

## CHAPTER 6

# Circle Packing

KEN STEPHENSON (text and software)

Complex analysis is in many ways the ultimate in “continuous” mathematics. It presents you with a *smooth* world: continuous variables, infinitely differentiable functions, smooth surfaces, plus a complex arithmetic with its power series, line integrals, and all sorts of handy formulas. In this chapter, however, we will develop a quite different view of the topic. Here the geometry behind analytic functions moves strongly to the fore as we see how one might “discretize” complex analysis.

An analogy to keep in mind is that of a mountain stream. In normal experience we treat the stream as a continuous medium and use continuous variables and functions to understand it — pressure, velocity, vorticity — modeled, perhaps, by Navier-Stokes and other PDE’s. Yet we know that the stream is in fact a collection of individual water molecules, a “discrete” medium, if you will. The macro behavior, the waves, eddies, and currents, emerge from the myriad local interactions among the discrete water molecules. So one might ask: *what purely local rules of interaction among these molecules could lead to the observed global behavior?*

In this chapter, we address this issue for continuous analytic functions. Circles will be our molecules, and “packing” conditions will provide the local rules for their interaction. Circle packings, configurations of myriad individual circles, each interacting only with its neighbors, will manifest macro behavior that we will recognize as a version of analyticity. Later in the chapter we will also see that our discrete objects converge under appropriate refinement to the familiar continuous counterparts. In effect, then, circle packing provides a *quantum complex analysis*, one that is classical in the limit.

This is a topic ideal for visualization and experimentation — in fact, it is accessible only through the computer. You will therefore find yourself tethered to the software more closely than in the earlier chapters, so don’t bypass those ***Try it out!*** signs. Persevere and you may develop a new appreciation for complex analysis. We will jump right in with a quick experiment from the genesis of the topic in  Cookie . ***Try it out!*** .

### 6.1. First Impressions

Circle packings were introduced to analysis by Bill Thurston in a talk in 1985. The context was the conformal mapping of bounded, simply connected, plane regions  $\Omega$

onto the unit disc  $\mathbb{D}$ . The central result is the famous Riemann Mapping Theorem stated in the Appendix, page 418. (Alternate statements are given in Theorem 3.33, page 210, and Theorem 4.16, page 250). For us, the preferred statement is this: *Given  $\Omega$  and designated points  $w_0, w_1$  in  $\Omega$ , there exists a unique one-to-one analytic mapping  $F$  from  $\mathbb{D}$  onto  $\Omega$  so that  $F(0) = w_0$  and so that  $w_1$  lies on the image of the positive real axis.* Figure 6.1 therefore illustrates the **Discrete** Riemann Mapping Theorem.

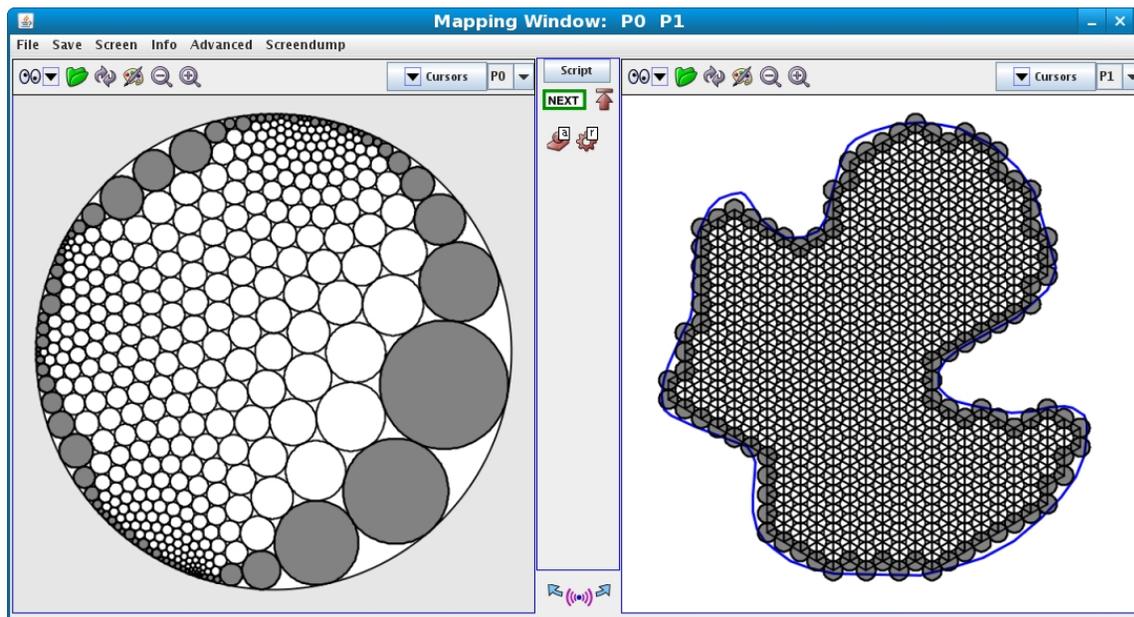


FIGURE 6.1. A conformal map display, domain (left) and range (right)

Figure 6.1 shows the mapping window as displayed by our experimental testbench, the software package *CirclePack*. Typical of our practice, the domain circle packing will be on the left in such side-by-side displays, and the range packing will be on the right. In this instance, the domain is  $\mathbb{D}$  and the range is the given region  $\Omega$ . The “function” itself can be interpreted in various ways. Each circle on the left has a corresponding circle on the right, so one can map circles to circles. Alternately, as we will see later, one can map the triangles formed by the circles on the left to those formed by the corresponding circles on the right. In any case, the two packings provide, in essence, a map from  $\mathbb{D}$  onto  $\Omega$ , a “discrete conformal map”. We will see the details shortly, but assuming you have run **S**Cookie**S** and have this example before you on the computer, try clicking the left mouse button on some circle in one packing: both it and the corresponding circle of the other packing should be highlighted (likewise, for faces, using the middle mouse button). *Try it out!*

**About *CirclePack*:** A few words are in order about *CirclePack* before we go on. Figure 6.2 is a screenshot showing the context for these mapping images, with various frames and panels for running the software. Unlike most of this book’s software, *CirclePack* is a Java “application” rather than a Java “applet”: it requires access to the file system for file manipulations, it is larger, more comprehensive (and reconfigurable), and it avails itself of shared C/C++ libraries for computationally intense experiments. It also has an important scripting feature: “script” files with prepared sequences of commands are provided with the book to guide your experiments. These files have file extension “.xmd” and are indicated in the text by  $\textcircled{S}$ Name $\textcircled{S}$ . To run an experiment, then, you start *CirclePack*, from within it you load Name.xmd, then you click through the prepared commands.

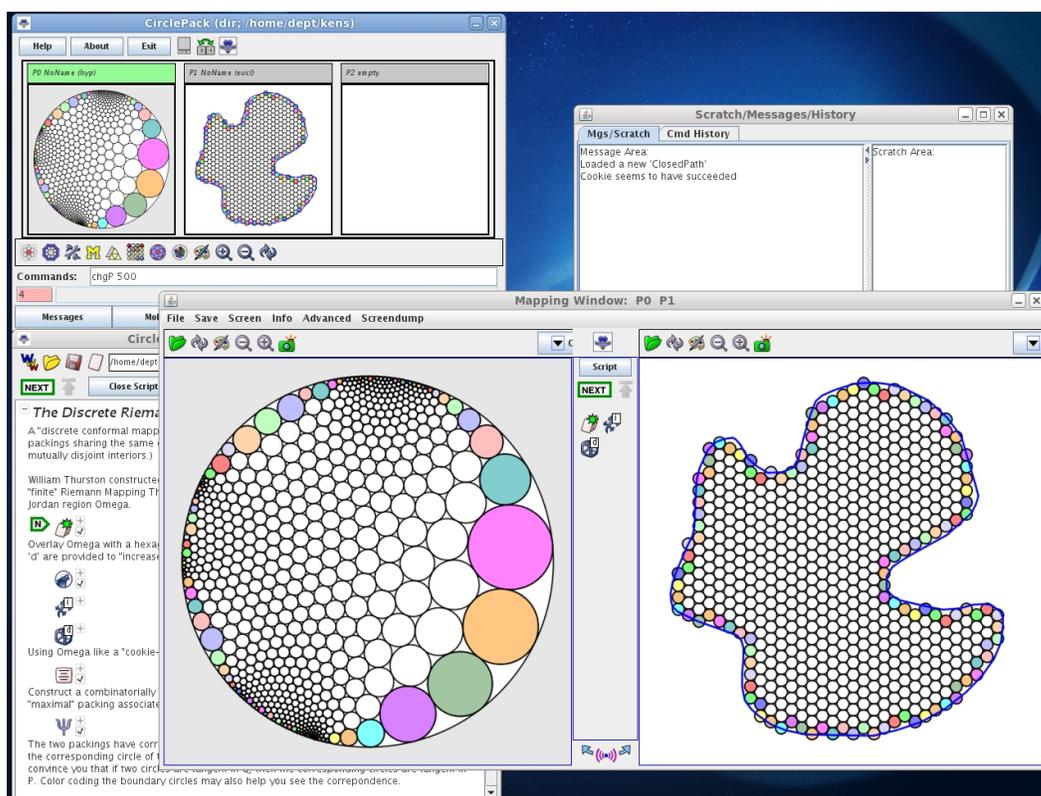


FIGURE 6.2. *CirclePack* in “domain/range” window mode

To recreate Figure 6.1, run the script  $\textcircled{S}$ Cookie $\textcircled{S}$ . This is a very structured experiment, so move *CirclePack* through the commands by successively pressing the “NEXT” button (or press “enter” when the mouse is in the canvas) until the experiment is complete. **Try it out!** (If things become jumbled in *CirclePack*— and they sometimes will! — it may be that you have outrun the computations: just click the “up” icon to

restart at the beginning of the script again, or if things have really gone sour, exit and restart *CirclePack*.)

Even in the very passive experimental mode of  **Cookie**  you can interact with the images — moving, focusing, clicking circles and faces, and so forth, as we describe later. However, it is hoped that as your experience and curiosity grow, you move through the following additional stages to gain more experimental independence:

- (1) The “Open Script” button will open the *CirclePack* script window. Here the commands of the script are laid out linearly and are typically accompanied by text explanations. If you didn’t understand what the experiment was purporting to show, try reading along as you execute the commands. Figure 6.3 illustrates the script window, with text, icons that encapsulate the commands, included files, etc.

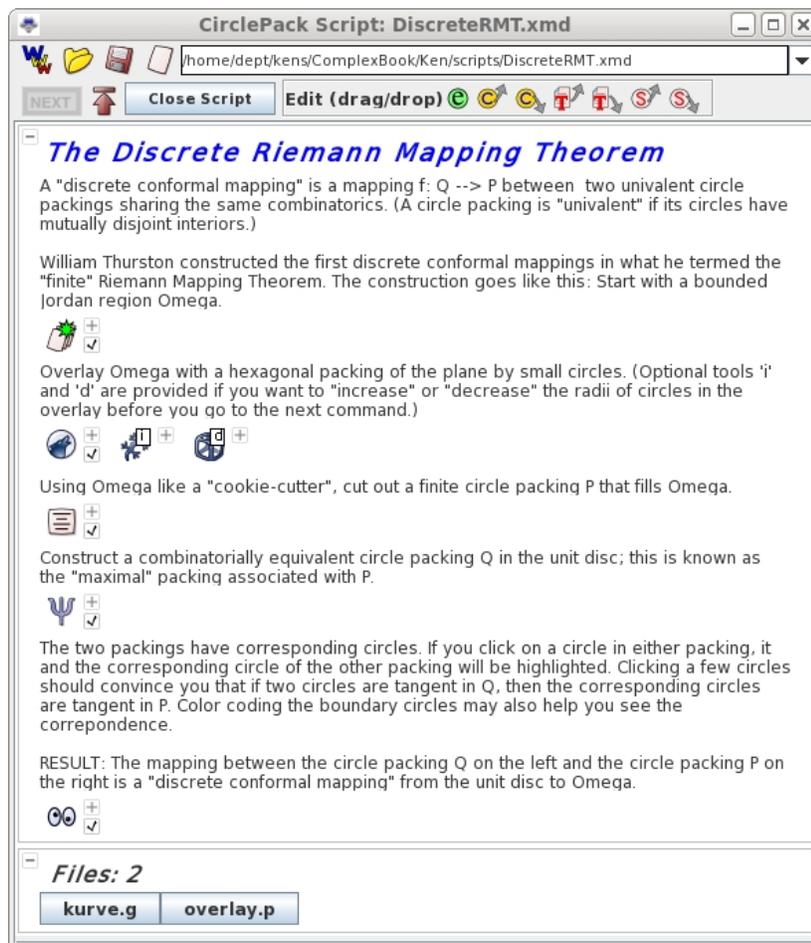


FIGURE 6.3. A typical “script” for *CirclePack*

(2) In the Script you can open a command icon (by clicking its small “+” box) to see the string of commands that it actually executes. For example.,

```
“infile_path gamma.g;cookie;disp -w -c -cf b”
```

tells *CirclePack* to read the path “gamma.g” from the script’s data section, cookie-cut the portion of the resident packing inside the path, then clear the canvas and display the circles of the resulting packing, with the boundary circles “filled”. Modify these commands and you can change the action in the experiment. In many of the scripts, hints are given about alterations you might try. In this first experiment, for example, after overlaying  $\Omega$  with a hexagonal circle packing, optional commands ‘i’ or ‘d’ will increase or decrease, respectively, the sizes of the circles in that overlay. After you have run through the experiment once, re-run it and try these adjustments to see what happens.

(3) If you want to dig deeper, click the “Advanced” button to bring up *CirclePack*’s main control frame, which gives you access to its full functionality. The program actually holds up to three packings, has panels with messages, history, and rudimentary error feedback, “Commands:” lines where you can issue explicit strings of commands, and various “droppable” icons: you just drag one onto a packing window and a prepared string of commands will be applied to that packing.

(4) For the truly ambitious, the Java code for *CirclePack* is freely available under the GNU open source license. It includes a “PackExtender” class which allows anyone (with enough work) to create specialized data structures and commands having full access to the core functionality of *CirclePack*. Several existing PackExtenders will be used in later experiments.

Figure 6.4 provides an annotated screen shot of *CirclePack* with various of its screens and auxiliary panels. (Note that the screenshots are current as of publication, but *CirclePack* may change over time.)

A final note on the scripts: bare-bones scripts are quite easy to write for yourself, but following, reading, and modifying those of others can be a real challenge. The prepared scripts are built for ease of use (believe it or not), so the commands are often long and involved. For example, a simple “disp -w -c” will clear a screen and draw the circles. However, “disp -w -cc20 i -ct5c180 b” will clear the screen then draw the interior circles thicker and in blue, and the boundary circles filled with red. Don’t let the complications bother you — go ahead and start your own scripts with simple commands and small steps. When things go wrong, adjust the commands. As you learn the *CirclePack* ropes, your scripts may become more involved, more sophisticated, but also, more effective.

We are going to move now to the basic definitions and notations we will use in our discrete analytic function theory. Although static images are helpful, there is nothing like running experiments, even in automatic mode, to bring the topic to life.

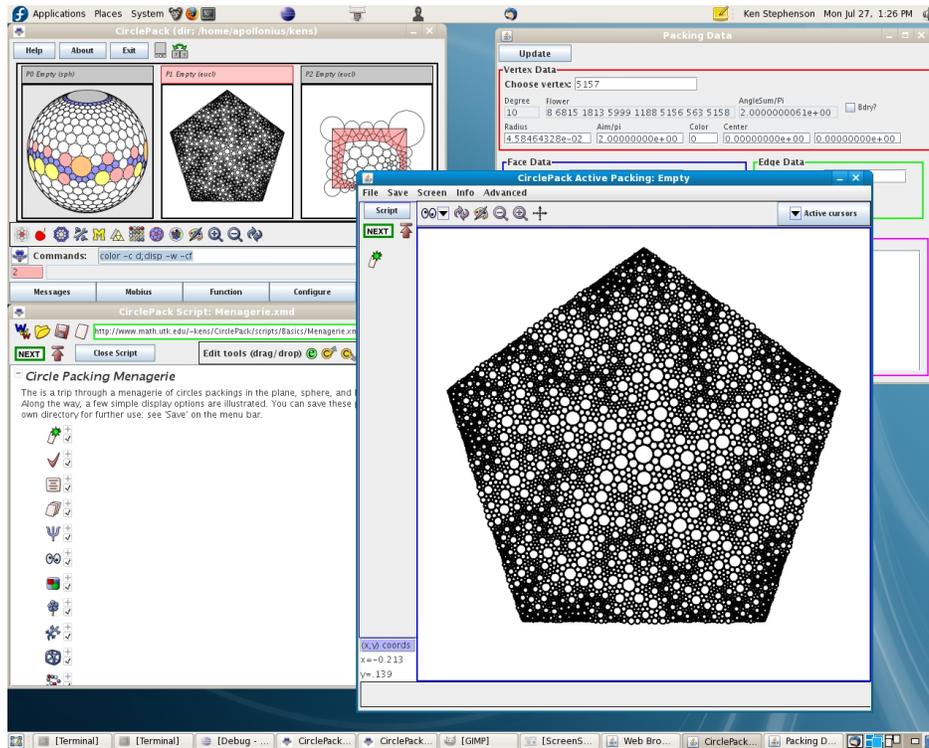


FIGURE 6.4. *CirclePack* in Mapping window mode

The conformal mapping experiment we did above may help, and I certainly hope it motivates the theory we will be investigating later. However, we have next a sequence of experiments tailored to display/explain the basics of the theory.

## 6.2. Basics of Circle Packing

DEFINITION 6.1. A **circle packing** is a configuration of circles with a prescribed pattern of tangencies.

Figure 6.5 illustrates a few circle packings from the script **Menagerie**; run the script to see these and others. *Try it out!* Among the things you might observe as you tour our Menagerie:

- The circles form mutually tangent **triples** and hence have triangular interstices.
- The packings can live in the euclidean plane,  $\mathbb{C}$ , in the hyperbolic plane,  $\mathbb{D}$  (the unit disc in  $\mathbb{C}$ ), or on the Riemann sphere,  $\mathbb{C}$ , represented as the unit sphere at the origin in  $\mathbb{R}^3$ .
- The “pattern” of tangencies shows up in the **carrier** of the packing, the geometric triangles formed by connecting the centers of tangent circles.
- This pattern can be simply connected or multiply connected.
- There are **interior** circles, those completely surrounded by their neighbors, and **boundary** circles, those on the edge of the configuration.
- There are numerous manipulations,

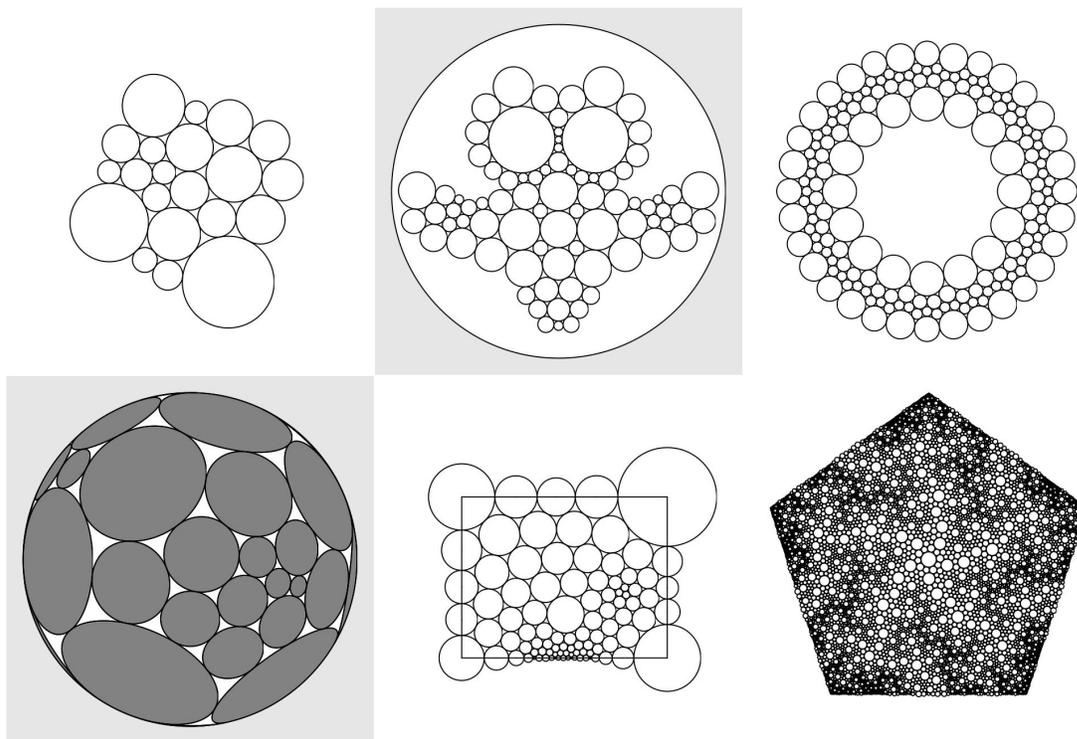


FIGURE 6.5. A small menagerie of circle packings.

colors, and display options for displaying a circle packing and/or its carrier; these are invaluable in highlighting the structures you need to study.

We will have to make formal definitions shortly, but I would suggest that you first play around a little with the CirclePack Owl in [S0w1Play](#). You might just run sequentially through the commands first, but then, by pressing the “Script” button, you can open the Script Window and follow the descriptions associated with the commands. There are suggestions along the way for free-form changes that you can make to see how the packings react. *Try it out!*

Mathematics requires some formalities; annoying at times, they are essential if you want to understand what’s going on and establish results that you and others can depend on. Let me lay out the formal objects involved in circle packing. Follow along in [S Objects](#), where these are illustrated explicitly in turn; read the descriptions in the script as you proceed. In other words, *Try it out!*

- **Complex,  $K$ :** The “prescribed pattern” of tangencies for a circle packing is represented as an abstract triangulation  $K$  of a topological surface (technically, a simplicial 2-complex). This is a combinatorial object:  $K$  has **vertices**  $v$ , **edges**  $\langle v, w \rangle$ , and **faces**  $\langle v, u, w \rangle$  formed by triples of vertices, but does not live in any

geometric setting. It may be finite or infinite, may or may not have boundary, may be simply or multiply connected. We will always assume  $K$  is oriented. Namely, in every face  $\langle v, u, w \rangle$ , the vertices are listed in positive (counterclockwise) order, and these orientations are globally consistent, meaning that if faces  $f, g$  share a directed edge  $e = \langle v, w \rangle$ , then  $e$  is positively oriented in one face and negatively oriented in the other. The **degree** of a vertex  $v$  is the number of edges from  $v$  (hence the number of **neighboring** vertices). As  $K$  is a “triangulation”, every vertex will have finite degree.

Of course, I cannot “show” you an abstract complex, but **Ⓢ Objects Ⓢ** begins with a random triangulation, one having no *a priori* connection with any circle packing — it is a simple graph you might have sketched by hand and then entered into some drawing program.

• **Circle Packing,  $P$ :** A circle packing for  $K$  is a configuration  $P = \{c_v\}_{v \in K}$  of circles with these properties:

- (1) There is a circle  $c_v$  associate with every vertex  $v$  of  $K$ .
- (2) If vertices  $v$  and  $u$  are neighbors (this is denoted  $v \sim u$  and means that  $\langle v, u \rangle$  is an edge of  $K$ ), then the circles  $c_v$  and  $c_u$  are (externally) tangent to one another.
- (3) if  $\langle v, u, w \rangle$  is an oriented face of  $K$ , then the circles  $c_v, c_u, c_w$  form a triple of mutually tangent circles which is positively (i.e., counterclockwise) ordered.

You will observe in **Ⓢ Objects Ⓢ** that the circles are indexed sequentially starting at 1. There is an tool there that displays the **flower** of a selected vertex; namely, the circle itself and the chain of neighboring **petal** circles. The script displays other distinctions: **Interior** circles are those having closed flowers (their petals wrap completely around them), while the **boundary** circles are those on the edge, with flowers that don’t close up. Likewise, the boundary/interior edges and faces are illustrated. A circle packing  $P$  is said to be **univalent** if its circles have mutually disjoint interiors.

• **Carrier,  $\text{carr}(P)$ :** The carrier of  $P$  refers to the union of geometry triangles formed by connecting the centers of the triples of circles in  $P$ . In the script, the carrier is displayed as the “complex”.  $P$  is **locally univalent** if for every interior vertex  $v$ , the faces in  $\text{carr}(P)$  corresponding to the flower of  $v$  have mutually disjoint interiors.

• **Label,  $R$ :** A label for  $K$  is a vector  $R = \{r_v\}_{v \in K}$  of putative radii, one for each vertex of  $K$ . As you might imagine, it is the circle radii that are of central importance in circle packing. These are maintained in *CirclePack* and can be found

using the “Info” button. (Note that radii and center data depend on which of the three geometries you are working in.)

If you try to create a circle packing “by hand”, say manipulating circle objects in some graphics program, you quickly learn how tightly choreographed their radii must be. The computational heart of circle packing involves the computation of compatible radii; the neutral term “label” is used in place of “radii”, because the algorithms typically start with values which do NOT fit together and then apply some iterative adjustments to approach a solution.

• **Angle Sums,  $\theta_R$ :** Given a complex  $K$  and an associated label  $R$ , how might you determine if circles with those radii would actually fit together in the pattern prescribed by  $K$ ? The key lies with angle sums. Consider an interior vertex  $v$  and let  $\{v_1, v_2, \dots, v_k\}$  be a list, in counterclockwise order, of its neighbors. Since the list is closed, it is convenient in computations to write  $v_{k+1} = v_1$ .

The faces containing  $v$  are  $\langle v, v_j, v_{j+1} \rangle$ ,  $j = 1, 2, \dots, k$ . Suppose  $r, r_j, r_{j+1}$  are the associated labels taken from  $R$ . For each such face we could place circles  $c_v, c_j, c_{j+1}$  in the appropriate geometry to form a triple. The triangle formed by their centers would have edge lengths which are the sums of radii, and we could appeal to the Law of Cosines to compute the angle in this triangle at  $v$ . In euclidean geometry, for instance, that angle would be

$$\alpha(r; r_j, r_{j+1}) = \arccos \left[ \frac{(r + r_j)^2 + (r + r_{j+1})^2 - (r_j + r_{j+1})^2}{2(r + r_j)(r + r_{j+1})} \right].$$

The sum of these angles at  $v$  in all the faces containing  $v$  is known as the **angle sum**:

$$\theta_R(v) = \sum_{\langle v, v_j, v_{j+1} \rangle} \alpha(r; r_j, r_{j+1}).$$

The neighbors of  $v$  will wrap precisely around  $v$  and only if  $\theta_R(v)$  is an integral multiple of  $2\pi$ . This is illustrated concretely in [Ⓢ Objects Ⓢ](#), but note that one does not need to actually *place* any circles to compute an angle sum, as computations are done entirely with labels.

• **Packing Label:** A label  $R$  for  $K$  is known as a packing label if  $\theta_R(v)$  is an integral multiple of  $2\pi$  for every interior vertex  $v$ . If  $P$  is a circle packing for  $K$  (in one of our geometries), then the list  $R$  of its radii will necessarily be a packing label. The converse is true for simply connected complexes  $K$ : if  $R$  is a packing label for  $K$ , then one can lay out circles, using  $R$  for radii, to form a circle packing  $P$  for  $K$ .

*CirclePack* is a platform for creating, manipulating, analyzing, and displaying circle packings. Its computational core lies in computing packing labels, and although the “packing” algorithms themselves are interesting and revealing mathematics, they are

not our target here. We will instead treat *CirclePack* as a blackbox computational engine so that we can concentrate on the underlying geometry. (Recall that there are several levels of usage for scripts; the more skill you develop, the more you'll learn from experiments: • click “NEXT” to simply watch the action; • press “Open Script” to follow commands and associated descriptions; • modify the given script commands; • add new commands to a script; and • create new scripts from scratch.)

### 6.3. Circle Packing Manipulations

We base our introduction to circle packing on a sequence of scripts revealing both the objects of study and the principal manipulations of them available within *CirclePack*.

1. **Patterns** concentrates on combinatorics. Starting with an empty canvas, a “seed 7” command creates the simplest type of packing, namely a single **flower** — an interior circle surrounded by, in this case, seven petal circles. As you click through the script the first time, disregard the occasional repackings that are needed. The emphasis here is on building and adjusting the combinatorics. You will see how to add generations of new neighbors, how to add/delete individual boundary circles, how to cut open or zip up a string of boundary circles, how to “double” a packing across a boundary segment, how to adjoin two packings, how to cookie out one packing from a larger one, and how the process of “hex” refinement works. *Try it out!*

2. **Geometries** demonstrates that *CirclePack* operates in any of the three standard geometries: euclidean, represented by the familiar complex plane  $\mathbb{C}$ ; hyperbolic, represented by the unit disc  $\mathbb{D} \subset \mathbb{C}$  with the Poincaré metric (density  $|ds| = |dz|/(1 - |z|^2)$ ); and spherical, represented by the **Riemann sphere**  $\overline{\mathbb{C}}$ , namely, the unit sphere  $\{(x, y, z) : x^2 + y^2 + z^2 = 1\}$  in  $\mathbb{R}^3$ . The geometries are said to have zero, negative, or positive curvature, respectively.

The three geometries are nicely nested,  $\mathbb{D} \subset \mathbb{C} \subset \overline{\mathbb{C}}$ , with this last inclusion *via* “stereographic” projection. Moreover, circles in one geometry are circles in the others (though centers and radii depending on the geometry). The script illustrates changes in geometry, displays circles and faces in each, and shows the effects of automorphisms. The plane is the most comfortable setting for most of us. The sphere is also quite familiar — after all, we live on a sphere — but in circle packing it is by far the most challenging. Hyperbolic geometry is perhaps least familiar, but also in many ways the richest. One detail to note in hyperbolic geometry is that we can have circles of “infinite” radius. Namely, a **horocycle** in  $\mathbb{D}$  is a circle which is internally tangent to the unit circle. It can be treated in a natural way as a circle of infinite hyperbolic radius whose center is at the point of tangency with the unit circle. *Try it out!*

3. **Layout** may help you understand the repacking/layout process; though we treat both these as blackbox operations in general, one should see at least a little about

them. In the script, an embedding of an abstract complex  $K$  is displayed (without any reference to circles). A randomly chosen label  $R$  is generated ( $R$  will be different each time you run the script) and circles are drawn using the labels as radii. The “**repack**” command adjusts the labels iteratively based on their current values and the neighbor relationships encoded in  $K$ . (One option for repacking lets you see the intermediate radii in stages.) Boundary radii (red circles) are fixed, interior radii (blue circles) change to meet the packing condition, but in the display the centers are fixed. Only when the radii are “right” (i.e., comprise a “packing” label), are we ready for the “**layout**” command to move the circles to consistent locations. First, the two circles for some edge are laid out to be tangent to one another; then the rest of the circles can be placed unambiguously in succession. The original combinatorial pattern is preserved, but now the locations for its vertices are determined by the circle centers. We have a packing! ***Try it out!***

4. **Ⓢ**BdryControl**Ⓢ** illustrates manipulations critical to our later work: solutions of boundary value problems for radii and boundary angle sums for a small euclidean packing. The first section of the script emphasizes boundary angle sum manipulations. With the cursor and key presses you can prescribe boundary angle sums of  $\pi$ ,  $\pi/2$ , or  $3\pi/2$  at various boundary vertices; other boundary vertices have unrestrained angle sums which necessarily adjust during repacking. A key observation is that while you control some things, the packing pushes back on others — there is a mixture of flexibility and rigidity. Here are some things to think about: Can you prescribe boundary angle sums arbitrarily? Can you generate some problem situations? What happens if you prescribe all but one of the boundary angle sums? In this case, can you compute that remaining one? Can you prescribe angles to get the region to overlap itself? Some of these issues will come up in Exercise 6.5.

It is interesting to contemplate the boundary situation in the other geometries as well. The script allows you to investigate quadrilaterals in hyperbolic geometry; the same controls work on boundary angle sums, for instance, but you will see that the constraints of the geometry are different. ***Try it out!*** On the sphere, there is little we or *CirclePack* can do. With no packing algorithm, we can’t generate or manipulate boundary values. (If you study spherical geometry, you can show that there are boundary constraints. For example, a spherical quadrilateral (if you could build it) would have boundary angle sums exceeding  $2\pi$ .)

5. **Ⓢ**Type67**Ⓢ** investigates connections between combinatorics and geometry using constant-degree packings, those in which every circle has the same degree (the same number of neighbors). The takehome message involves these associations: degree 6 circles, flat (zero curvature); degree 5 or less, positive curvature; degree 7 or more, negative curvature. ***Try it out!***

6. Maximal packings are particularly important to our work and are investigated in three scripts. A **maximal packing** for a complex  $K$  is a univalent circle packing whose carrier “fills” the underlying geometric space.

The theorem at the very foundation of circle packing is the Koebe-Andreev-Thurston Theorem. (We say  $K$  is a sphere when we mean that it is a *triangulation of a topological sphere*.)

**THEOREM 6.2 (KAT).** Let  $K$  be a sphere. Then there is univalent circle packing  $P_K$  for  $K$  living on the Riemann sphere,  $\overline{\mathbb{C}}$ . Moreover,  $P_K$  is unique up to Möbius transformations of the sphere (and inversions).

Note that we assume our complexes  $K$  and associated packings  $P$  have the same orientation, so we will not allow inversions. Also, we say a circle packing for  $K$  is **essentially unique** if it is unique up to conformal automorphisms of the underlying space. In our terminology, then, the KAT Theorem states: *Every complex  $K$  which is a sphere has a maximal packing  $P_K$  of the Riemann sphere, and  $P_K$  is essentially unique.*

If  $K$  is simply connected, but does not triangulate a sphere, then it necessarily triangulates a topological disc. The central case, both in theory and computations, is when  $K$  triangulates a *closed* topological disc (we will say that  $K$  is a closed disc.) Specifically, the conditions are that  $K$  be finite, simply connected, and have one connected boundary component (boundary edges and vertices).

**THEOREM 6.3.** If  $K$  is a closed disc, then there is a univalent packing  $P_K$  for  $K$  in the hyperbolic plane having the property that all its boundary circles are horocycles. Moreover,  $P_K$  is unique up to Möbius transformations of  $\mathbb{D}$ .

📄 **MaxPackHyp** 📄 explores this theorem. Since this is key to much of our later work, I recommend that you go through this script first. You will see the practical computations in action, but you might also glean the intuition behind the formal proof.

📄 **MaxPackSph** 📄 should be run next. It illustrates that the KAT Theorem and Theorem 6.3 are equivalent, each implies the other. Since packing computations cannot (yet) be carried out in spherical geometry, the hyperbolic computations are a prerequisite.

That leaves 📄 **MaxPackEucl** 📄. If  $K$  is simply connected but neither a sphere nor a closed disc, then it is necessarily infinite (possibly having boundary). This third script investigates some infinite maximal packings. These are only suggestive, of course, since one cannot actually compute with or display infinitely many circles. However, the intuition is theoretically sound.

So, the suggested order is: 📄 **MaxPackHyp** 📄, 📄 **MaxPackSph** 📄, 📄 **MaxPackEucl** 📄. **Try it out!** Note that in practice, all simply connected complexes  $K$  are finite; *CirclePack* has a single command ‘`max_pack`’ (or the dropable icon with the big maize ‘M’) which will automatically compute a normalized maximal packing  $P_K$  in the sphere

or the unit disc, as appropriate. (Maximal packings exist for non-simply connected complexes, too. We will see triangulations of the torus in a later script.)

7. **Ⓢ**Branching**Ⓢ** illustrates one further source of control beyond the boundary conditions we have studied so far: namely, branching. A circle  $c_v$  is **branched** if it is an interior circle whose neighbors wrap more than once around it. For “simple” branching, the angle sum at  $v$  is  $4\pi$  and the neighbors wrap twice around it. In general, if the angle sum is  $2\pi n$ , the neighbors wrap  $n$  times around and the circle is said to have **branch order**  $n - 1$ . Investigate the local geometry at branch circles in this script. *Try it out!*

8. **Ⓢ**InputOutput**Ⓢ** gives you a tour of *CirclePack* input/output operations. After you work with *CirclePack* for a while, you may want to read/write packings, save/load scripts, create PostScript images and \*.jpg screendumps, etc. *Try it out!*

#### 6.4. Discrete Function Theory

With experience from the previous scripts, the reader should be ready for the investigation of discrete analytic function theory. To see how *CirclePack* displays functions, repeat **Ⓢ**Cookie**Ⓢ**, which creates the discrete conformal mapping of Figure 6.1.

The script begins with a curve defining a Jordan region  $\Omega$ . A regular hexagonal packing  $H_\epsilon$  (i.e., all circles the same radius  $\epsilon$ ) overlays  $\Omega$  in the canvas on the right. Using  $\Omega$  like a cookie cutter, a packing  $P$  of  $\Omega$  is cut from  $H_\epsilon$ . The associated complex  $K$  has a maximal packing  $Q = P_K$  in  $\mathbb{D}$ , which *CirclePack* computes and displays in the canvas on the left. We now have, more or less, the situation as pictured in Figure 6.1. This is the typical way we will represent our functions  $f : Q \rightarrow P$  in *CirclePack*: “domain” on the left, “range” on the right. Here’s the official definition.

**DEFINITION 6.4.** A **discrete analytic function** is a map  $f : Q \rightarrow P$  between circle packings  $Q$  and  $P$  which preserves tangency and orientation.

As a practical matter, our discrete analytic functions will always have domain and range packings that share a common complex  $K$  (though, of course,  $K$  may vary from one example to another). If  $v$  is a vertex of  $K$  with circle  $C_v$  in  $Q$ , then  $c_v = f(C_v)$  is the circle for  $v$  in  $P$ . If  $\langle v_1, v_2, v_3 \rangle$  is a positively oriented face in  $K$ , then  $\langle C_{v_1}, C_{v_2}, C_{v_3} \rangle$  form a mutually tangent, positively oriented triple of circles in  $Q$ , and  $\langle c_{v_1}, c_{v_2}, c_{v_3} \rangle$  form a mutually tangent, positively oriented triple of circles in  $P$ .

Note that because of Theorem 6.3, we have already generated data for several discrete analytic functions. In particular, every packing,  $P$ , having a simply connected complex  $K$  is the range of a discrete analytic function: just build its associated maximal packing  $P_K$ , and *viola*,  $f : P_K \rightarrow P$  is a discrete analytic function. You might well ask “What is the big deal about functions?” The function is not about the range (alone); it’s about how the domain changes into the range — it’s about the *mapping*. Which circles grow, which shrink? How does the boundary behave? Do the circles remain

disjoint, or do some come to overlap? How does the geometry change under  $f$ ? Which properties are preserved? which ones change?

Subsequent scripts will teach you how *CirclePack* can aid in generating mappings and investigating their properties. In the domain/range mode, as in Figure 6.1, you can click the left mouse button on a circle in one packing and both it and its counterpart in the other packing will be highlighted. Check a few pairs of tangent circles in  $Q$  to see that their counterparts are indeed tangent in  $P$ . Also observe that the circle centered at 0 in  $Q$  corresponds to the circle of  $P$  which is also at 0: this is what we will mean when we say  $f(0) = 0$ . Clicking the middle mouse button will highlight corresponding faces of  $Q$  and  $P$ . *CirclePack* provides various buttons for refreshing, zooming, etc.; I'll leave it to you to experiment with the interface. Your goal is to be able to probe the packings to discover properties of the mapping  $f$ .

**6.4.1. Conformal Maps.** The “cookie” method demonstrated in [S](#)Cookie[S](#) is a very practical way to create **discrete conformal** mappings, so called because they parallel the conformal (i.e., one-to-one analytic) mappings of the classical theory.

There is a second approach which is modeled on the Schwarz-Christoffel methods seen in Section 5.2. The aim is to find a conformal map of the upper half plane to the region bounded by a given polygonal path  $\Gamma$ .

The classical formula is (67) (and there is a parallel version when the domain is the disc). Having such a formula for  $f$  suggests a straightforward computation, but that is not the case. The  $\beta_j$  are known in advance from the corners of  $\Gamma$ , but the “prevertices”, the  $x_j$  in the formula, are the points mapping to those corners, and those are NOT known. WHERE does the turning occur? That is the real challenge with Schwarz-Christoffel. In the discrete version, one confronts this issue head on. Figure 6.6 illustrates a polygonal packing: the boundary edges of the carrier define a polygonal curve  $\Gamma$ . The problem, of course, is that you are given  $\Gamma$  and a complex  $K$  first. See how it goes in the next exercise.

**EXERCISE 6.5.** First a little euclidean geometry. Given a polygonal path  $\Gamma$ , its **turning angles** reside at the corners: when you travel in the positive (counterclockwise) direction along  $\Gamma$  and you come to a corner, the angle you turn through to start the next edge is by convention taken as positive if you turn left, negative if you turn right. Compute the algebraic total of all the turning angles for simple a closed polygonal path.

In Schwarz-Christoffel notation, the  $j$ th turning angle is  $\beta_j\pi$ . We can obtain the boundary angle sum  $\theta_j$  at the corner by observing (see Figure 6.6) that  $\theta_j = \pi - \beta_j\pi$ . Conditions on the total of turning angles therefore give conditions on the total of all boundary angle sums: this will constrain your experiments, so you should work this out before jumping into the script.

[S](#)DiscreteSC[S](#) starts with a worked out example for  $\Gamma$  a square. It then challenges you with a slightly more complicated  $\Gamma$ . You will have to open and read the

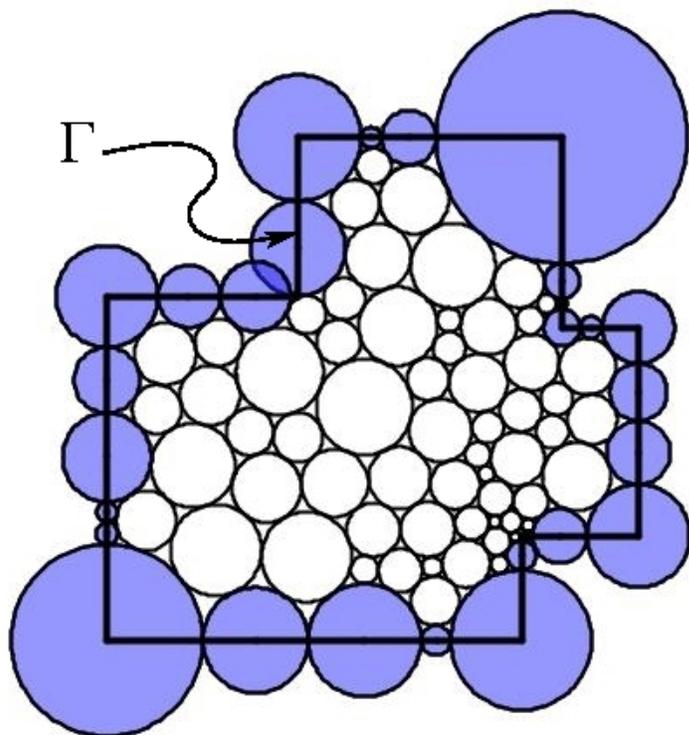


FIGURE 6.6. Packing of a polygonal region

script to see how the machinery has been set up and how to manipulate the packing with mouse and key presses. /dtn.

**EXERCISE 6.6.** The Riemann Mapping Theorem says that every simply connected proper subdomain  $\Omega$  of the plane is conformally equivalent to a particular “model” simply connected domain, namely, the open unit disc. This has been extended to certain *multiply* connected regions  $\Omega$ . For instance, if  $\Omega$  is a connected domain in the plane bounded by some finite number  $n$  of disjoint Jordan curves, then it is called  *$n$ -connected*. (A “Jordan” curve is a closed curve without self-intersections.) In this case, there is a conformal mapping between  $\Omega$  and a standard model multiply connected domain. There are various possible “models”, but of particular recent interest are so-called “circle domains”. A circle domain  $C$  in the plane is one bounded by a finite number of circles. Despite the several numerical methods for computing the maps for simply connected domains — such as Schwarz-Christoffel — the multiply connected cases are much tougher. The discrete version is quite elementary, however. See [S CircleDomains](#) [S](#). *Try it out!*

**6.4.2. Basic Function Theory.** Let’s look at scripts in which we construct and study additional types of functions. We start with maps of the unit disc into itself

which are important in classical function theory and especially convenient for us since we can work in hyperbolic geometry. **ⓈSchwarzⓈ** starts with a somewhat random, rather small range packing  $P$  in  $\mathbb{D}$ . Incrementally increasing the hyperbolic boundary radii to infinity (in practice, 5 is sufficiently large!) and  $P$  morphs into its maximal packing  $P_K$ . The resulting map  $f : P_K \rightarrow P$  is a discrete analytic function of  $\mathbb{D}$  into itself. It is convenient to introduce some notation here.

**DEFINITION 6.7.** Let  $f : Q \rightarrow P$  be a discrete analytic function, where  $Q$  and  $P$  are packings with common complex  $K$ . The associated **ratio** function  $f^\# : K \rightarrow \mathbb{R}$  is defined by

$$(99) \quad f^\#(v) = \frac{\text{radius}(f(C_v))}{\text{radius}(C_v)}, v \in K,$$

where  $C_v$  is the circle for  $v$  in  $Q$  and  $f(C_v)$  is the corresponding circle in  $P$ .

$f^\#(v)$  is the “stretch factor” at the  $v$ , the amount that the associated circle is stretched in going from  $Q$  to  $P$ . With this for notation, we can illustrate our first piece of **discrete function theory**, the Discrete Schwarz Lemma.

**THEOREM 6.8 (Discrete Schwarz Lemma).** Let  $f : P_K \rightarrow P$  be a discrete analytic self-map of the unit disc with  $f(0) = 0$ . Then  $f^\#(0) \leq 1$ . Moreover, equality holds if and only if  $f$  is a rotation; that is,  $P = \lambda P_K$  for some complex  $\lambda$  with  $|\lambda| = 1$ .

**EXERCISE 6.9.** Run through **ⓈSchwarzⓈ** again, but more carefully. **Try it out!** Can you see the Discrete Schwarz Lemma in action? The ratio function plays the role in the discrete theory that the absolute value of the derivative,  $|F'|$ , plays in the classical theory, so the Discrete Schwarz Lemma precisely parallels its classical counterpart, even up to the equality statement.

**EXERCISE 6.10.** There is a classical extension of the Schwarz Lemma known as the Dieudonné-Schwarz Lemma. If  $F : \mathbb{D} \rightarrow \mathbb{D}$  is analytic and  $F(0) = 0$ , then by the Schwarz Lemma,  $|F'(0)| \leq 1$ . According to this extension, there is in fact a certain universal constant  $\mathcal{C}$ ,  $0 < \mathcal{C} < 1$  (i.e., independent of  $F$ ) so that  $|F'(z)| \leq 1$  whenever  $|z| \leq \mathcal{C}$ . **ⓈDieudonneⓈ** sets up experiments so you can try to estimate this constant: You manipulate a range packing  $P$  in  $\mathbb{D}$ ; with color coding, *CirclePack* shows you the circles in  $P_K$  whose (euclidean) radii become smaller in  $P$ . How close can these circles come to the origin in  $P_K$ ? I can give you a bit of help:  $\mathcal{C}$  is larger than  $1/2$  but quite a bit smaller than 1. **Try it out!** (The correct value can be found in the text at the end of the script.)

The Discrete Schwarz Lemma is an example of a “monotonicity” result — if you decrease the boundary radii of  $Q$  to those of  $P$ , then the radius of the circle at the origin also goes down. Details are given in [6], but the basic ideas can be seen in the following Exercise:

EXERCISE 6.11. Consider a mutually tangent triple of circles in euclidean geometry as illustrated in Figure 6.7. Using the euclidean Law of Cosines, show that if the radius of circle  $C_v$  is increased, then the angle  $\alpha$  at  $C_v$  will decrease while the angles  $\beta, \gamma$  at the other two circles will increase.

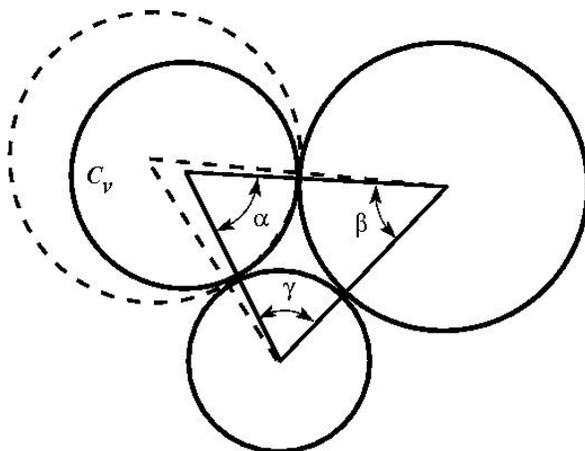


FIGURE 6.7. Monotonicity in euclidean triples

EXERCISE 6.12. Using the previous fact, prove the following:

THEOREM 6.13 (Discrete Maximum Principle). Let  $f : Q \rightarrow P$  be a discrete analytic function between euclidean packings  $Q$  and  $P$ , and suppose  $Q$  is locally univalent. Then the ratio function  $f^\#(v)$  attains its maximum at some boundary vertex  $v$ .

Analogous results hold in hyperbolic geometry and are central to the computation of packing radii. The failure of this monotonicity in spherical geometry may explain the lack of a packing algorithm there.

EXERCISE 6.14. Show with a computation in spherical geometry (using the Spherical Law of Cosines) that the monotonicity of 6.11 can fail in spherical geometry. What appears to be the trigger for such failure? **Try it out!**

The analogy between  $f^\#$  and the classical  $|F'|$  is not perfect. In particular,  $|F'|$  is zero at branch points of  $F$ , whereas  $f^\#$  will never be zero. Nevertheless, the fact that discrete functions correctly realize the geometry underlying branching can be seen in the following Exploration.

EXPLORATION 6.15. An important class of classical analytic self-maps of the disc, the **finite Blaschke products**, was introduced in Chapter 4. With the explicit product formula shown in (4.6), a Blaschke product  $B : \mathbb{D} \rightarrow \mathbb{D}$  has properties subject to direct computation. For instance, each factor in the formula accounts for one zero

of  $B$ , and  $B$  extends to a continuous map on the unit circle,  $\partial\mathbb{D}$ , mapping  $\partial\mathbb{D}$   $n$  times around itself, where  $n$  is the number of factors in  $B$ .

However, there is a more geometric characterization of finite Blaschke products: they are precisely the **proper** analytic maps  $f$  of  $\mathbb{D}$  onto itself. “Proper” means that whenever  $\{z_j\}$  is a sequence of points in  $\mathbb{D}$  converging to the unit circle,  $|z_j| \rightarrow 1$ , then the images,  $\{f(z_j)\}$ , also converge to the unit circle,  $|f(z_j)| \rightarrow 1$ . By orientation,  $B$  wraps the boundary around itself some number  $n$  times, and from this the argument principle implies that  $B$  will have  $n - 1$  branch points inside the disc. It is these geometric properties which lead you to **discrete** finite Blaschke products, as you will see in **Ⓢ Blaschke Ⓢ**. **Try it out!** You also get a closeup look at branch points — a chance to demystify the local geometric behavior.

Analytic functions present this fascinating mixture of rigidity within flexibility — I’m convinced that this explains why they have attracted such attention for two centuries now. You get to manipulate some features, but then you have to live with the consequences that conformal rigidity brings. This mixture is rather difficult to get a handle on, but there are two particular notions mathematicians have invented that give some insight: **harmonic measure** and **extremal length**.

**6.4.3. Harmonic Measure.** Let  $\Omega$  be a Jordan region  $\Omega$  in the plane (that is, one bounded by a simple closed curve) and  $A$  a subset of its boundary,  $A \subset \partial\Omega$ . There is a unique harmonic function  $h$  defined on  $\Omega$  which goes to 1 as  $z$  approaches a point of  $A$  and goes to 0 as  $z$  approaches a point of  $\partial\Omega \setminus A$ . (We necessarily suppress finer details about measurability and continuity here.)  $h$  is known as the **harmonic measure function** and written as  $h(z) = \omega(z, A, \Omega, \cdot)$  (Notations like  $\omega(z, A, \Omega, m)$  may be new to you: the  $\omega$  denotes a function whose value depends on three things, the point  $z \in \Omega$ , the boundary arc  $A$  in  $\partial\Omega$ , and the region  $\Omega$  itself. Hold any of these fixed, and it’s a function of the others.)

We also say that  $\omega(z, A, \Omega, \cdot)$  is the **harmonic measure** of  $A$  at  $z$  with respect to  $\Omega$ . This is called a “measure” because if you keep  $z$  and  $\Omega$  fixed, then it is a probability measure on the variable sets  $A$ . Observe that  $\omega(z, A, \Omega, \cdot) \in [0, 1]$ , with  $\omega(z, \emptyset, \Omega, \cdot) = 0$ ,  $\omega(z, \partial\Omega, \Omega, \cdot) = 1$ , while if  $A_1, A_2$  are disjoint in  $\partial\Omega$ , then  $\omega(z, A_1 \cup A_2, \Omega, \cdot) = \omega(z, A_1, \Omega, \cdot) + \omega(z, A_2, \Omega, \cdot)$ .

There are two especially nice properties of harmonic measure in the classical setting:

(1) It is easy to compute in certain simple situations: In particular, if  $\Omega$  is the unit disc  $\mathbb{D}$  and  $z$  is taken as the origin, then  $\omega(0, A, \mathbb{D}, \cdot) = \text{arclength}(A)/(2\pi)$ , the proportion of the unit circle that  $A$  occupies as seen from the origin.

(2) It is a **conformal invariant**; that is, it is invariant under conformal maps. Put more explicitly, suppose  $F : \Omega_1 \rightarrow \Omega_2$  is a conformal map (i.e., a conformal homeomorphism) between two domains. Under quite weak conditions, it is known that  $F$  can be extended to map the boundary onto the boundary,  $F : \partial\Omega_1 \rightarrow \partial\Omega_2$ .

Given  $A_1 \subset \partial\Omega_1$ , define  $A_2$  to be its image  $A_2 = F(A_1)$  in  $\partial\Omega_2$ , and suppose  $z_1 \in \Omega_1$  and  $z_2 = F(z_1) \in \Omega_2$ . Then  $\omega(z_1, A_1, \Omega_1, \omega(z_2, A_2, \Omega_2))$ .

If you put these two properties together with the Riemann Mapping Theorem, one can, in theory, find the harmonic measure function for any simply connected region. How might this be useful? Here's but one of numerous applications. Suppose a homogeneous, thin metal plate is cutout in the shape of  $\Omega$ . Suppose a segment  $A$  of the boundary is held at 100 degrees (centigrade), while the remainder of the boundary is held at 0 degrees. After the plate has had time to reach thermal equilibrium (so temperatures are no longer changing), what is the temperature at a point  $z$  in the interior of the plate? It is evidently somewhere between 0 and 100, it would approach 100 as  $z$  comes closer to  $A$ , and 0 as  $z$  gets closer to other parts of the boundary. But what is the temperature precisely at  $z$ ? The answer is precisely  $100 \cdot \omega(z, A, \Omega, \cdot)$ .

In  $\textcircled{S}$ HarmMeasure $\textcircled{S}$  we investigate the discrete analogy. I can share the definition now, since it is motivated directly by property (1). Suppose  $K$  triangulates a closed topological disc,  $A$  is a chain of edges in  $\partial K$ , and  $v \in K$  is an interior vertex of  $K$ . Let  $P_K$  be the maximal packing for  $K$  in  $\mathbb{D}$  which has the circle for  $v$  centered at the origin. Each edge  $e \in A$  corresponds to an arc  $\alpha_e \subset \partial\mathbb{D}$  between the centers of the two horocycles forming  $e$ .

DEFINITION 6.16 (Discrete Harmonic Measure). Given  $K, A$ , and  $v$  as described, the **discrete harmonic measure** of  $A$  at  $v$  with respect to  $K$  is

$$\omega(v, A, K, \cdot) = \sum_{e \in A} \text{arclength}(\alpha_e) / (2\pi).$$

So, in fact, discrete harmonic measure is defined directly on  $K$ . The invariance of property (2) above in the discrete setting is merely due to the definition. In the script, however, you will see its geometric nature exhibited, much as you would see the temperature distributed about a plate in the application noted above. A couple of experiments are quite convincing that the same geometry is at work in both the discrete and the classical settings: we discuss harmonic measure and exit probabilities, comparing the classical and discrete versions for a region, and we illustrate the **extension of domain principle**.

**6.4.4. Extremal Length.** Extremal length is a slightly more nuanced measure of conformal “shape”. Start with the Jordan region  $\Omega$  again, but this time suppose we are given disjoint closed subarcs  $A$  and  $B$  of its boundary; the triple  $\mathcal{Q} = \{\Omega; A, B\}$  is described as a “conformal quadrilateral”.

A classical theorem states that there exists a conformal mapping  $F : \Omega \rightarrow R$ , where  $R$  is a euclidean rectangle, so that  $F$  extends continuously to the boundary and so that the images  $F(A), F(B)$  are opposite ends of  $R$ .  $F$  is unique up to euclidean similarities. (You might sense a similarity to a Schwarz-Christoffel situation, as investigated earlier, but this is quite different. The four boundary points mapping to corners are specified, but the shape of the image rectangle is not set in advance.)

Figure 6.8 illustrates the analogous discrete result you will see in **ExtLength**. By a rigid motion, image rectangles can be positioned as shown here, with the images of  $A$  and  $B$  on the left and right, respectively. The **aspect ratio** of  $R$  will be defined as its width over its height,  $\text{Aspect}(R) = W/H$ .

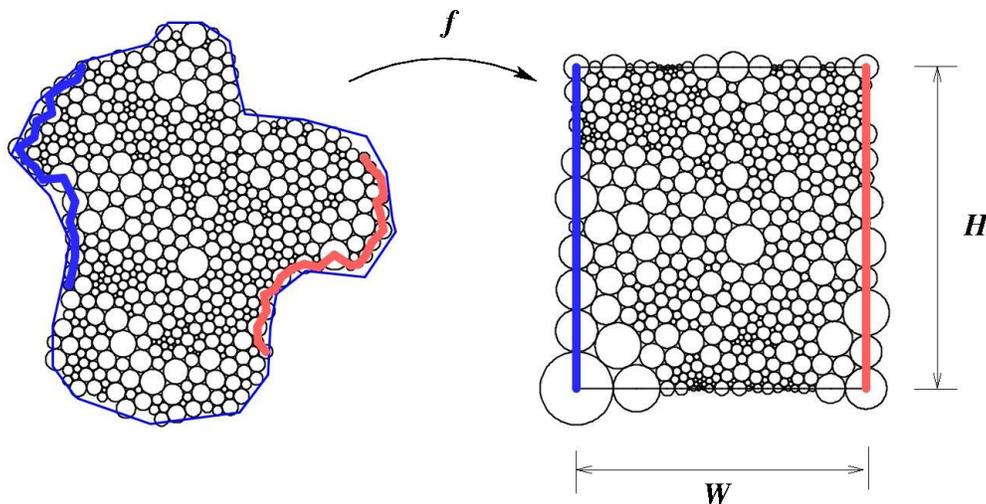


FIGURE 6.8. Packing for extremal length

Continuing our discussion of the classical setting,  $\text{Aspect}(R)$  is known as the **extremal distance** from  $A$  to  $B$ , relative to  $\Omega$ , or the **extremal length** of the quadrilateral  $\mathcal{Q}$ , denoted  $\text{EL}(\mathcal{Q})$ . As with harmonic measure, there are two key properties:

(i). Extremal length is easy to compute in certain simple situations: For instance, if  $\Omega$  is a euclidean rectangle width  $W$  and height  $H$ , and  $A$  and  $B$  are its two ends, then clearly  $\text{EL}(\{\Omega, A, B\}) = W/H$ .

(ii). Extremal length is a **conformal invariant**; that is, if  $\mathcal{Q}_1 = \{\Omega_1; A_1, B_1\}$  and  $\mathcal{Q}_2 = \{\Omega_2; A_2, B_2\}$  are conformal quadrilaterals and  $F : \mathcal{Q}_1 \rightarrow \mathcal{Q}_2$  is a conformal mapping such that  $F(A_1) = A_2$  and  $F(B_1) = B_2$ , then  $\text{EL}(\mathcal{Q}_1) = \text{EL}(\mathcal{Q}_2)$

Also as with harmonic measure, there are physical interpretations. Suppose  $\Omega$  is a uniform conducting metal plate in the plane, the boundary segment  $A$  is attached to a 1 volt power supply, the boundary segment  $B$  is grounded, and the remaining segments of the boundary are insulated. Then the current flow between  $A$  and  $B$  in amperes will be proportional to  $\text{EL}(\{\Omega, A, B\})$ , the constant depending on the properties of the metal.

Discrete analytic mappings to rectangles having designated boundary vertices as corners are quite easy to compute by methods you have already seen in **RePack**:

you simply specify that all boundary vertices have angle sum  $\pi$  except for the four which are to be the “corners”, and these you set to  $\pi/2$ . `ExtLength` illustrates the construction of Figure 6.8 and explains how *CirclePack* reports the extremal length when you have a rectangle. You are asked to modify the constructions to answer a question about current flow. You might then test your intuition with an extremal length analogue of the extension of domain principle we saw with harmonic functions.

One last script for this section, `MiscFtnTheory`, illustrates the range of miscellaneous behaviors you might encounter with discrete analytic functions. These can provide concrete insights you might carry back to the classical setting.

## 6.5. Function Construction

We have “hands-on” control of our discrete functions, but this is not like the formula-driven and algebraic control available in the classical setting. In the discrete world, we have limited ability to add or multiply functions, and essentially no way to consider compositions of the type studied in Chapter ???. To build functions we require something geometric to hook into. We have seen several examples already: to get maximal packings in  $\mathbb{D}$ , we send boundary radii to infinity; to build euclidean polygons, we control their boundary angle sums; to create Blaschke products, we designate branch circles. The next sequence of scripts give additional examples, some involving techniques, others relying on special circumstances or outright tricks. There’s plenty of room for creative thinking in this.

**6.5.1. Discrete Exponentials.** `Doyle` challenges you to adjust parameters in a family of fascinating circle packings known as Doyle spirals. With patience you can build a range packing which clearly mimics the classical exponential function. ***Try it out!*** Read the script for guidance, but then experiment on your own. The answer is revealed at the end.

**6.5.2. Discrete Rational Functions.** A classical **rational** function is one of the form  $F(z) = P(z)/Q(z)$ , where  $P(z), Q(z)$  are complex polynomials (assumed to have no common factors). Projecting stereographically to the sphere in both domain and range,  $F$  may be considered as a map from the Riemann sphere to itself,  $F : \overline{\mathbb{C}} \rightarrow \overline{\mathbb{C}}$ , and that will be our view here. In particular, if  $P(z) = 0$ , then  $F(z) = 0$ , while if  $Q(z) = 0$ , then  $F(z) = \infty$ . The behavior at  $\infty$  in the domain depends on relative degrees: if  $\deg(P) > \deg(Q)$ , then  $F(\infty) = \infty$ , if  $\deg(P) < \deg(Q)$ , then  $F(\infty) = 0$ , and if  $\deg(P) = \deg(Q)$ ,  $F(\infty)$  is a non-zero complex number.

The sphere is compact, and  $F : \overline{\mathbb{C}} \rightarrow \overline{\mathbb{C}}$  is an open continuous map, so key properties of rational maps are purely topological:  $F$  has a constant **valence**,  $N > 0$ ; that is, for every point  $p \in \overline{\mathbb{C}}$ ,  $F^{-1}(p)$  has  $N$  points (counting multiplicities at branch points). The valence  $N$  satisfies  $N = \max\{\deg(P), \deg(Q)\}$ . Moreover, there are precisely  $2(N - 1)$  branch points. (This counts multiplicities at branch points, which

may include branching at infinity.) As you build examples you should watch for these properties directly.

Ⓢ **Rational1** Ⓢ (???? Poly) shows one technique for the construction of rational maps, since polynomials  $P$  are themselves rational maps (take  $Q \equiv 1$ ): the valence  $N$  is just the degree of  $P$ . The discrete construction in Ⓢ **Rational1** Ⓢ illustrates these “polynomial-like” rational maps. To get a taste for what should be “more typical” rational maps Ⓢ **Rational2** Ⓢ gives a particular example that was constructed using the special geometry of “Schwarz” triangles on the sphere. This is a beautiful example: it is the discrete analogue of  $F(z) = z^2(3z^5 - 1)/(z^5 + 3)$ , with valence 7 and 12 simple branch points. Lovely as it is, it just highlights a major capability that is missing in our discrete theory: we have no algorithm for computing packing labels directly in spherical geometry.

**6.5.3. Ratio Function Constructions.** Recall that if  $f : Q \rightarrow P$  is a discrete analytic function, then the associated local scaling, the analogue of  $|F'|$  for analytic functions, is given by its ratio function  $f^\#$ . (See Definition 6.7.) In particular, if we wish to build a discrete analogue  $f$  for some classical function  $F$ , we might begin from this differential level, mimicking  $|F'|$ .

Ⓢ **RatioBuild** Ⓢ builds a discrete analogue for the classical function  $F(z) = (1 + z)^2$ , which was illustrated in Chapter 4, Figure 4.2, using the applet *ComplexTool*.

Ratio function constructions also give an approach to entire functions. We illustrate in Ⓢ **Erf** Ⓢ for a function central to statistics, the **error function**  $erf(z)$  defined by

$$erf(z) = \frac{2}{\sqrt{\pi}} \int_0^z e^{-\zeta^2} d\zeta.$$

When  $z$  is restricted to the real axis,  $z = x$ , then the derivative of  $erf(x)$  is the Gaussian distribution, the famous “bell-shaped” curve. You may recall that this function has no closed form anti-derivative, which explains the integral representation for  $erf(z)$ .

The construction of the discrete error function illustrates several construction issues. Let me highlight a few to prepare you.

(1) First,  $F$ , as an entire function, is defined on the whole complex plane. Discrete entire functions would likewise be defined on infinite complexes, as the discrete exponential we defined earlier on the regular hexagonal packing. Experiments and computations, however, are restricted to finite complexes. Our approach relies on “exhaustion”: we mimic  $F$  using finite packings that take up successively larger, but finite, portions of  $\mathbb{C}$ .

(2)  $F$  has infinite valence: for every  $w \in \mathbb{C}$ ,  $F^{-1}\{w\}$  has infinitely many points. In fact, its “mapping” behavior is quite fascinating to watch. Early in the construction, it may appear that we are omitting two points,  $w_1, w_2$ , much as the exponential omits the image value 0. That, however, would contradict the Little Picard Theorem, which tells

us that a nonconstant entire function can omit at most one value in the complex plane. Watch more stages in the construction and see how our growing function manages, in fact, to cover everything. It turns out that though not omitted,  $w_1$  and  $w_2$  are what is known as **asymptotic values**, that is, there are paths in the domain whose image paths converge to  $w_1$  or  $w_2$ . Can you find such paths in the *CirclePack* experiments?

(3) Does the sequence of packings of this construction ultimately converge to an infinite circle packing? If so, it would answer in the negative one of the earliest questions in circle packing:

QUESTION 1. Are the regular hexagonal packings and the Doyle spirals the only locally univalent hexagonal circle packings in the plane?

But is there, in fact, an infinite packing behind this construction? Each stage of construction is finite, so this is a convergence issue, and it remains unresolved. The script has both positive and negative indications.

**6.5.4. Harmonic Mappings.** Recall from Section 4.31 that harmonic mappings  $H$  on the disc can be written in the form  $F(z) = H(z) + \overline{G}(z)$  where  $H, G$  are analytic in  $\mathbb{D}$ . While the “addition” here is not a geometric feature that can be carried directly by our circles, there is a discrete fact that captures the local behavior which we will discuss in a moment. The upshot is that if one just jumps in with discrete functions, the fundamental global phenomena appear to persist. In this section we start with the mechanics, see some examples parallel to ones in Chapter 4, then set up some lines of open experimentation for you.

Take a quick run through [HGBasics](#). **Try it out!** The way *CirclePack* operates,  $h$  and  $g$  will be discrete analytic functions on the disc, so, within each example they will share a complex  $K$  and have its maximal packing  $P_K$  as their common domain. In fact, there will be no need to view  $P_K$ , but in [HGIntro](#)  $h$  is taken as the identity function, so  $P_K$  itself appears in p0 as the range of  $h$ . The image of  $g$  appears in p1. On execution of 'h\_g\_bar', the sum  $h + \bar{g}$  is displayed in p2. *CirclePack* requires that the range packings for  $h$  and  $g$  both be euclidean, and a key feature in [HGIntro](#) is that their boundary radii are identical. The script starts with  $g$  also the identity, so the initial  $f$  is pretty boring. However, if you press 'b' on an interior circle of  $h$ , you will that to become a branch circle for  $g$ . The image of  $f$  is immediately more interesting. You can add additional branch points and see some of the behavior described in Chapter 4. In particular, since the boundary radii are identical for  $h$  and  $g$ , the complex dilatation function  $w(z)$  is a discrete finite Blaschke product.

Also in [HGIntro](#) we display, strictly for purposes of comparison, the addition  $h + g$  (no conjugation bar on  $g$  now). The result  $f = h + g$ , which appears in p2, will approximate the classical sum of these analytic functions. However, the radii associated with  $f$  are simply the sums of radii from  $h$  and  $g$ , and when you display the circles in p2 you see that they are not a *packing* label. In other words, in this algebraic situation, the circles are no longer doing the work for you.

The next script, [HGexamples](#), illustrates a few additional examples related to Chapter 4.

A last script, [HGtrials](#), sets up various adjustments you can implement with mouse and keys for more open-ended experiments. And, of course, you can also add actions of your own devising. Certain keys will increase boundary radii of  $g$ , and this introduces you to another feature: faces of  $f$  (in p2) are colored blue if  $f$  reverses their orientation. We know from the classical theory, that this can happen only if  $|G'|/|H'| > 1$  on some part of  $\partial\mathbb{D}$ . In the discrete setting, absolute values of derivatives are replaced by sharp functions; if you return to the definition of the sharp function and recall that  $h, g$  share the same domain packing, you see that the discrete condition becomes  $g^\# / h^\# > 1$  on some vertices of  $\partial K$ . Therefore, by manipulating the relative sizes of the boundary circles in p0 and p1, we affect the orientation of  $f = h + \bar{g}$ .

**EXERCISE 6.17.** Verify this bit of local geometry, which may account for the properties of these discrete harmonic functions. Consider euclidean triangles  $T$  and  $t$  with vertices  $\{Z_1, Z_2, Z_3\}$  and  $\{z_1, z_2, z_3\}$ , respectively. Assume that each of these triples is positively oriented. Now form a new triangle  $\Delta$  with vertices  $\{w_1, w_2, w_3\}$ , where  $w_j = Z_j + \bar{z}_j, j = 1, 2, 3$ . (The bar here denotes complex conjugation of  $z_j$ .) The issue is whether  $\Delta$  is necessarily positively oriented.

In general, the answer is "no", but here is a pretty fact that you should try to prove: *If  $T$  and  $t$  happen to be formed by mutually tangent triples of circles  $\{C_1, C_2, C_3\}$  and  $\{c_1, c_2, c_3\}$ , and if  $\text{radius}(c_j) / \text{radius}(C_j) \leq 1, j = 1, 2, 3$ , then  $\Delta$  is indeed positively oriented.* In other words, the classical boundary condition for preserving orientability of  $H + \bar{G}$ , namely  $|G'|/|H'| < 1$ , carries over to the discrete case at the local face level.

Use your favorite computer math package to build examples. Show first that  $\Delta$  can indeed have reverse orientation in general. But even more is true: show that even if you arrange that the side lengths of  $t$  are individually less than the corresponding side lengths of  $T$ ,  $\Delta$  may still end up with reverse orientation. Mysteriously, however, when those side lengths are **sums of radii**, and the radii for  $t$  are individually less than those for  $T$ , then  $\Delta$  **remains** positively oriented!

## 6.6. Convergence

We have introduced the basic local geometry of circle packings, moved on to defining mappings between packings, and investigated the emergent global geometric properties. The claim has been made — with convincing evidence, I hope — that these mappings deserve the name "discrete" analytic functions. Can we, in fact, move beyond mere parallels?

The answer is a resounding yes. Though granularity prevents discrete analytic functions from actually **being** analytic functions, with a suitable *refinement* process for improving resolution, they can be made to approximate analytic functions. Circle packing got started with exactly this approximation issue with the celebrate Rodin and Sullivan Theorem explained below. Let's see it in action first.

Ⓢ `DiscreteRMT` Ⓢ picks up from our earlier script Ⓢ `Cookie` Ⓢ. Recall that you can build a range packing  $P$  by cookie cutting a region  $\Omega$  from a regular hexagonal packing. The maximal packing  $Q$  having the same combinatorics becomes the domain in  $\mathbb{D}$ . A standard *CirclePack* normalization puts the circle for the  $\alpha$  vertex at the origin in both  $Q$  and  $P$ , and a  $\gamma$  vertex on the imaginary axis in each. The map  $f : Q \rightarrow P$  is then a discrete conformal map.

At issue: How well do these discrete mappings approximate the classical conformal mapping  $F : \mathbb{D} \rightarrow \Omega$  guaranteed by the Riemann Mapping Theorem? In Ⓢ `Cookie` Ⓢ it was suggested that you try the optional 'i' and 'd' commands to increase or decrease the radii of the hexagonal overlay. If you didn't try them, you have another chance here in Ⓢ `DiscreteRMT` Ⓢ. Making those circles smaller — hence using more of them — is a key method for improving resolution. The effectiveness of such refinements is the content of the foundational theorem by Burt Rodin and Dennis Sullivan, paraphrased here.

**THEOREM 6.18** (Rodin and Sullivan, [5]). Let  $\Omega$  be a bounded simply connected domain in the plane and for each  $\epsilon > 0$  let  $f_\epsilon$  be the discrete conformal mapping created as in the script, based on a regular hexagonal packing with circles of radius  $\epsilon$ . Let  $F : \mathbb{D} \rightarrow \Omega$  be the classical conformal mapping with the same normalization. Then the functions  $f_\epsilon$  converge to  $F$  uniformly on compact subsets of  $\mathbb{D}$  as  $\epsilon \rightarrow 0$ . Moreover, the ratio functions  $f^\#$  converge uniformly on compact to the modulus classical derivative,  $|F'|$ .

The script illustrates the quality of a particular discrete mapping in two ways. One is rather self explanatory, imposing a spoke-and-wheel grid on the domain disc and having the discrete analytic function carry that forward to  $\Omega$ . This result is compared in Figure 6.9 with the analogous result computed using Toby Driscoll's Schwarz-Christoffel software (citeDSCT). (These markings are related to the “flow” lines employed in Chapter 3.)

A second test of quality is illustrated in Ⓢ `DiscreteRMT` Ⓢ *via* color-coding. Let me briefly describe this. First, treat both  $Q$  and  $P$  as euclidean packings and pay attention to their carriers. We can reimagine  $f : Q \rightarrow P$  as a more familiar type of function; namely, as a function  $f : \text{carr}(Q) \rightarrow \text{carr}(P)$  mapping each face of  $\text{carr}(Q)$  onto the corresponding face of  $\text{carr}(P)$ , with the vertices (circle centers) mapped to corresponding vertices. This can be accomplished nicely by defining  $f$  to be **piecewise affine**, that is affine on each face of  $\text{carr}(Q)$ . I will let the reader look into the definition. For our purposes it is enough to note that if  $T_1$  is a face of  $\text{carr}(Q)$ , then  $f$  maps  $T_1$  to the corresponding face  $t_1$  of  $\text{carr}(P)$  and has the form

$$f(x, y) = (ax + by + x_0, cx + dy + y_0), \quad (x, y) \in T_1,$$

for some real values  $a, b, c, d, x_0, y_0$ . Such a mapping is **quasiconformal** with **dilatation** given by  $K = w + \sqrt{w^2 - 1}$ , where  $w = (a^2 + b^2 + c^2 + d^2)/(2(ad - bc))$ . In particular,  $K \geq 1$  and  $f$  is conformal (analytic) on  $T_1$  iff  $K = 1$ , that is, iff  $T_1$  and  $t_1$

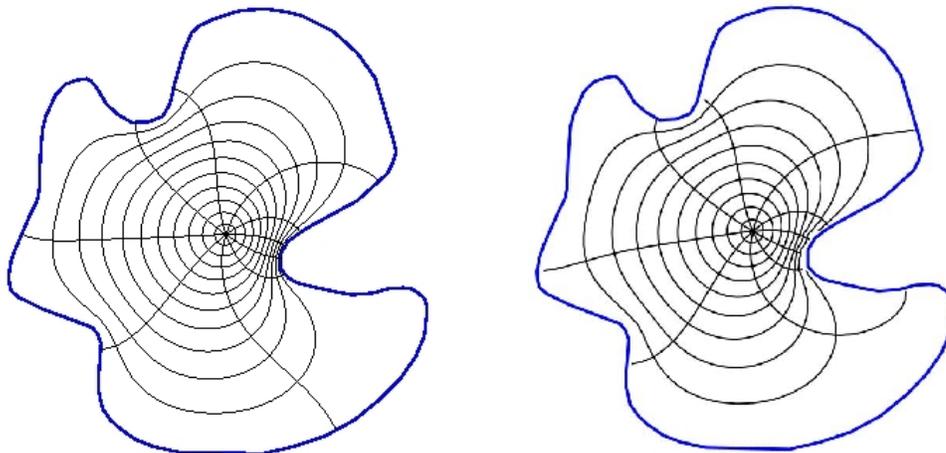


FIGURE 6.9. Two conformal approximations: (a) circle packing, (b) Schwarz-Christoffel

are similar triangles. In general, the size of  $K$  reflects the how close the mapping  $f$  is to being analytic. The experiments in `DiscreteRMT` color code the faces of the range packings. For each face, the further the dilatation of  $f$  on that face is from 1, the more red its color, up to some cutoff, (e.g.  $K = 1.2$ ); for dilatations larger than the cutoff, the face is colored blue. In this scheme, the closer the color gets to white, the closer the dilatation is to 1, that is, the closer the function is to being analytic. In the script commands have been set up to display grids and this color coding: run the script with increasing refinements levels to see the affects on accuracy.

The convergence of discrete objects to their classical counterparts seen in the Rodin/Sullivan theorem actually holds much more generally, indeed, in essentially all the settings we have discussed in this chapter. First, a few cautionary words on increasing the complexity:

- Computational penalty: repacking and display times can increase dramatically.
- Roundoff errors: these can accumulate, interfering with layout, display, and other operations.
- Display crowding: objects and indices are harder to see, harder to pick out with the mouse, and so forth, and display updates are slower.

Nevertheless, *CirclePack* can handle packings with hundreds of thousands of circles, so give these scripts a try. A few ill advised mouse clicks can push you into a computational neverland — but of course, you can always kill *CirclePack* and start over!

**EXERCISE 6.19.** `RandomRefine` illustrates the use of random packings in place of regular hexagonal packings. Again, you should create you own simply connected region  $\Omega$ , then apply the commands developed in the script to create and investigate the

discrete conformal mappings. Refinement is accomplished by increasing the number of circles in the random packing.

EXERCISE 6.20. `MultiRefine` illustrates maps between multiply connected regions, as in `CircleDomains`. If you want to modify the region  $\Omega$ , note that each component of  $\partial\Omega$  requires its own “PATH” file.

EXERCISE 6.21. Run `ErfRefine` to pursue hex refined packings for the “error” function we encountered in `Erf` (see [6, §14.4]). One method for judging how closely the results mimic the classical  $erf(z)$  is to compare the images of the unit circle under the discrete map and the classical map. Methods are given in the script to display the former. However, you should use your favorite math package to generate the latter; the script explains how to read and display a path given as a list of  $x, y$  coordinates.

EXERCISE 6.22. This exercise sets a simple task: estimate the complex number  $B(3i/4)$  for a classical finite Blaschke product  $B$ . However, all you know about  $B$  is this:  $B(0) = 0$ ,  $B(1) = 1$ ,  $B$  has three zeros (counting multiplicities), and  $B$  has branching (i.e.,  $B'$  vanishes) at  $z_1 = (1 + i)/2$  and  $z_2 = -1/2$ . It is known that appropriate discrete finite Blaschke products  $b_n$  will converge uniformly on compact subset of  $\mathbb{D}$  to  $B$  (an extension of the Rodin/Sullivan theorem to branched packings). `BlaschkeRefine` allows you to create discrete functions  $b_n$  (at various refinement levels) based solely on this type of data. You will have to tailor the construction and normalizations to match this particular  $B$ . Then you are asked to apply two methods for estimating  $B(3i/4)$ : (1) Use the values  $b_n(3i/4)$  for the  $b_n$  you constructed. (2) Locate the zeros of the  $b_n$ , use those to build a classical finite Blaschke product  $\hat{B}$  close to  $B$ , then compute  $\hat{B}(3i/4)$  in your favorite mathematics software. The point is that classical Blaschke products are best constructed from their zeros, while their discrete analogues are best constructed from their branch points.

Nearly every script from Sections 6.4 and 6.5 involves packings which are subject to the types of refinements we’ve just discussed, and invariably the discrete objects will converge under this refinement to their classical counterparts. The reader should pick a topic and create a script which implements appropriate refinement: discrete exponentials, discrete rational maps, discrete ratio function constructions, discrete harmonic functions — each project will have its distinct character.

## 6.7. Wrapup

I hope the reader has enjoyed this chapter. Perhaps it has given you a new way to look at and think about analytic functions. As discrete theories go, circle packing is arguably unique in the breadth, depth, and fidelity with which it captures its subject. The essential phenomena appear with even the simplest of packings, where you seem to glimpse the heart of analyticity. As complexity increases, that fidelity only grows,

to the point that you can reliably investigate — perhaps even discover new phenomena — in the classical theory. So it is a full-featured “quantum” theory, classical in the limit, with experimental and visualization capabilities to boot.

This chapter may also have introduced you to a topic of interest in its own right. There is a lot of room for growth in circle packing: packings on surfaces, packings of random triangulations, graph embedding, and a growing list of applications. And wherever circle packing goes, there, too, will be complex analysis!

### 6.8. Additional Exercises

Here are three additional exercises to stretch your understanding of circle packing. The first challenge in each is simply understanding the mechanisms being illustrated in the scripts. After that, however, you may see new directions for experiments, and you are encouraged to set your own path of discovery.

EXERCISE 6.23. Seeing the connection between triangulations and circle packings may have piqued your curiosity about more general triangulations. Whereas our complexes  $K$  have been simply connected up to this point, the situation in  $\textcircled{S}$  Torus  $\textcircled{S}$  is quite different: here  $K$  triangulates a torus. A torus is topologically equivalent to the surface of an inner tube, so you would rightly guess that one cannot realize a circle packing for  $K$  in the plane. Yet the script presents you with the “packing”  $P$  shown on the left in Figure 6.10.

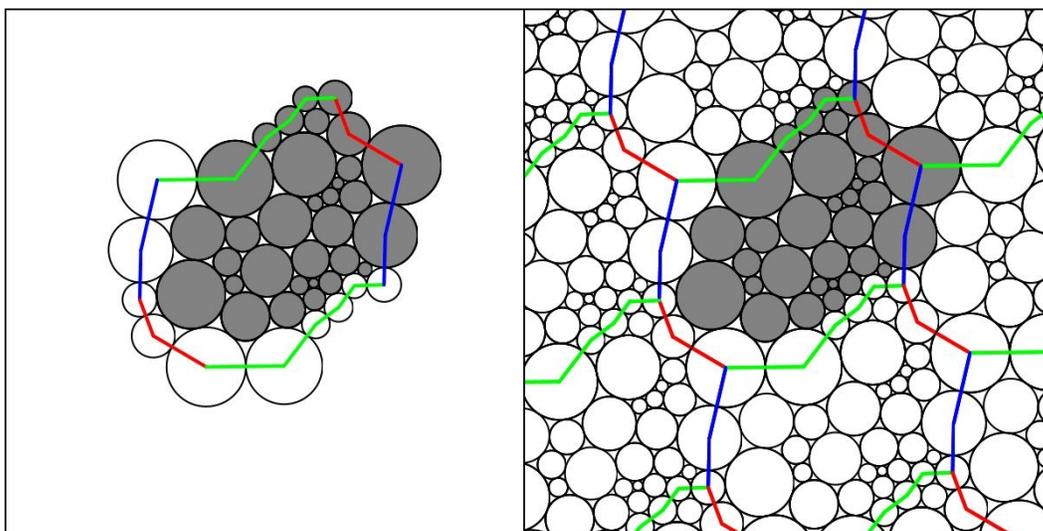


FIGURE 6.10. Circle Packing of a torus with color-coded side pairings

Interpretation of  $P$  rests on the color coded “side-pairings”: The 32 shaded circles form the packing, but additional tangencies exist on the torus. Since we cannot realize these in the plane, we create ghost circles (the unshaded ones on the left): the circles

centered along one colored edge are identified with corresponding circles along the other edge sharing that color. You may be able to see the matchings in the image either from the combinatorics or the radii, but the script should help you explore. The image in the right of Figure 6.10 presents this in another fashion; here translated copies of the packing tiling the plane.

Ultimately, however, it's up to you to come to terms with this representation for circle packings of tori. It may help to create your own triangulated torus and insert it into the script data section in place of the current packing. Encoding of complexes is described in the “Formats” tab of the Help Frame of *CirclePack*. Let me offer a specific challenge: create a combinatorial torus  $K$  having just 7 vertices — the smallest number possible for a proper triangulation (that is, one in which any two faces are disjoint, share a single vertex, or share an edge and its end vertices). This may be a little harder than you expect.

**EXERCISE 6.24.** An experimental capability of *CirclePack* that we touched on briefly in [Patterns](#) involves the “adjoin” operation for triangulations. Observe that when two simply connected triangulations  $K_1$  and  $K_2$  have the same number of boundary vertices, they can be pasted to one another along the full length of their boundaries. The result is a new triangulation  $K$  of the sphere. A packing  $P$  for  $K$  in  $\overline{\mathbb{C}}$  then provides simultaneous packings for  $K_1$  and  $K_2$  while also giving an embedding for their common boundary curve, the so-called **welding curve**, on the sphere.

The basics are demonstrated with Owl in [OwlWelding](#). Among other things, interest lies in how the shape of the welding curve is determined by our choices in attaching  $K_1$  and  $K_2$ . There is a connection with conformality; suffice it to say that it is related to a classical topic called **conformal welding** (see [9]). Learn about the “double” and “adjoin” operations, and let the experiments begin.

**EXERCISE 6.25.** The *CirclePack* algorithms for computing circle packings work in euclidean and hyperbolic geometry. Unfortunately, there is as yet no general algorithm that works directly in spherical geometry, and the spherical packings in previous scripts have been stereographic projections from the plane or disc. All is not lost, however. At least a smattering of examples can be created by hand using basic spherical geometry in conjunction with combinatorial symmetries. Figure 6.11 illustrates an example with the combinatorics of the soccer ball: (a) shows the maximal packing, while (b) is a branched packing having the same combinatorics. Investigate this further in [SoccerPack](#). **Try it out!**

Both packings in Figure 6.11 are constructed using the geometry of Schwarz triangles. The univalent packing  $Q$  shows the symmetry of the regular dodecahedron: 12 symmetrically distributed circles each with five neighbors, the remaining 20 circles each with 6 neighbors. (This packing can, in fact, be created by standard methods in *CirclePack*; see [MaxPackSph](#)). The packing  $P$  shares the same combinatorics, but the degree five circles are all branch circles, so  $P$  defines a 7-fold covering of the

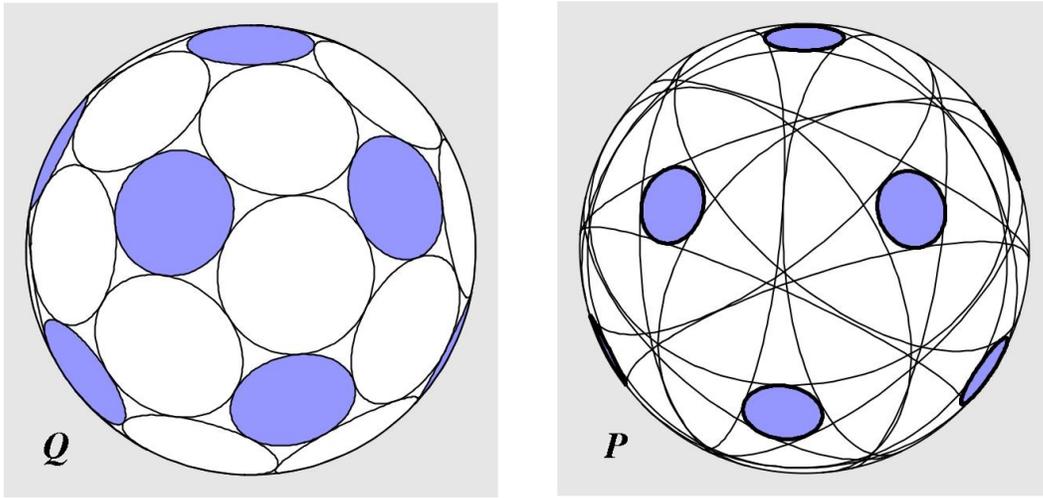


FIGURE 6.11. Circle packings with soccer ball combinatorics

sphere. The map  $f : Q \rightarrow P$  is thus a discrete rational function with 12 simple branch points and valence 7.

Both  $Q$  and  $P$  can be created using the spherical triangles of Figure 6.12, two of several triangles known as “Schwarz” triangles. The Schwarz  $\langle 2, 3, 5 \rangle$  triangle  $t$  (i.e., having angles  $\pi/2, \pi/3$ , and  $\pi/5$ ) is pictured in Figure 6.12(a). A second triangle, the  $\langle 2, 3, 5/2 \rangle$  Schwarz triangle  $T$ , is shown in (b) (at a different scale). Dashed lines on  $T$  show how it is built from 7 copies of  $t$ . A Pattern of two circular arcs is imprinted on  $t$  in (a) and the analogous pattern is imprinted on  $T$  in (b).

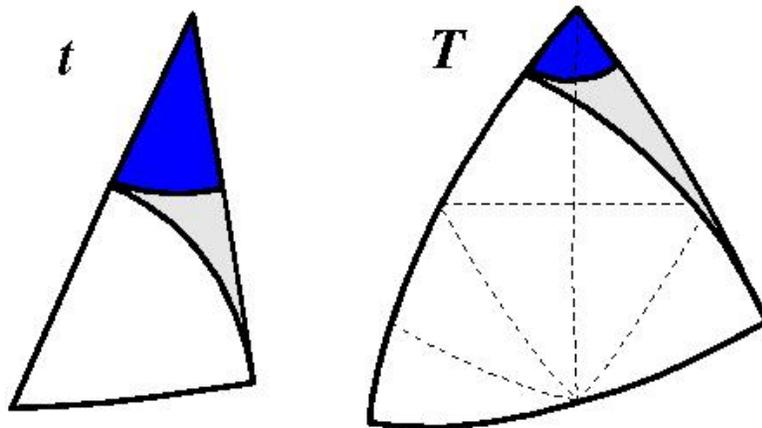


FIGURE 6.12. Schwarz triangles  $t$  and  $T$

A Schwarz triangle can be reflected across any of its three edges to form a copy of itself with reverse orientation. Repeated reflections will tile the sphere; in the case of  $t$  this tiling involves 120 non-overlapping copies. Carrying the imprinted arcs along during the reflection process gives us a univalent circle packing with 42 circles, our maximal packing  $Q$ . The identical reflection process, but now starting with  $T$ , gives 120 copies which tile the sphere 7 layers deep and whose imprinted arcs give our packing  $P$ . In any copy of  $T$ , it takes 10 reflections around the vertex with angle  $2\pi/5$  to close up, and this accounts for the 12 branch points of  $P$ . The script [S Soccer S](#) lets you check one of these visually to see the 5 (huge) neighboring circles wrap twice around it.

Schwarz triangles were first used to generate branched circle packings of the sphere in [1] (see Appendix H of [6]). However, the example of Figure 6.13 and other equally beautiful examples were created by Samantha Corveno, one of my REU (Research Experience for Undergraduates) students.

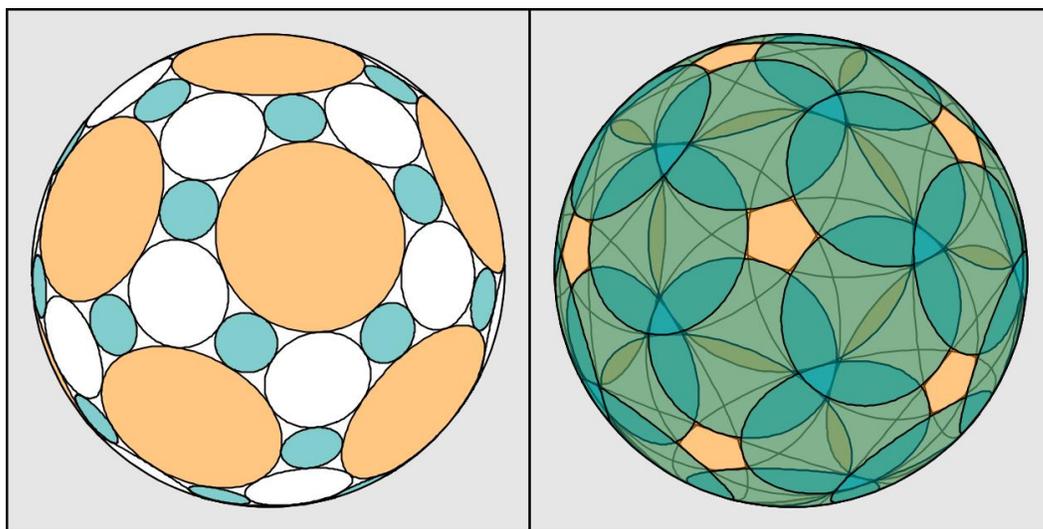


FIGURE 6.13. A branched packing due to Samantha Corvino

There are a handful of additional Schwarz triangles, and other options for circular markings, so the challenge to you is to create yet more examples. For each, you will need to encode the combinatorics  $K$  for *CirclePack*, which can then generate the associated maximal packing. However, *CirclePack* cannot find the radii and centers for the branched packing. For this you will have to apply some spherical trigonometry, record the results in a packing file and *CirclePack* will then display them. To get you started, I've put another of Samantha's examples with more details in [S Tri432 S](#). ***Try it out!***

## Bibliography

1. Philip L. Bowers and Kenneth Stephenson, *A branched Andreev-Thurston Theorem for circle packings of the sphere*, Proc. London Math. Soc., **73** (1996), 185–215.
2. Charles R. Collins and Kenneth Stephenson, *A circle packing algorithm*, Computational Geometry: Theory and Applications **25** (2003), 233–256.
3. Tobin A. Driscoll and Lloyd N. Trefethen, *Schwarz-Christoffel Mapping*, Camb. Univ. Press, New York, 2002.
4. Tobin Driscoll, *Schwarz-Christoffel Toolbox, for Matlab*, <http://www.math.udel.edu/~driscoll/software/SC/>
5. Burt Rodin and Dennis Sullivan, *The convergence of circle packings to the Riemann mapping*, J. Differential Geometry **26** (1987), 349–360.
6. Kenneth Stephenson, *Introduction to circle packing: the theory of discrete analytic functions*, Camb. Univ. Press, New York, 2005.
7. Kenneth Stephenson, *Circle packing: A mathematical tale*, Notices Amer. Math. Soc. **50** (2003), no. 11, 1376–1388.
8. William Thurston, *The finite Riemann mapping theorem*, 1985, Invited talk, An International Symposium at Purdue University in celebrations of de Branges' proof of the Bieberbach conjecture, March 1985 (unpublished).
9. G. Brock Williams, *Discrete conformal welding*, Indiana Univ. Math. J., **53** (2004), 765–804.