

# SCHOTTKY DOCUMENTATION

DANNY CALEGARI AND ALDEN WALKER

## 1. INTRODUCTION

The program `schottky`[2] is a tool for studying the space of two generator iterated function systems generated by complex dilations. It is intended as a companion to the paper[1], which is the theoretical reference to the algorithms implemented in `schottky`. This document describes how to compile and use the program.

## 2. COMPILATION

On linux, the only thing that should be required is a C++ compiler, the program `make`, and the X11 development headers. If the headers `Xlib.h` and `Xlib.h` are in the subdirectory `X11/` in a location that the C++ compiler searches, everything should be fine. Change lines 5–6 of `ifs_gui.h` if the headers are in a nonstandard location. Similarly, the `X11` library should be in the search path. Adjust the makefile so that it points to the correct place if necessary.

On OSX, the program XQuartz (the version of X11 for OSX) needs to be installed. It is available at <http://xquartz.macosforge.org/>. The computer also needs to have XCode tools installed to get the C++ compiler. This is available from the Apple store.

## 3. LIVE TECHNICAL SUPPORT

If something goes wrong or is confusing, the best thing to do is to email Alden at [akwalker@math.uchicago.edu](mailto:akwalker@math.uchicago.edu).

## 4. USAGE

The left panel shows the limit set. The right panel shows parameter space. The check boxes allow the user to change the appearance of the limit set and change what is plotted in parameter space.

### 4.1. Limit set.

**Clicking:** Clicking on the panel recenters the drawing.

**Depth:** Change the depth to which the limit set is drawn. This changes  $n$ , where the picture drawn is the image of a point (non-chunky mode) or disk (chunky mode) under all words in  $f, g$  of length  $n$ .

**Auto depth:** Dynamically adjust the depth so that the limit set always looks sharp. This is helpful for zooming in a lot. In this mode, the selected depth is ignored.

**Zoom:** Zoom in or out

**Chunky:** If chunky is not selected, the drawing is the image of a point in  $\Lambda_z$  under all words of length  $n$ . If chunky is selected, the drawing is a union of disks. As explained in [1], if chunky is selected, the drawing is guaranteed to contain the limit set.

**Plot uv graph:** This mode (1) indicates the labels on all the balls at the given depth (match the drawing depth and the uv graph depth to see this) and (2) connects these vertices as appropriate. The vertices are connected using the following rules: there is an edge connecting a vertex in  $f\Lambda_z$  to one in  $g\Lambda_z$  exactly when these vertices realize the distance between  $f\Lambda_z$  and  $g\Lambda_z$  (they are a closest pair). There is an edge connecting two vertices  $v, w$  in  $f\Lambda_z$  or  $g\Lambda_z$  if there is a shortest pair of vertices  $a \in f\Lambda_z$ ,  $b \in g\Lambda_z$  such that (1) a prefix of  $a$  is a suffix of  $v$  and (b) the matching prefix of  $b$  is a suffix of  $w$ . Paths in this graph essentially construct short hop paths.

**nIFS:** This stands for  $n$ -generator IFS. Although right now it is set to simply plot the 3-generator IFS which computes the differences between points in  $\Lambda_z$ . Consider navigating to a renormalization point, turning off chunky and colors, and zooming in to observe the similarity with the set  $\mathcal{M}$  described in [1].

**2d IFS:** This ignores the imaginary part and only works when the real part of the parameter is between 0.6 and 0.68. It plots the affine limit set as described in [1].

#### 4.2. Mandelbrot.

**Clicking:** Clicking on the panel changes the selected parameter. Clicking and dragging works. Right clicking selects a new parameter and zooms in.

**Recenter:** Recenter the window on the selected parameter.

**Zoom:** Zoom in and out

**Mesh size:** How many true pixels make up one “pixel” in the grid. Setting mesh size smaller makes the picture sharper.

**Connectedness:** Plot set  $\mathcal{M}$  (run the disconnectedness algorithm). The shade of blue indicates how hard the program had to work to find a pair of disks which intersect at the required depth.

**Contains 1/2:** Plot set  $\mathcal{M}_0$ .

**Traps:** Find traps in a region. This will not work at all until the panel window is quite zoomed in. It functions by far the best at the boundary of  $\mathcal{M}$ .

**Limit traps:** If this is selecting, it will find limit traps instead of traps. This *only* works in a neighborhood of the specific renormalization point  $\omega$  from [1].

**Dirichlet:** In the disconnectedness region, plot how many pairs of points in  $\Lambda_z$  are closest. Darker shades indicate more pairs of closest points. Note if  $\mathcal{M}_0$  and Dirichlet are both turned on, the white (a single pair) Dirichlet regions and  $\mathcal{M}_0$  limit on each other.

**Set C:** Highlight some parameters which are known to have simply connected limit set. This is  $\mathcal{M}_1$  as defined in [1]. This feature is somewhat experimental.

**Theta:** Plot a rotation parameter. This is also an experimental feature.

**Write window coords:** This outputs the window to the terminal.

**Write picture:** This writes the current picture to a square bitmap with the given resolution. See the terminal for the status.

**Draw path:** Allows the user to draw a path by clicking in parameter space.  
**Find boundary path:** Draws a path obtained by following the boundary of the component of Schottky space in which the current parameter lies. This only functions if the selected parameter is *not* in set  $\mathcal{M}$ .

**Path options:**

**Delete:** Delete the path

**Find traps along path:** Find a sequence of balls of traps to certify that the path lies in the interior of  $\mathcal{M}$ . This will use the trap depth as selected above and use limit traps if that is selected.

**Find coords along path:** Experimental feature

**Create movie along path:** Create a movie. This requires that the program `ffmpeg` be installed in the directory `../ffmpeg/`.

**Find uv words along path:** Finds a sequence of closest pairs of balls along the path.

**Find half words along path:** This finds balls of points guaranteed to be in set  $\mathcal{M}_1$ , as described in [1]. This is somewhat tricky to use. It uses the limit set depth to set the initial size of the chunks it finds. Thus, to certify a hole, one should navigate to the hole, zoom in on the limit set and increase the depth until the hole is evident there, then decrease the depth a little. Then draw the path and find half words along it.

## REFERENCES

- [1] D. Calegari, S. Koch, and A. Walker *Roots, Schottky semigroups, and a proof of Bandt's conjecture*, preprint: arXiv: 1410.8542
- [2] D. Calegari and A. Walker, `schottky`, computer program available from the authors' web-pages.

DEPARTMENT OF MATHEMATICS, UNIVERSITY OF CHICAGO, CHICAGO, ILLINOIS, 60637  
*E-mail address:* `dannyc@math.uchicago.edu`

DEPARTMENT OF MATHEMATICS, UNIVERSITY OF CHICAGO, CHICAGO, ILLINOIS, 60637  
*E-mail address:* `akwalker@math.uchicago.edu`