

Reflection about a geodesic passing through two given points in the Poincaré and Klein disk models of hyperbolic geometry

Frank Nielsen
Sony Computer Science Laboratories Inc
Tokyo

15 December 2023

Abstract

We report in this note a direct closed-form formula for the reflection operation about a geodesic defined by two given points in the Poincaré disk.

1 Geodesics and pregeodesics

In metric geometry (including Riemannian geometry) with distance ρ , a geodesic $\gamma(l)$ is parameterized by a constant speed parameter l so that we have

$$\rho(\gamma(l), \gamma(l')) = |l - l'| \rho(\gamma(0), \gamma(1)).$$

The constant speed parameterization is related to arc length parameterization by $s = l \rho(\gamma(0), \gamma(1))$, where $p_0 = \gamma(0)$ and $p_1 = \gamma(1)$ are the geodesic endpoints.

To contrast with geodesics, a pregeodesic $\bar{\gamma}(t) = \bar{\gamma}(l(t))$ is a reparameterization of the geodesic such that $l(t)$ is a smooth and invertible with inverse function $t(l)$. Pregeodesics can yield simplified mathematical expressions of parameterizations, and express equivalently the geodesic curves:

$$c_\gamma = \{\gamma(l) : l \in [0, 1]\} = \{\bar{\gamma}(t) : t \in [a, b]\}.$$

For example, in the Klein model of hyperbolic geometry, pregeodesics passing through two given points k_1 and k_2 of the unit disk \mathbb{D} are Euclidean line segments parameterized by:

$$\bar{\gamma}_K(t) = k_1 + t(k_2 - k_1),$$

with geodesic curve $c_{\gamma_K} = [k_1 k_2]$. The geodesic equation with constant speed parameterization in Klein model has been reported in [3], i.e., the function $l(t)$ such that

$$\gamma_K(l) = \bar{\gamma}_K(l(t))$$

is given in closed-form.

2 Geodesic curves in Klein model

To find the complete geodesic $\Gamma_K = \{\gamma_K(l) : l \in \mathbb{R}\}$ in Klein model, we need to find t_m and t_M such that Γ_K is the line passing through $[k_1 k_2]$ clipped to the unit disk \mathbb{D} . t_m and t_M are the two solutions of the quadratic equation:

$$\langle k_1 + t(k_2 - k_1), k_1 + t(k_2 - k_1) \rangle = 1$$

A model of a geometry is said conformal if the angles of two curves $c_1(t)$ and $c_2(t)$ intersecting at t_0 match the Euclidean angles. The Poincaré disk model is conformal but not the Klein model (except at the origin).

$$\begin{aligned}
f a 0 = & - \frac{(b 2 - b 1) b 3^2 + (-b 2^2 + b 1^2 - a 2^2 + a 1^2) b 3 + b 1 b 2^2 + (-b 1^2 + a 3^2 - a 1^2) b 2 + (a 2^2 - a 3^2) b 1}{(2 a 2 - 2 a 1) b 3 + (2 a 1 - 2 a 3) b 2 + (2 a 3 - 2 a 2) b 1}, b 0 = \\
& \frac{(a 2 - a 1) b 3^2 + (a 1 - a 3) b 2^2 + (a 3 - a 2) b 1^2 + (a 2^2 - a 1^2) a 3 + a 1 a 2^2 - a 1^2 a 2}{(2 a 2 - 2 a 1) b 3 + (2 a 1 - 2 a 3) b 2 + (2 a 3 - 2 a 2) b 1}, r 0 = \text{sqrt}((b 2^2 - 2 b 1 b 2 + b 1^2 + a 2^2 - 2 a 1 a 2 + a 1^2) b 3^4 + \\
& (-2 b 2^3 + 2 b 1 b 2^2 + (2 b 1^2 - 2 a 2^2 + 4 a 1 a 2 - 2 a 1^2) b 2 - 2 b 1^3 + (-2 a 2^2 + 4 a 1 a 2 - 2 a 1^2) b 1) b 3^3 + (b 2^4 + 2 b 1 b 2^3 + (-6 b 1^2 + 2 a 3^2 + (-2 a 2 - 2 a 1) a 3 + 2 a 2^2 - 2 a 1 a 2 + 2 a 1^2) b 2^2 + \\
& (2 b 1^3 + (-4 a 3^2 + (4 a 2 + 4 a 1) a 3 + 2 a 2^2 - 8 a 1 a 2 + 2 a 1^2) b 1) b 2 + b 1^4 + (2 a 3^2 + (-2 a 2 - 2 a 1) a 3 + 2 a 2^2 - 2 a 1 a 2 + 2 a 1^2) b 1^2 + (2 a 2^2 - 4 a 1 a 2 + 2 a 1^2) a 3^2 + \\
& (-2 a 2^3 + 2 a 1 a 2^2 + 2 a 1^2 a 2 - 2 a 1^3) a 3 + a 2^4 - 2 a 1 a 2^3 + 2 a 1^2 a 2^2 - 2 a 1^3 a 2 + a 1^4) b 3^2 + (-2 b 1 b 2^4 + (2 b 1^2 - 2 a 3^2 + 4 a 1 a 3 - 2 a 1^2) b 2^3 + \\
& (2 b 1^3 + (2 a 3^2 + (4 a 2 - 8 a 1) a 3 - 4 a 2^2 + 4 a 1 a 2 + 2 a 1^2) b 1) b 2^2 + \\
& (-2 b 1^4 + (2 a 3^2 + (4 a 1 - 8 a 2) a 3 + 2 a 2^2 + 4 a 1 a 2 - 4 a 1^2) b 1^2 + (-2 a 2^2 + 4 a 1 a 2 - 2 a 1^2) a 3^2 + (4 a 1 a 2^2 - 8 a 1^2 a 2 + 4 a 1^3) a 3 - 2 a 1^2 a 2^2 + 4 a 1^3 a 2 - 2 a 1^4) b 2 + \\
& (-2 a 3^3 + 4 a 2 a 3 - 2 a 2^2) b 1^3 + ((-2 a 2^2 + 4 a 1 a 2 - 2 a 1^2) a 3^2 + (4 a 2^2 - 8 a 1 a 2^2 + 4 a 1^2 a 2) a 3 - 2 a 2^4 + 4 a 1 a 2^3 - 2 a 1^2 a 2^2) b 1) b 3 + (b 1^2 + a 3^2 - 2 a 1 a 3 + a 1^2) b 2^4 + \\
& ((-2 a 3^2 + 4 a 1 a 3 - 2 a 1^2) b 1 - 2 b 1^3) b 2^3 + \\
& (b 1^4 + (2 a 3^2 + (-2 a 2 - 2 a 1) a 3 + 2 a 2^2 - 2 a 1 a 2 + 2 a 1^2) b 1^2 + a 3^4 + (-2 a 2 - 2 a 1) a 3^3 + (2 a 2^2 + 2 a 1 a 2 + 2 a 1^2) a 3^2 + (-4 a 1 a 2^2 + 2 a 1^2 a 2 - 2 a 1^3) a 3 + 2 a 1^2 a 2^2 - 2 a 1^3 a 2 + a 1^4) \\
& b 2^2 + ((-2 a 3^2 + 4 a 2 a 3 - 2 a 2^2) b 1^3 + (-2 a 3^4 + (4 a 2 + 4 a 1) a 3^3 + (-2 a 2^2 - 8 a 1 a 2 - 2 a 1^2) a 3^2 + (4 a 1 a 2^2 + 4 a 1^2 a 2) a 3 - 2 a 1^2 a 2^2) b 1) b 2 + (a 3^2 - 2 a 2 a 3 + a 2^2) b 1^4 + \\
& (a 3^4 + (-2 a 2 - 2 a 1) a 3^3 + (2 a 2^2 + 2 a 1 a 2 + 2 a 1^2) a 3^2 + (-2 a 2^3 + 2 a 1 a 2^2 - 4 a 1^2 a 2) a 3 + a 2^4 - 2 a 1 a 2^3 + 2 a 1^2 a 2^2) b 1^2 + (a 2^2 - 2 a 1 a 2 + a 1^2) a 3^4 + \\
& (-2 a 2^3 + 2 a 1 a 2^2 + 2 a 1^2 a 2 - 2 a 1^3) a 3^3 + (a 2^4 + 2 a 1 a 2^3 - 6 a 1^2 a 2^2 + 2 a 1^3 a 2 + a 1^4) a 3^2 + (-2 a 1 a 2^4 + 2 a 1^2 a 2^3 + 2 a 1^3 a 2^2 - 2 a 1^4 a 2) a 3 + a 1^2 a 2^4 - 2 a 1^3 a 2^3 + a 1^4 a 2^2) / \\
& ((2 a 2 - 2 a 1) b 3 + (2 a 1 - 2 a 3) b 2 + (2 a 3 - 2 a 2) b 1)]
\end{aligned}$$

Figure 1: Solution of the systems of equation defined in Eq. 1.

3 Reflections about geodesics in the Poincaré disk model

Geodesics in the Poincaré model are either arc of circles perpendicular to the disk boundary $\partial\mathbb{D}$, or line segments passing through the origin. Poincaré geodesic curves are sometimes called clines [1] (circles or lines). The isometries in the Poincaré model seen as a subset of the complex plane \mathbb{C} are Möbius transformations. In particular, reflections around a complete geodesic curve $\Gamma_{z_1 z_2}$ passing through two given points $z_1, z_2 \in \mathbb{D}$ is an Euclidean circle inversion, and thus defined by

$$\text{reflect}(z) = \frac{r_0^2}{\bar{z} - \bar{z}_0} + z_0.$$

For example, such reflection operations are used for embeddings tree structures into the Poincaré disk [4]. Usually, reflections are implemented via translations to the origin of the disk. We report in this note a direct closed-form formula.

The reflection leaves the geodesic $\Gamma_{z_1 z_2}$ invariant:

$$\forall z \in \Gamma_{z_1 z_2}, \text{reflect}(z) = z,$$

and we have $\text{reflect}(\text{reflect}(z)) = z$ for any $z \in \mathbb{D}$.

Let $z_3 \in \Gamma_{z_1 z_2}$ be any point inside the Poincaré geodesic. For example, we consider k_3 the Klein interior point given by $\frac{k_1 + k_2}{2}$ where

$$k_i = \frac{2}{1 + z_i \bar{z}_i} z_i, \quad i \in \{1, 2\},$$

and we convert the Klein geodesic interior point to the Poincaré disk:

$$z_3 = \frac{1 - \sqrt{1 - \|k_3\|^2}}{\|k_3\|^2} k_3$$

Then we write the system of equations with $z_j = a_j + ib_j$ for $j \in \{0, 1, 2, 3\}$:

$$\begin{cases} \text{reflect}(a_1 + ib_1) &= a_1 + ib_1, \\ \text{reflect}(a_2 + ib_2) &= a_2 + ib_2, \\ \text{reflect}(a_3 + ib_3) &= a_3 + ib_3. \end{cases} \quad (1)$$

Next, we use the computer algebra system **Maxima**¹ to solve the system of Eq. 1 (Figure 1) using the

¹<https://maxima.sourceforge.io/>

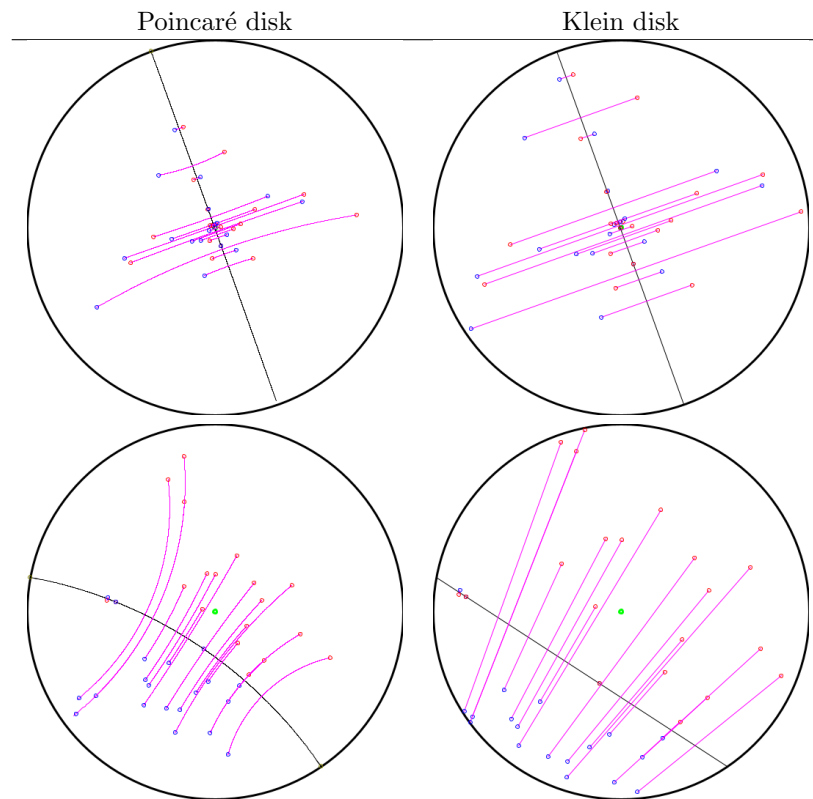


Figure 2: Illustrating reflections around geodesics on the Poincaré and Klein disks.

code:

```

z1: a1+b1%i;
z2: a2+b2%i;
z3: a3+b3%i;

/* circle inversion */
reflect(z, a0, b0, r0) := (r0**2/(conjugate(z)-conjugate(a0+b0%i)))+a0+b0%i;

/* reflection of the points z1, z2, z3 should be identity */
reflect(z1, a0, b0, r0);
realpart(%); ratsimp(%);
eq1: %=a1;
reflect(z1, a0, b0, r0);
imagpart(%); ratsimp(%);
eq2: %=b1;
reflect(z2, a0, b0, r0);
realpart(%); ratsimp(%);
eq3: %=a2;
reflect(z2, a0, b0, r0);
imagpart(%); ratsimp(%);
eq4: %=b2;
reflect(z3, a0, b0, r0);
realpart(%); ratsimp(%);
eq5: %=a3;
reflect(z3, a0, b0, r0);
imagpart(%); ratsimp(%);
eq6: %=b3;

solve([eq1, eq2, eq3, eq4, eq5, eq6], [a0, b0, r0]);
solution: %[2];

```

Maxima reports the unique solution for $z_0 = a_0 + ib_0$ and r_0 defining the reflection:

$$a_0 = -\frac{(b_2 - b_1) b_3^2 + (-b_2^2 + b_1^2 - a_2^2 + a_1^2) b_3 + b_1 b_2^2 + (-b_1^2 + a_3^2 - a_1^2) b_2 + (a_2^2 - a_3^2) b_1}{(2a_2 - 2a_1) b_3 + (2a_1 - 2a_3) b_2 + (2a_3 - 2a_2) b_1}$$

$$b_0 = \frac{(a_2 - a_1) b_3^2 + (a_1 - a_3) b_2^2 + (a_3 - a_2) b_1^2 + (a_2 - a_1) a_3^2 + (a_1^2 - a_2^2) a_3 + a_1 a_2^2 - a_1^2 a_2}{(2a_2 - 2a_1) b_3 + (2a_1 - 2a_3) b_2 + (2a_3 - 2a_2) b_1}$$

Parameter r_0 has a complex expression reported in the screenshot 1.

A point $z = a + ib$ is reflected to a point $z' = a' + ib'$ with

$$a' = \frac{(a - a_0)r_0^2}{(b_0 - b)^2 + (a - a_0)^2} + a_0,$$

$$b' = \frac{(b - b_0)r_0^2}{(b_0 - b)^2 + (a - a_0)^2} + b_0.$$

Figure 2 illustrates some examples of reflections obtained in the Poincaré disk and then converted into the Klein disk.

It is usually easy and numerically robust to implement geometric algorithms like the Voronoi diagram [2] in the Klein disk and visualize the result in the conformal Poincaré disk. The Poincaré disk may be directly considered for some problems like calculating the smallest enclosing hyperbolic ball since a hyperbolic ball in the Poincaré disk model is a Euclidean ball with a shifted center [5].

```

class Reflection
{
    double a0, b0, r0;

    public static double sqr(double x) {return x*x;}

    // Reflect a point
    Point2D reflect(Point2D p)
    {
        double a=p.x; double b=p.y; double xx, yy;
        xx= ((a-a0)*sqr(r0))/(sqr(b0-b)+sqr(a-a0))+a0;
        yy= ((b-b0)*sqr(r0))/(sqr(b0-b)+sqr(a-a0))+b0;

        return new Point2D(xx, yy);
    }

    Reflection(double a, double b, double r)
    {
        a0=a; b0=b; r0=r;
    }
}

// Given three points on a Poincare geodesic (usually p3 is interior point on p1 p) calculates the coefficients of
// the circle inversion
// which gives the hyperbolic reflection
Reflection(Point2D p1, Point2D p2, Point2D p3)

```

```

{
    double a1, b1, a2, b2, a3, b3;
    a1=p1.x;
    b1=p1.y;
    a2=p2.x;
    b2=p2.y;
    a3=p3.x;
    b3=p3.y;

    a0 = -((b2-b1)*Math.pow(b3, 2)+((-Math.pow(b2, 2))+Math.pow(b1, 2)-Math.pow(a2, 2)+Math.pow(a1, 2))*b3+b1*Math.
        pow(b2, 2)+((-Math.pow(b1, 2))+Math.pow(a3, 2)-Math.pow(a1, 2))*b2+(Math.pow(a2, 2)-Math.pow(a3, 2))*b1)
        /((2*a2-2*a1)*b3
        +(2*a1-2*a3)*b2+(2*a3-2*a2)*b1);

    b0 = ((a2-a1)*Math.pow(b3, 2)+(a1-a3)*Math.pow(b2, 2)+(a3-a2)*Math.pow(b1, 2)+(a2-a1)*Math.pow(a3, 2)+(Math.pow
        (a1, 2)
        -Math.pow(a2, 2))*a3+a1*Math.pow(a2, 2)-Math.pow(a1, 2)*a2)/((2*a2-2*a1)*b3+(2*a1-2*a3)*b2+(2*a3-2*a2)*b1);

    r0 = Math.sqrt((Math.pow(b2, 2)-2*b1*b2+Math.pow(b1, 2)+Math.pow(a2, 2)-2*a1*a2+Math.pow(a1, 2))*Math.pow(b3,
        4)+
        ((-2*Math.pow(b2, 3))+2*b1*Math.pow(b2, 2)+(2*Math.pow(b1, 2)-2*Math.pow(a2, 2)+4*a1*a2-2*Math.pow(a1, 2))*b2
        -2*Math.pow(b1, 3)+((-2*Math.pow(a2, 2))+4*a1*a2-2*Math.pow(a1, 2))*b1)*Math.pow(b3, 3)+(Math.pow(b2, 4)+2*b1
        *Math.pow(b2, 3)
        +((-6*Math.pow(b1, 2))+2*Math.pow(a3, 2)+((-2*a2)-2*a1)*a3+2*Math.pow(a2, 2)-2*a1*a2+2*Math.pow(a1, 2))*Math.
        pow(b2, 2)
        +(2*Math.pow(b1, 3)+((-4*Math.pow(a3, 2)))+(4*a2+4*a1)*a3+2*Math.pow(a2, 2)-8*a1*a2+2*Math.pow(a1, 2))*b1)*b2+
        Math.pow(b1, 4)
        +(2*Math.pow(a3, 2)+((-2*a2)-2*a1)*a3+2*Math.pow(a2, 2)-2*a1*a2+2*Math.pow(a1, 2))*Math.pow(b1, 2)+(2*Math.
        pow(a2, 2)-4*a1*a2+2*Math.pow(a1, 2))*Math.pow(a3, 2)
        +((-2*Math.pow(a2, 3))+2*a1*Math.pow(a2, 2)+2*Math.pow(a1, 2)*a2-2*Math.pow(a1, 3))*a3
        +Math.pow(a2, 4)-2*a1*Math.pow(a2, 3)+2*Math.pow(a1, 2)*Math.pow(a2, 2)-2*Math.pow(a1, 3)*a2+Math.pow(a1, 4))
        *Math.pow(b3, 2)+
        ((-2*b1*Math.pow(b2, 4)))+(2*Math.pow(b1, 2)-2*Math.pow(a3, 2)+4*a1*a3-2*Math.pow(a1, 2))*Math.pow(b2, 3)+(2*
        Math.pow(b1, 3)
        +(2*Math.pow(a3, 2)+(4*a2-8*a1)*a3-4*Math.pow(a2, 2)+4*a1*a2+2*Math.pow(a1, 2))*b1)*Math.pow(b2, 2)
        +((-2*Math.pow(b1, 4)))+(2*Math.pow(a3, 2)+(4*a1-8*a2)*a3+2*Math.pow(a2, 2)+4*a1*a2-4*Math.pow(a1, 2))*Math.
        pow(b1, 2)+((-2*Math.pow(a2, 2))+4*a1*a2-2*Math.pow(a1, 2))*Math.pow(a3, 2)
        +(4*a1*Math.pow(a2, 2)-8*Math.pow(a1, 2)*a2+4*Math.pow(a1, 3))*a3-2*Math.pow(a1, 2)*Math.pow(a2, 2)+4*Math.
        pow(a1, 3)*a2-2*Math.pow(a1, 4))*b2
        +((-2*Math.pow(a3, 2))+4*a2*a3-2*Math.pow(a2, 2))*Math.pow(b1, 3)+((-2*Math.pow(a2, 2))+4*a1*a2-2*Math.pow(
        a1, 2)
        +4*a1*Math.pow(a2, 3)-2*Math.pow(a1, 2)*Math.pow(a2, 2))*b1)*b3+(Math.pow(b1, 2)+Math.pow(a3, 2)-2*a1*a3+Math.
        pow(a1, 2))*Math.pow(b2, 4)+
        (((-2*Math.pow(a3, 2))+4*a1*a3-2*Math.pow(a1, 2))*b1-2*Math.pow(b1, 3))*Math.pow(b2, 3)+(Math.pow(b1, 4)+(2*
        Math.pow(a3, 2)
        +((-2*a2)-2*a1)*a3+2*Math.pow(a2, 2)-2*a1*a2+2*Math.pow(a1, 2))*Math.pow(b1, 2)+Math.pow(a3, 4)+((-2*a2)-2*a1
        )*Math.pow(a3, 3)
        +(2*Math.pow(a2, 2)+2*a1*a2+2*Math.pow(a1, 2))*Math.pow(a3, 2)+((-4*a1*Math.pow(a2, 2))+2*Math.pow(a1, 2)*a2
        -2*Math.pow(a1, 3))*a3+2*Math.pow(a1, 2)*Math.pow(a2, 2)-2*Math.pow(a1, 3)*a2+Math.pow(a1, 4))*Math.pow(
        b2, 2)
        +(((2*Math.pow(a3, 2))+4*a2*a3-2*Math.pow(a2, 2))*Math.pow(b1, 3)+((-2*Math.pow(a3, 4)))+(4*a2+4*a1)*Math.pow
        (a3, 3)+((-2*Math.pow(a2, 2))-8*a1*a2-2*Math.pow(a1, 2))*Math.pow(a3, 2)
        +(4*a1*Math.pow(a2, 2)+4*Math.pow(a1, 2)*a2)*a3-2*Math.pow(a1, 2)*Math.pow(a2, 2))*b1)*b2+(Math.pow(a3, 2)-2*
        a2*a3+Math.pow(a2, 2))*Math.pow(b1, 4)
        +(Math.pow(a3, 4)+((-2*a2)-2*a1)*Math.pow(a3, 3)+(2*Math.pow(a2, 2)+2*a1*a2+2*Math.pow(a1, 2))*Math.pow(a3,
        2)+((-2*Math.pow(a2, 3)
        +2*a1*Math.pow(a2, 2)-4*Math.pow(a1, 2)*a2)*a3+Math.pow(a2, 4)-2*a1*Math.pow(a2, 3)+2*Math.pow(a1, 2)*Math.
        pow(a2, 2))*Math.pow(b1, 2)+(Math.pow(a2, 2)-2*a1*a2+Math.pow(a1, 2))*Math.pow(a3, 4)
        +((-2*Math.pow(a2, 3))+2*a1*Math.pow(a2, 2)+2*Math.pow(a1, 2)*a2-2*Math.pow(a1, 3))*Math.pow(a3, 3)
        +(Math.pow(a2, 4)+2*a1*Math.pow(a2, 3)-6*Math.pow(a1, 2)*Math.pow(a2, 2)+2*Math.pow(a1, 3)*a2+Math.pow(a1, 4)
        )*Math.pow(a3, 2)
        +((-2*a1*Math.pow(a2, 4))+2*Math.pow(a1, 2)*Math.pow(a2, 3)+2*Math.pow(a1, 3)*Math.pow(a2, 2)-2*Math.pow(a1,
        4)*a2)*a3+Math.pow(a1, 2)*Math.pow(a2, 4)
        -2*Math.pow(a1, 3)*Math.pow(a2, 3)+Math.pow(a1, 4)*Math.pow(a2, 2))/((2*a2-2*a1)*b3+(2*a1-2*a3)*b2+(2*a3-2*a2
        )*b1);
}
}

```

References

- [1] Michael P Hitchman. *Geometry with an introduction to cosmic topology*. Jones & Bartlett Learning, 2009.
- [2] Frank Nielsen and Richard Nock. Hyperbolic Voronoi diagrams made easy. In *2010 International Conference on Computational Science and Its Applications*, pages 74–80. IEEE, 2010.
- [3] Frank Nielsen and Richard Nock. The hyperbolic Voronoi diagram in arbitrary dimension. *arXiv preprint arXiv:1210.8234*, 2012.
- [4] Frederic Sala, Chris De Sa, Albert Gu, and Christopher Ré. Representation tradeoffs for hyperbolic embeddings. In *International conference on machine learning*, pages 4460–4469. PMLR, 2018.
- [5] Toshihiro Tanuma, Hiroshi Imai, and Sonoko Moriyama. Revisiting hyperbolic Voronoi diagrams from theoretical, applied and generalized viewpoints. In *2010 International Symposium on Voronoi Diagrams in Science and Engineering*, pages 23–32. IEEE, 2010.