

Minimum Constructability for the Apollonian Circle

Gabriel Eiseman

DIMACS at Rutgers New Brunswick

June 13 2018

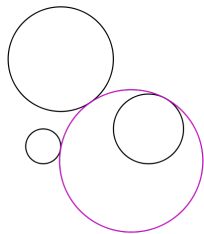
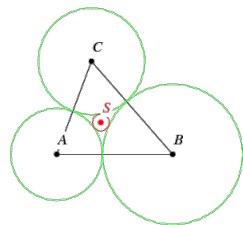
DIMACS

*Center for Discrete Mathematics & Theoretical Computer Science
Founded as a National Science Foundation Science and
Technology Center*



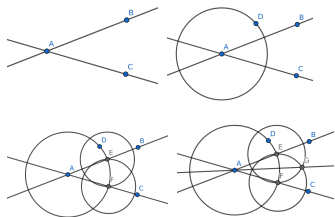
What are Circles?

and more pertinently what is an Apollonian circle?



- For any three tangent circles, there are exactly two circles tangent to all three of them, one inside and one outside.
- These are called the Apollonian circles after Apollonius who considered how mutually tangent circles could be constructed by compass and straightedge.
- We can also think of this circle and its center as things we construct from the triangle between the centers of the three circles.

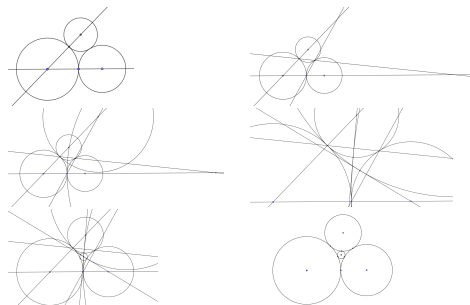
Measuring Efficiency of Constructions



- For compass and straightedge geometry, there are many ways we could measure the efficiency of a construction.
- The simplest and the one most commonly used is the number of lines and circles drawn. Drawing points is not counted.

The Most Efficient Known Solution

Baragar and Kontorovitch discovered a construction for the Apollonian circle that takes 7 steps.



- Draw two lines through the points of tangency besides the one not on a line.
- Draw a circle around the intersection of each of these lines with the original lines and through the points of tangency.
- Draw a diameter of each original circle through that circle's intersection with the new circle.
- Draw the third circle.

This is conjectured to be optimal. 🔍 ↻

Is this Construction Optimal?

The problem I explored is whether or not this construction is optimal. The simplest way to show it is optimal is to try every construction with 6 steps and show none of them work. How many of these could there be?

Is this Construction Optimal?

An upper bound on the number of constructions

- Initially, there are 10 points and 5 geometric objects.
- After each step, we create 1 geometric object and at most 2 times the current number of geometric objects points.
- After n steps we have at most $n(n - 1) + 10n + 10$ points, and there are at most $3 \binom{p}{2} - g$ choices for objects to draw at step n with p points and g objects.
- So there are at most 18,038,147,914,226,419,200 possible constructions with 6 steps.

Reducing the Number of Constructions to Explore

18 quintillion is hardly a tractable number of constructions to explore.

- However, we know that the last step has to be drawing the circle, which requires its center and a point on it to exist, so these must exist after step 5.
- For its center point to exist at step 5, we need something through the center after step 4, so we can throw out the vast majority of constructions after 4 steps, and there are only 300 million 4 step constructions, which is few enough that a computer can probably brute force it.

Towards a Computer Assisted Search

```
if(depth == 3){
    dfs_cons__2(base, candidate_cb, cb_data);
    return;
}
for(int i = 0; i < base->points_len - 1; ++i){
    const Point *restrict a = base->points + i;
    for(int j = i + 1; j < base->points_len; ++j){
        const Point *restrict b = base->points + j;
        Line l;
        construct_line(&l, a, b);
        ApproxCons next = *base;
        if(add_line_single(&next, &l)){
            remove_duplicate_points(&next, base->points_len);
            record_step(&next, i, j, GEOM_LINE);
            dfs_cons(&next, candidate_cb, cb_data, depth - 1);
        }
        Circle c;
        construct_circle(&c, a, b);
        next = *base;
        if(add_circle_single(&next, &c)){
            remove_duplicate_points(&next, base->points_len);
            record_step(&next, i, j, GEOM_CIRCLE);
            dfs_cons(&next, candidate_cb, cb_data, depth - 1);
        }
        construct_circle(&c, b, a);
        next = *base;
        if(add_circle_single(&next, &c)){
            remove_duplicate_points(&next, base->points_len);
            record_step(&next, j, i, GEOM_CIRCLE);
            dfs_cons(&next, candidate_cb, cb_data, depth - 1);
        }
    }
}
```

This is some very real code from my program.

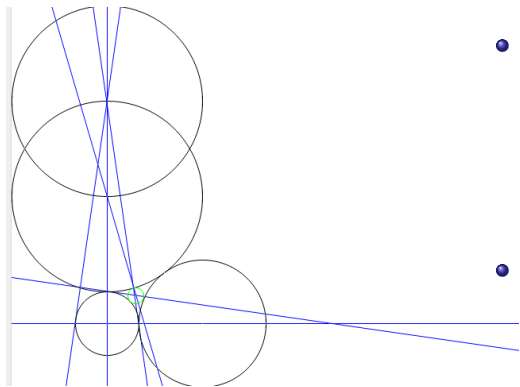
It would be hard to test if constructions work in general by computer, but if a construction works in general it must work for every particular triangle. Also, we can just use floating point math on the computer and not exact computation, as long as we make sure the error tolerance is high enough we can only get false positives not false negatives.

Towards a Computer Assisted Search

Living with Inexact Computations

- Floating point numbers on a computer are represented as a 53 bit integer times a (possibly negative) power of two.
- This means even for correctly rounded floating point operations like addition, multiplication, and square root, we can get a relative error up to 2^{-53} .
- Some operations like inverse sine have larger maximum error and if we compose all the errors from constructing and intersecting geometric objects for 5 steps we can get an upper bound on the error and use it to determine the error tolerance we need to use to avoid false negatives.

Computer Assisted Search Results



- My computer search found 451929 6 step constructions that worked for the 3-4-5 triangle I used to generate candidate solutions. This took 40 minutes.
- 10884 of these also worked for the 5-12-13 triangle, but NONE also worked for the 40-13-37 triangle!

So we're done right?

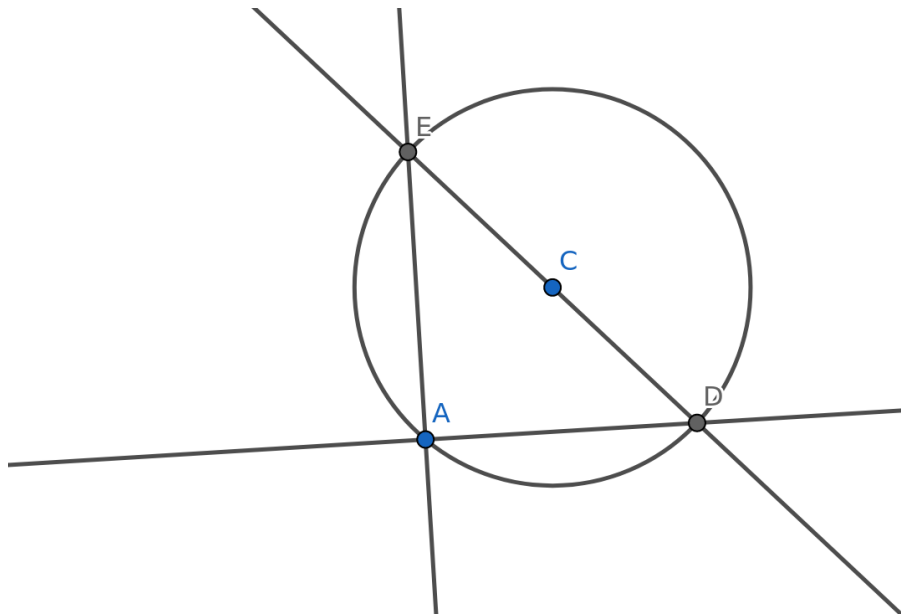
I exhaustively searched all possible six step constructions and none of them worked, so I tried to figure out ways it could be wrong.

- I considered only unbranched constructions that didn't use arbitrary points, but this is justified.
- If we used a construction where we make a decision based on how many intersections some parts of the geometry have, one of those possibilities would not generate the points we need and would effectively require a five step solution for that branch.
- But any five step solution would have been found in our six step search with a useless step.
- As for arbitrary points, we don't really need these because we want to construct something that is fully constrained.

Pop Quiz



Pop Quiz



So How Bad are the Arbitrary Points?

- Relative to one line or circle, there are only two regions of the plane up to isometries of that line or circle: on and off!
- If we consider the regions relative to all pairs of lines or circles, then we need to be able to pick a point in any region delimited by all the lines and circles.

So How Bad are the Arbitrary Points?

Really bad

This is really bad, because it makes the search space about 1000 times bigger but also requires a lot more computation per step, so the computation would now take much longer than 40,000 hours.

Where are we now?

- Many of my results from the simpler time before arbitrary points are still valid, such as showing 3 steps is optimal for an equilateral triangle, and there are 5 and 6 step constructions for isosceles and right triangles respectively.
- However, I still don't know for sure that there are no general 6 step constructions.
- If having arbitrary points in every region delimited by one line or circle is sufficient then my computation is still valid, but although I have found many restrictions on what arbitrary points are necessary I haven't been able to get a result that strong yet.
- I've mostly updated the computer search to handle arbitrary points, but efficiency is now more important than ever.

Some pictures were retrieved from Wolfram Math World and Wikipedia, and others were generated by Geogebra and my own compass and straightedge construction rendering program. Code is available at github.com/hacatu/apollonian_constructability Special thanks to the Rutgers Math Department, Professor Alex Kontorovich, and the National Science Foundation (grant DMS-1802119).