Measuring cocircularity in a point set

Andrea de las Heras[*1], Guillermo Esteban[†2, 3], Delia Garijo[‡4], Clemens Huemer[§1], Antoni Lozano[¶1], Nicolau Oliver[‖1], and David Orden[**3]

[1]Universitat Politècnica de Catalunya, Barcelona, Spain
[2]Carleton University, Ottawa, Canada
[3]Universidad de Alcalá, Alcalá de Henares, Spain
[4]Universidad de Sevilla, Sevilla, Spain

Abstract

In a given set S of n points in the plane, how close are four points of S to be cocircular? We define several measures to study this question, and present bounds on this almost-cocircularity in a point set. Algorithms for cocircularity are presented as well.

1 Introduction

A set S of points in the plane is in general position if no three points of S are collinear and no four points are cocircular. Most algorithms in Computational Geometry require the input points to be in general position. This simplifies the design of the algorithms as most degenerate situations arise from collinearity, but also from cocircularity. It is well known that any sufficiently large set of points contains three points that are almost collinear. In particular, a result by Erdős and Sekeres [1] states that for every set S of 2^n points in the plane, the largest angle defined by points of S is bounded from below by $\pi \cdot (1 - 1/n)$.

We study how close are four points from S to being cocircular. We define several measures of cocircularity in point sets and give bounds on these measures.

On the algorithmic side, the minimum area triangle defined by points of a given set S, which can be considered a measure of collinearity, can be found in $O(n^2)$ time using duality [3, 4]. Our goal is to design algorithms to find the tuple of four points of S closest to cocircularity. We present several $O(n^3)$-time algorithms for this problem. A related, and well studied, algorithmic problem to our research is computing the annulus of smallest width that contains S, see e.g. [5] and references therein.
In Section 2 we define three measures for cocircularity, the *Thales* cocircularity, the *Voronoi* cocircularity, and the *Determinant* cocircularity. We present properties and differences among them. In Section 3 we show bounds on the Thales cocircularity, and Section 4 is mainly devoted to the design of algorithms.

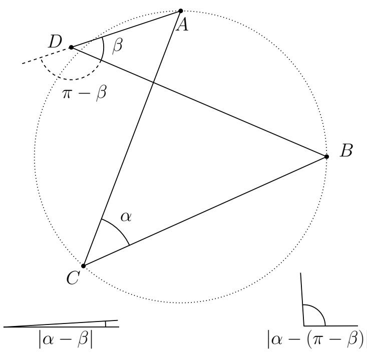

Figure 1: Angles used in Thales cocircularity.

2 Measures of cocircularity

Definition 1 *The Thales cocircularity of four points A, B, C, D is $\mathcal{T}(A,B,C,D) = \min_P\{\min\{|\alpha-\beta|, |\alpha-(\pi-\beta)|\}\}$, where $\alpha = \angle ACB$ and $\beta = \angle ADB$, and the minimum $\min_P$ is taken over all permutations P of the four points A, B, C, D. See Figure 1.*

The Thales cocircularity is motivated by Thales' theorem, also known as the inscribed angle theorem. $\mathcal{T}(A,B,C,D)$ is invariant under translation and scaling.

Definition 2 *The Determinant cocircularity $\mathcal{D}(A,B,C,D)$ of four points $A = (A_x, A_y)$, $B = (B_x, B_y)$, $C = (C_x, C_y)$, $D = (D_x, D_y)$ is the absolute value of the determinant:*

$$\begin{vmatrix} A_x & A_y & A_x^2 + A_y^2 & 1 \\ B_x & B_y & B_x^2 + B_y^2 & 1 \\ C_x & C_y & C_x^2 + C_y^2 & 1 \\ D_x & D_y & D_x^2 + D_y^2 & 1 \end{vmatrix}$$

*Email: andrea.de.las.heras@estudiantat.upc.edu.
†Email: g.esteban@uah.es.
‡Email: dgarijo@us.es.
§Email: clemens.huemer@upc.edu.
¶Email: antoni.lozano@upc.edu.
‖Email: nicolau.oliver@estudiantat.upc.edu.
**Email: david.orden@uah.es.

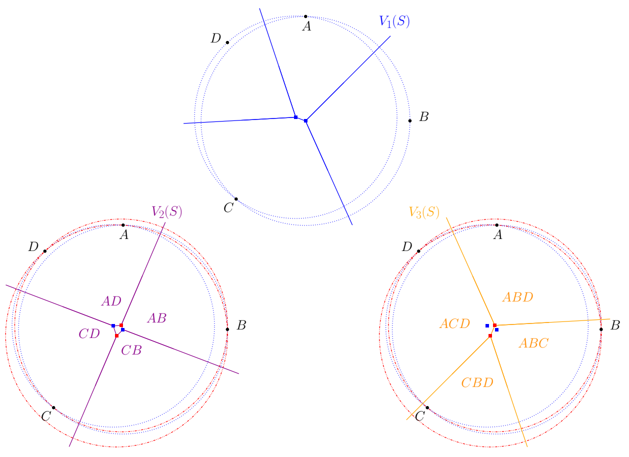

Figure 2: Top: $V_1(S)$ of a set of four almost cocircular points $S = \{A, B, C, D\}$. Bottom-left: $V_2(S)$, where for each unbounded cell the closest pair of points of S is indicated, and the bounded cell has AC as the closest pair of points. Bottom-right: $V_3(S)$, where in each cell the closest triplet of points of S is indicated. Centers of circles are depicted as red or blue squares.

The Determinant cocircularity is a very common tool to check if four points are cocircular: four points A, B, C, D in the plane lie on a common circle if and only if $\mathcal{D}(A, B, C, D) = 0$. More general, a determinant test is often used to check if $d+1$ points in $\mathbb{R}^d$ are in general position, since the volume the simplex defined by $d+1$ points is given by a determinant. $\mathcal{D}(A, B, C, D)$ is invariant under translation, but under a scaling by a factor c, the determinant varies in a factor of c^4. We therefore only consider the Determinant cocircularity in Section 4 on algorithms to show a relation to the 4-SUM problem.

We now introduce another measure of cocircularity. The order-k Voronoi diagram of a point set S, denoted by $V_k(S)$, is a partition of the plane into cells that have the same k closest points of S. The order-1 Voronoi diagram of four cocircular points is composed of one vertex of degree four and four rays from it, the vertex being the center of the circle. If we perturb the points slightly so that the cocircularity disappears, the Voronoi diagram changes: the vertex of degree four gets replaced by two vertices of degree three, connected by a short segment (there are two rays from each of them). Each vertex is the center of a circle through three of the four points considered, with none of them in the interior. See Figure 2, top. Note that there can be shorter segments in $V_2(A, B, C, D)$ or in $V_3(A, B, C, D)$, see Figure 2, bottom.

Definition 3 *The* Voronoi cocircularity *of four points A, B, C, D, denoted by $\mathcal{V}(A, B, C, D)$, is the length of the shortest edge in all $V_k(A, B, C, D)$ for $k = 1, 2, 3$. $\mathcal{V}(A, B, C, D)$ is zero if some $V_k(A, B, C, D)$ has a vertex of degree four.*

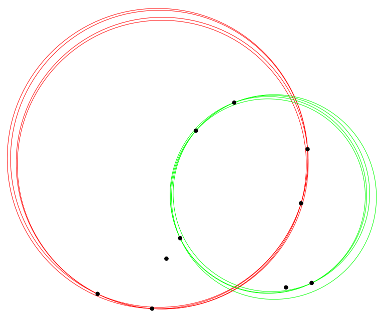

Figure 3: Green (resp., red) circles pass through the four most cocircular points of a set S of $n = 10$ points according to Voronoi (resp., Thales) measure.

$\mathcal{V}(A, B, C, D)$ is invariant under translation. Also, under scaling by a factor c, the length of the shortest Voronoi edge scales linearly with c.

Definition 4 *The* Thales/Determinant/Voronoi cocircularity *of a set S of points is the minimum of the Thales/Determinant/Voronoi cocircularity among all 4-tuples of points of S.*

2.1 Properties and differences

All the measures described in the previous section are zero when the points are cocircular.

We next give a formula for the length of an edge in a Voronoi diagram of S. Each endpoint of such an edge is the center of a circle passing through three points of S, and where two of these three points are the same for both circles.

Proposition 5 *Let A, B, C, D be four points in the plane, and let C_A, C_B be the centers of the circles through C, A, D and C, B, D, respectively. If C_A and C_B are on the same side of the line $\overline{CD}$, then*

$$|C_A C_B| = \frac{|CD|}{2}\big(\big||\cot\beta| - |\cot\alpha|\big|\big),$$

where α and β are the angles $\angle CAD$ and $\angle CBD$, respectively. Otherwise,

$$|C_A C_B| = \frac{|CD|}{2}(|\cot\beta| + |\cot\alpha|).$$

We show in Figure 3 that the measures of Voronoi and Thales cocircularity of a point set are different, in the sense that they do not yield the same four-tuple of points minimizing the cocircularity measure.

3 Bounds on cocircularity

We present two bounds on the measure of Thales cocircularity.

Algorithm 1 Detecting cocircularities with inversions

```
for every point $p_i \in S$ do
    • Invert all points of $S \backslash \{p_i\}$ with respect to the center $p_i$ and radius $r = 1$.
    • Execute as subroutine the sweep-line algorithm to detect collinearities, see [3].
end for
```

Proposition 6 *For any set S of n points and any two points $A, B \in S$ there exist $C, D \in S$ such that $\mathcal{T}(A, B, C, D) \leq \pi/(n-3)$.*

Theorem 7 *For any set S of n points in convex position, there exist $A, B, C, D \in S$ such that $\mathcal{T}(A, B, C, D) \in O\left(\frac{1}{n^2}\right)$.*

4 Algorithms

Another line of research is to detect these cocircularities. The decision problem is: Are there four cocircular points in S? The optimization problem is: Find the 4-tuple of points of S that minimizes the measure.

4.1 Inversions

Proposition 8 *There is an $O(n^3)$-time algorithm that decides whether there exist four cocircular points in a set of n points, using inversions.*

An *inversion* transformation is determined by two parameters: The center O and the radius R of inversion. Two points P, P' are *i*nverses if they lie in the same half-line with origin at O and the Euclidean distances $|OP|$ and $|OP'|$ satisfy $|OP| \cdot |OP'| = R^2$. We need the following property of inversions:

Property 9 *Given a center O and a radius R of inversion, any circle containing O is inverted into a line.*

Thus, if we perform an inversion at a point A that is cocircular with points B, C, D, then inverting B, C, D results in three points being collinear. Property 9 allows us to propose Algorithm 1. The cost of the subroutine is $O(n^2)$, and it is executed in a loop with $O(n)$ iterations. Thus, the total cost is $O(n^3)$ time and $O(n)$ space.

We note that inversions do not preserve a relationship between the measures of cocircularity and collinearity. Then, by using inversions, we can only solve the decision problem of detecting cocircularities.

4.2 Higher order Voronoi diagrams

Proposition 10 *The Voronoi cocircularity of a set S of n points in general position in the plane can be computed in $O(n^3)$ time.*

To compute the Voronoi cocircularity of S is equivalent to finding the shortest edge among the diagrams $V_k(S)$, for $k = 1, \ldots, n-1$. All diagrams $V_k(S)$ can be obtained in time $O(n^3)$ [4, 6], and the number of edges of a diagram $V_k(S)$ is at most $O(k(n-k))$. Hence, the computation can be done in $O(n^3)$ time.

4.3 Reduction from 4-SUM

The k-SUM problem asks if a list of n integers contains k integers whose sum is zero. This is a prominent problem for $k = 3$ since there is a large list of problems, called 3SUM-hard, that have been proved to be as difficult as 3-SUM; among them, the problem to decide if three points of a given set are collinear [2].

There are easy quadratic-time algorithms to solve both the 3-SUM and the 4-SUM problem in the integer RAM model. As the quadratic-time algorithm for the 4-SUM problem uses hashing, under the real RAM model there is a similar algorithm without hashing of complexity $O(n^2 \log n)$. We prove the following result, which has as a consequence that the problem to decide whether four points are cocircular is 4SUM-hard.

Proposition 11 *Let $[x_1, x_2, \ldots, x_n]$ be a list of n integers, and let S be the set of n points on the parabola $y = x^2$, with coordinates (x_i, x_i^2). Then, S contains four cocircular points if and only if the sum of the x-coordinates of these four points is zero.*

4.4 Cost of the exact problem

We show that to decide whether there exist four cocircular points in a set of n points can be done in $\mathrm{O}(n^3 \log n)$ time in the worst case or, using hashing, in expected $\mathrm{O}(n^3)$ time[1]. Both algorithms can find a solution, thus solving the optimization problem, and work by storing the radius and center of the circle defined by each triplet of points, and then detecting collisions —in the first case, by sorting the circles, which contributes with the additional factor of $\log n$.

Proposition 12 *There is an algorithm working in expected $O(n^3)$ time that decides whether there exist four cocircular points in a set of n points.*

Proof. Suppose S is a set of n points. Our algorithm uses hashing with separate chaining. In particular, H represents a hash table of point sets indexed by triples of points. Let f be the function that, given a set T of three points, returns the triple (x, y, r) s.t. (x, y) and r are the center and radius, resp., of the circle defined by the points in T. We will use the operations:

- **Insertion**: $H.insert(T)$ adds information T with key $f(T)$ to H.

[1]The use of hashing was also proposed in `https://cs.stackexchange.com/questions/49316/largest-set-of-cocircular-points`

Algorithm 2 Exact cocircularity with hashing

```
given a set $S$ of $n$ points
let $H$ be a hash table of size $n^3$
initialise $H$ with all entries containing $\emptyset$
for every distinct $T \subseteq S$ with $|T| = 3$ do
    if $H[T] = \emptyset$ then
        $H[T] \leftarrow T$
    else
        return $H[T] \cup T$
    end if
end for
return "no cocircular points"
```

- **Search**: *H.search*(T) returns $\emptyset$ if H does not contain information for key $f(T)$ or a set R s.t. $f(T) = f(R)$, otherwise.

The method is shown in Algorithm 2. Since insertion and search have constant average cost, the for loop and the initialization of H give an $\mathrm{O}(n^3)$ expected cost for the whole algorithm. □

Proposition 13 *There is an $O(n^3 \log n)$-time algorithm that decides whether there exist four cocircular points in a set of n points.*

Proof. In Algorithm 3, the hash table of Algorithm 2 has been substituted by a vector V of size n containing 4-tuples (x, y, r, T), where n is the number of points; here T represents a set of three points and (x, y) and r are the center and radius resp. of the circle defined by the points in T. We use a stable sorting algorithm (like *mergesort*) and apply it to V with respect to the first, second, and finally third component of the vector. Since the sorting algorithm is stable, after the three iterations the entries corresponding to the same circle are contiguous in V and can be detected in linear time. The cost of the first and third for loops is $\mathrm{O}(n^3)$, while the second for loop has a cost $\mathrm{O}(n^3 \log n^3) = \mathrm{O}(n^3 \log n)$, which is also the cost of Algorithm 3 in the worst case. □

5 Conclusions

We initiated the study of almost cocircularity in point sets. We chose measures of cocircularity that we considered to be very natural, though other measures could be studied as well. Several questions remain open, and we plan to continue this line of research.

Open problem 14 *Can the decision problem of detecting cocircularities be solved in sub-cubic time?*

Open problem 15 *Can the bound for the Thales measure for convex point sets be extended to arbitrary point sets in general position?*

Algorithm 3 Exact cocircularity

```
given a set $S$ of $n$ points
let $V$ be a size $n^3$ vector of tuples $(x, y, r, T)$
  for reals $x$, $y$, $r$ and a three point set $T$
$i \leftarrow 1$
for every distinct $T \subseteq S$ with $|T| = 3$ do
    let $(x, y)$ and $r$ be the center and radius, resp.,
      of the circle defined by the points in $T$
    $V[i] \leftarrow (x, y, r, T)$
    $i \leftarrow i + 1$
end for
for $i \in \{1, 2, 3\}$ do
    sort $V$ with respect to component $i$
end for
for $i = 1$ to $n^3 - 1$ do
    if $V[i][j] = V[i+1][j]$ for each $j \in \{1, 2, 3\}$ then
        return $V[i][4] \cup V[i+1][4]$
    end if
end for
return "no cocircular points"
```

Open problem 16 *Find families of point sets that are "far away" from having four cocircular points.*

Acknowledgments. This work was initiated at the TOPPING workshop in Barcelona, July 2022. We thank Mercè Claverol, Dolores Lara, Alejandra Martínez, and Pablo Perez for their valuable discussions and contributions. The authors were supported by projects PID2019-104129GB-I00/MCIN/AEI/10.13039/501100011033 and Gen.Cat. 2021 SGR 00266.

References

[1] P. Erdős, G. Szekeres, On some extremum problems in elementary geometry, *Ann. Univ. Sci. Budapest* **3-4** (1960/1961), 53–62.

[2] A. Gajentaan, M. H. Overmars, On a class of $O(n^2)$ problems in computational geometry. *Comput. Geom. Theory Appl.* **5**(3) (1995), 165–185.

[3] H. Edelsbrunner, L. J. Guibas, Topologically sweeping an arrangement, in: *Proc. eighteenth Annual ACM Symposium on Theory of Computing*, 1986, 389–403.

[4] H. Edelsbrunner, J. O'Rourke, R. Seidel, Constructing arrangements of lines and hyperplanes with applications, *SIAM J. on Comput.*, **15**(2), 1986, 341–363.

[5] J. García-López, P. A. Ramos, J. Snoeyink, Fitting a set of points by a circle, *Discrete Comput. Geom.* **20** (1998), 389–402.

[6] K. Mulmuley, On levels in arrangements and Voronoi diagrams, *Discrete Comput. Geom.* **6** (1991), 307–338.