Sol*, Exposition. Math. **16** (1998), no. 5, 441–479. MR 1656902 [113]

[Wee] Jeffrey Weeks, *Curved Spaces*, a flight simulator for multiconnected universes, available from `http://www.geometrygames.org/CurvedSpaces/`. [8, 15]

[Wee02] ———, *Real-time rendering in curved spaces*, IEEE Computer Graphics and Applications **22** (2002), no. 6, 90–99. [15, 61]

[Won] Jamie Wong, *Ray marching and signed distance functions*, `http://jamie-wong.com/2016/07/15/ray-marching-signed-distance-functions/`. [10]

Rémi Coulon
Univ Rennes, CNRS
IRMAR - UMR 6625
F-35000 Rennes, France
`remi.coulon@univ-rennes1.fr`
`http://rcoulon.perso.math.cnrs.fr`

Elisabetta A. Matsumoto
School of Physics
Georgia Institute of Technology
837 State Street, Atlanta, GA, 30332, USA
`sabetta@gatech.edu`
`http://matsumoto.gatech.edu`

Henry Segerman
Department of Mathematics
Oklahoma State University
Stillwater, OK, 74078, USA
`segerman@math.okstate.edu`
`https://math.okstate.edu/people/segerman/`

Steve J. Trettel
Stanford University
450 Jane Stanford Way,
Stanford, CA 94305
`trettel@stanford.edu`
`http://stevejtrettel.site`