Dirichlet Domains

Alex Karlovitz

In this note, we discuss the details involved in computing a Dirichlet domain for a group Γ acting discretely on the upper half-plane model of hyperbolic 2-space. As an example, we construct the compact arithmetic surface described by Kontorovich in [Kon11].

Constructing a Dirichlet Domain

To construct a Dirichlet domain, one starts with a specific point $z \in \mathbb{H}$ (we often take $z = 2i$). Then, we find points in the orbit of z under Γ which are relatively close to z. For each such point z_*, we find the geodesic of points which are equidistant from z and z^*. The set of points in $\mathbb{H}$ which are closer to z than to any of the z_*'s will constitute a fundamental domain for $\Gamma\backslash\mathbb{H}$ so long as we took enough points in the orbit.

Here is pseudocode for finding the geodesic of points equidistant from two points z_1 and z_2 in $\mathbb{H}$.

```
def find_equidistant_geodesic(z1, z2) :
    # give names to real and imaginary parts for ease of reading
    x1, y1 = real(z1), imag(z1)
    x2, y2 = real(z2), imag(z2)

    # Step 1: get geodesic G through z1 and z2

    if x1 == x2 :
        # G is vertical line
        G = line((x1, 0), (x1, y1))
    else :
        # G is circle of center c and radius r
        c = (|z1|^2 - |z2|^2)/2/(x2 - x1)
        r = sqrt( (x1 - c)^2 - y1^2 )
        G = circle(c, r)

    # Step 2: get point z0 on G equidistant from z1 and z2 (in hyperbolic distance)

    eqn1 = |z - z1|^2/4/imag(z)/y1 == |z - z2|^2/4/imag(z)/y2
    if isCircle(G) :
        eqn2 = (z - c)^2 == r^2
    else :
        eqn2 = real(z) == x1
    z0 = solve((eqn1, eqn2), unknown=z)

    # Step 3: get geodesic G0 through z0 which meets G at a right angle

    x0, y0 = real(z0), imag(z0)
    if isCircle(G) :
        # first get slope of g at z0
        s = (c - x0)/y0

        # if the slope is 0, return vertical line through z0
```

```
    if s == 0 :
        return line((x0, 0), (x0, y0))

    # solve for center a and radius R
    a = -y0/s + x0
    R = sqrt( (x0 - a)^2 + y0^2 )

    G0 = circle(a, R)
else :
    # z0 is at top of circle
    G0 = circle(x0, y0)

return G0
```

References

[Kon11] Alex Kontorovich. “EXPOSITORY NOTE: An Arithmetic Surface”. 2011.