

## Article

# Discovering Geometric Inequalities: The Concourse of GeoGebra Discovery, Dynamic Coloring and Maple Tools

Tomás Recio <sup>1,\*</sup>, Rafael Losada <sup>2,†</sup>, Zoltán Kovács <sup>3,†</sup> and Carlos Ueno <sup>4,†</sup>

<sup>1</sup> Departamento de Ingeniería Industrial, Escuela Politécnica Superior, Universidad Antonio de Nebrija, C/Santa Cruz de Marcenado 27, 28015 Madrid, Spain

<sup>2</sup> Sociedad Asturiana de Educación Matemática “Agustín de Pedrayes”, Federación Española de Sociedades de Profesores de Matemáticas, Plaza Club Patín Gijón Solimar 1, 33213 Gijón, Spain; rafael.losada@gmail.com

<sup>3</sup> The Private University College of Education of the Diocese of Linz, Salesianumweg 3, 4020 Linz, Austria; zoltan@geogebra.org

<sup>4</sup> CEAD Profesor Félix Pérez Parrilla, C/Dr. García Castrillo, 22, 35005 Las Palmas de Gran Canaria, Spain; cuenjac@cead-laspalmas.net

\* Correspondence: trecio@nebrija.es

† These authors contributed equally to this work.

**Abstract:** Recently developed GeoGebra tools for the automated deduction and discovery of geometric statements combine in a unique way computational (real and complex) algebraic geometry algorithms and graphic features for the introduction and visualization of geometric statements. In our paper we will explore the capabilities and limitations of these new tools, through the case study of a classic geometric inequality, showing how to overcome, by means of a double approach, the difficulties that might arise attempting to ‘discover’ it automatically. On the one hand, through the introduction of the dynamic color scanning method, which allows to visualize on GeoGebra the set of real solutions of a given equation and to shed light on its geometry. On the other hand, via a symbolic computation approach which currently requires the (tricky) use of a variety of real geometry concepts (determining the real roots of a bivariate polynomial  $p(x, y)$  by reducing it to a univariate case through discriminants and Sturm sequences, etc.), which leads to a complete resolution of the initial problem. As the algorithmic basis for both instruments (scanning, real solving) are already internally available in GeoGebra (e.g., via the Tarski package), we conclude proposing the development and merging of such features in the future progress of GeoGebra automated reasoning tools.

**Keywords:** automated theorem proving in geometry; automated deduction in geometry; automated reasoning in geometry; Dynamic Geometry; GeoGebra; computational algebraic geometry



**Citation:** Recio, T.; Losada, R.; Kovács, Z.; Ueno, C. Discovering Geometric Inequalities: The Concourse of GeoGebra Discovery, Dynamic Coloring and Maple Tools. *Mathematics* **2021**, *9*, 2548. <https://doi.org/10.3390/math9202548>

Academic Editor: Juana Sendra

Received: 4 September 2021

Accepted: 6 October 2021

Published: 11 October 2021

**Publisher’s Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

In recent years we are witnessing a rich interaction between two geometric instruments which were initially born with different intentions in mind, but are converging and joining forces nowadays to create a new generation of software for doing Geometry. On the one hand there were the Geometric Automatic Theorem Provers (GATPs), which can be traced back to the seminal work by Wu [1], with an increasing variety of methods and approaches developed along the years, such as the Gröbner Basis Method, the Area Method, the Full-Angle Method. . . . These methods have succeeded to automatically prove a large collection of geometric statements, as can be seen for instance [2].

On the other hand, there were the Dynamic Geometry Software (DGS) programs (some of the pioneers being *The Geometric Supposer*, *Cabri*, *The Geometer’s Sketchpad*, *Geometry Inventor* or *Thales*) which initially were conceived as a tool for teaching Geometry in schools, as a kind of digital assistant for the constructions traditionally made by ruler and compass on the blackboard.

The GATPs based on algebraic techniques, such as Wu’s Method ([1,3]) or the Gröbner Basis method ([4,5]), require frequently the manipulation of a large set of polynomials,

which represents a given geometric construction. Unfortunately, this translation of geometric properties into algebraic equations can be a tedious process. Dynamic Geometry programs turned out to be an ideal interface to introduce geometric information, and the extension of this software with convenient algebraic tools allowed the creation of a natural environment to work on automated proving in Geometry. This fruitful cooperation was put into practice in several DGS programs such as *Geometry Expert*, *GDI* or *GCLC*, with *GeoGebra* joining this list in recent years. Since its inception, GeoGebra put special emphasis on becoming something more than a DGS, but a Mathematical Software where not only the geometrical objects, but their algebraic counterparts, could be accessible and visualized simultaneously. This, together with its increasing worldwide popularity ([6–8]) and free access for non-commercial use, makes GeoGebra a top choice for carrying out this integration process with algebraic proving tools.

At present, GeoGebra is actively developing a set of new Geometric Automated Reasoning Tools (GARTs) which not only are capable of proving geometric equalities or inequalities, but allow the user to *explore* and *discover* facts about geometric figures. The present work is aimed at showing some of these new tools, implemented in recent versions of GeoGebra Discovery (an experimental version of GeoGebra), emphasizing their ability to handle geometric inequalities. We also comment on its current strengths and limitations, and how the visualization possibilities that offer the graphic view of GeoGebra can illuminate the automatic resolution of geometric problems which involve certain difficulties, due to the high complexity of some elements appearing along the process (see [9] for an unpublished paper with a similar approach). To that end, we will focus our attention on *re-discovering* and proving by computational means the geometric inequality 5.3 of the collection [10]:

**Proposition 1.** *If the lengths of the sides of a triangle are  $a$ ,  $b$ ,  $c$ , and the length of its circumradius is  $R$ , then*

$$a + b + c \leq 3\sqrt{3} R. \quad (1)$$

*Equality holds if and only if  $a = b = c$ .*

In summary, our research goal is to show how to approach this Proposition through the combination of a dynamic color scanning method and (real algebra) symbolic computations, exemplifying the need to automatize this combination of tools, to address and to solve some automated reasoning challenging issues that arise in the current *GeoGebra Discovery* version.

In Section 2, we briefly describe the tools that are already available in the official releases of GeoGebra, and which allow proving and discovering geometric equalities in a given construction. In Section 3, we present new tools for dealing with geometric inequalities, currently implemented in *GeoGebra Discovery*, an experimental version of GeoGebra maintained by the first author where GARTs are first checked before official release. In Section 4, we introduce the *Dynamic Color Scanning Method*, a powerful and flexible visualization technique devised by the second author on GeoGebra that can be helpful when symbolic computations or other visualization methods fail. To illustrate how the previous tools can be employed and combined, in Section 5, we approach the task of automatically proving the classical geometric inequality in Proposition 1, highlighting the accomplishments that the new symbolic tools in GeoGebra Discovery achieve when investigating geometric relations and inequalities, but commenting also on the current limitations that hinder a full resolution of the problem. Section 6 combines the visualization power of the dynamic color scanning method, together with the algebraic capabilities of the software Maple to definitely settle Proposition 1. Finally, in Section 7, we include some conclusions and propose lines of research to improve the performance and versatility of GeoGebra in future versions.

## 2. The GeoGebra Automated Reasoning Tools

Since 2014, GeoGebra is making a systematic and permanent effort in order to develop new Geometric Automated Reasoning Tools (GARTs) based on algebraic computations that go beyond proving geometric statements, providing better feedback than just deciding the truth or falsehood of a statement, and introducing instruments that actually lead to the discovery of geometric facts (see [11,12], for some papers both dealing with the algebraic algorithms and their implementation in GeoGebra). Given the school environment where DGS programs are mainly utilized, these new features are aimed at providing students with a kind of expert companion, a Geometer Automata (GA) as described in [13], to help them explore geometric configurations and gain insight into the relations that exist among its elements.

Currently, the CAS engine embedded in GeoGebra that performs the symbolic computations required to use these reasoning tools is Giac ([14,15]), which has support for symbolic computations such as finding the Gröbner basis of a polynomial ideal or determining whether a polynomial belongs or not to a given ideal. At present, the GARTs in GeoGebra make use of Giac in order to establish equality relations among the elements of a geometric construction. In the next section, we will present new developments that expand the possibilities of these tools.

To give a rough sketch of how things work, let us start by setting up the basic scenario. Let us assume that  $K$  (usually  $\mathbb{Q}$ ) is a field with an algebraically closed extension  $L$  (usually  $\mathbb{C}$ ). Roughly speaking, let us assume that we have a geometric construction and that we want to show whether a thesis concerning this construction always holds. The construction starts by placing some free points  $P_i(u_{2i-1}, u_{2i})$ ,  $i = 1, \dots, d$ , with their coordinates taken in the field  $K$ . From these points we can construct new elements (points, segments, lines, circles, etc.) that depend on the initial free points, eventually introducing new free variables in the construction. These elements satisfy geometric relations that can be expressed in form of polynomial equations (we often make an abuse of notation by identifying a polynomial  $p(x_1, \dots, x_n)$  with its corresponding polynomial equation  $p(x_1, \dots, x_n) = 0$  and/or its set of solutions)  $h_k(u_1, \dots, u_n) \in K[u_1, \dots, u_{2d}, u_{2d+1}, \dots, u_n]$ . Among the variables  $\{u_i\}$  we can choose a maximal set of independent variables (that is, which do not hold polynomial relations among them). Usually, this set of independent variables contains the coordinates of the initial free points, and its cardinal is related to the *degrees of freedom* of the configuration. As a simple example, let us consider the construction

*Take a point  $P$  in the perpendicular bisector of a segment  $AB$ .*

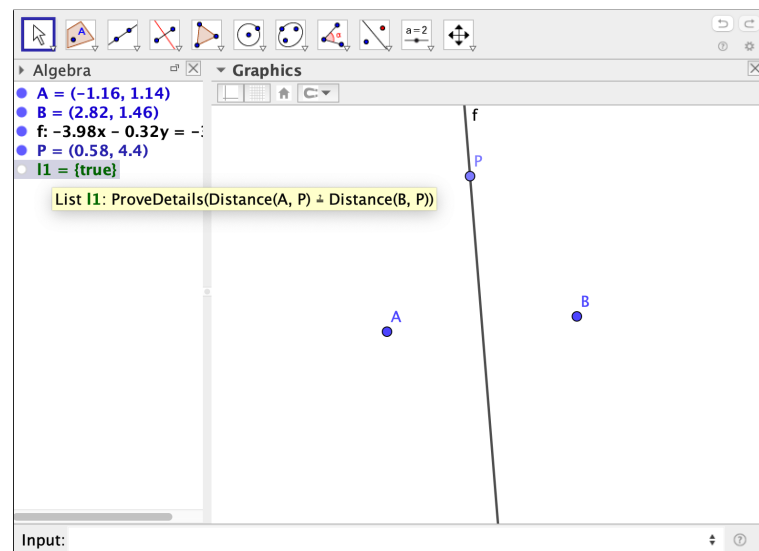
We require two free points  $A(u_1, u_2)$ ,  $B(u_3, u_4)$ , and a third point  $P(u_5, u_6)$  that must lie in the perpendicular bisector of the segment  $AB$ , whose equation is given by

$$H_1 := \left(u_5 - \frac{u_1 + u_3}{2}\right)(u_3 - u_1) + \left(u_6 - \frac{u_2 + u_4}{2}\right)(u_4 - u_2) = 0. \quad (2)$$

We are using here six variables, and there is one polynomial relation among them. Intuitively speaking, our construction has five degrees of freedom (four for the coordinates of the points  $A$ ,  $B$ , plus one more that decides the height of the point  $P$  on the bisector). Of course, for more complex constructions we would get not just one, but a collection of polynomials  $H = \{H_1, \dots, H_m\}$ . The points in  $L^n$  satisfying the polynomials in  $H$  form the *hypothesis variety*  $V(H)$ , and each of these points correspond to a particular instance of the figure under study.

Now, after having settled our construction we would like to prove something. Let us suppose we want to prove the following thesis (see Figure 1):

*The distances from  $P$  to  $A$  and  $B$  are equal.*



**Figure 1.** Proving with GeoGebra that the distances from a point  $P$  in the perpendicular bisector of segment  $AB$  to its extremes are equal.

This thesis can also be expressed as a polynomial:

$$T := (u_5 - u_1)^2 + (u_6 - u_2)^2 - (u_5 - u_3)^2 - (u_6 - u_4)^2 = 0, \quad (3)$$

and the set of its solutions in  $L^6$  is called the *thesis variety*  $V(T)$ . Now, the proposition

*If  $P$  is a point on the perpendicular bisector of the segment  $AB$ , then the distances from  $P$  to  $A$  and  $B$  are equal.*

Would be proved if we check that, for each possible configuration  $u := (u_1, \dots, u_6) \in L^6$  satisfying the relation  $H_1(u) = 0$ , then  $T(u) = 0$  also holds. In other words, if we check that  $\{H_1(u) = 0 \implies T(u) = 0\}$ . Using the language of algebraic geometry, this amounts to say that the hypothesis variety  $V(H)$  is contained in the thesis variety  $V(T)$ . Now, in this new algebraic setting, we can take advantage of all the machinery related to varieties, polynomial rings and ideals (with coefficients in the algebraically closed field  $L$ ). Since proving that  $V(H) \subset V(T)$  is equivalent to show that  $V(H) \cap (L^n \setminus V(T)) = \emptyset$ , by adding an extra variable  $t$  and considering all our polynomials in  $L[u_1, \dots, u_n, t]$  we are led to show that  $V(H) \cap V(Tt - 1) = \emptyset \subset L^{n+1}$ . By the Nullstellensatz Theorem, this is the case if the ideal  $\langle H \rangle + \langle Tt - 1 \rangle \subset L[u_1, \dots, u_n, t]$  is  $\langle 1 \rangle$ . However, checking that 1 belongs to an ideal given by a set of generators can be algorithmically decided by checking whether a Gröbner basis of the ideal reduces to  $\{1\}$  ([16], (Chapter 2)).

**Remark 1.** Notice that in the example above we actually have  $V(H_1) = V(T)$ , since  $T = 2H_1$ . In other words, the hypothesis and the thesis are equivalent.

Unfortunately, working in a Euclidean Geometry environment introduces certain complications in the description given above, due to several reasons:

- (i) *The use of complex versus real geometry tools:* Points in the Euclidean plane have real coordinates, while the algebraic algorithms used within this framework are designed for algebraically closed fields. Therefore, when we state that  $V(H) \subset V(T)$  we are including points in  $V(H)$  with complex coordinates, which is more than what we actually need. We could be missing valid theses which hold for all real points in  $V(H)$ , but not on the whole complex variety (see [17,18]). For example, let us consider the following statement, usually called Clough's conjecture [19], namely, the equality of

- the sum of segments  $l = EC$ ,  $m = FB$ , and  $n = GA$ , where  $E, F, G$  are the feet of the perpendiculars to the sides of an equilateral triangle from an arbitrary point  $D$ ,
- and  $3p/2$ , where  $p$  is the length of side  $AB$ , i.e.,  $3p/2$  is half the perimeter  $3p$  of the equilateral triangle  $ABC$ .

See [20] for detailed synthetic and algebraic proofs. Now, the algorithms we have implemented in GeoGebra, if asked about the Relation between  $l + m + n$  and  $3/2 \cdot p$ , proceed, first, checking the numerically approximate equality of both quantities and, if it holds, starting a symbolic protocol by considering the ideal generated by all the *polynomial equations* describing the hypotheses. We emphasize here the terms *polynomial equations*, because all the lengths  $l, m, n, p$  are described by expressing  $l^2, m^2, n^2, p^2$  as the sum of the squares of the differences of the coordinates of the extremes of the corresponding segments, say,  $l^2 = (c_1 - e_1)^2 + (c_2 - e_2)^2$ , etc. There are neither square roots nor choice of positive signs for defining  $l, m, n, p$ , as this approach would require real algebraic geometry methods. Thus, our hypotheses ideal includes much more cases (in the complex geometry context) than intuitively expected, and the thesis  $l + m + n = 3/2 \cdot p$ , holds only on some of them (for example, on those cases where  $l, m, n, p$  are positive).

Accordingly, *GeoGebra Discovery* declares that the formulated statement is “true on parts, false on parts”, see Figure 2. This means that the underlying algorithm detects, without having to compute a primary decomposition of the involved hypotheses ideal, that the statement holds true on some components and is false on some others, see item iii) below.

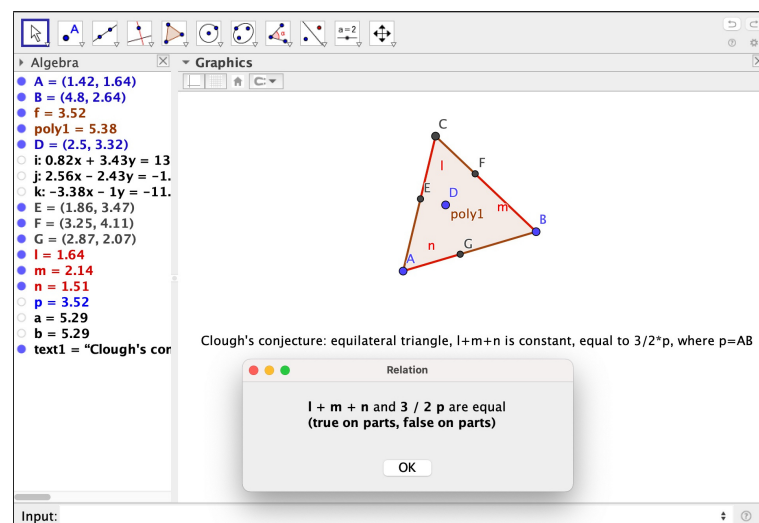
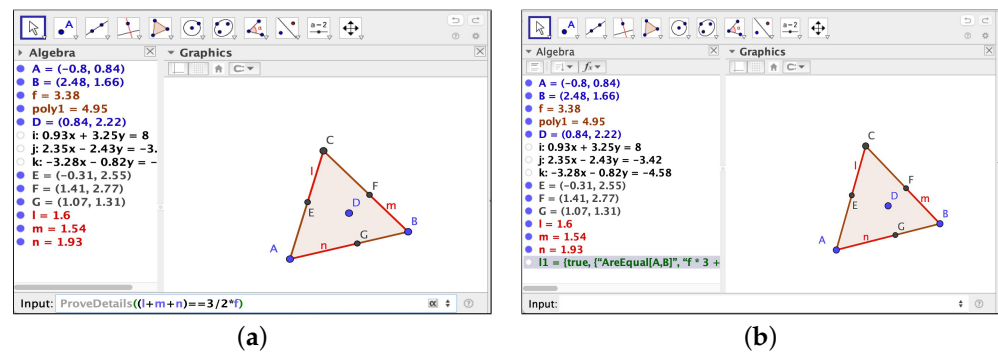


Figure 2. Checking Clough’s conjecture through GeoGebra Discovery yields “true on parts”

On the other hand, asking directly for the proof of Clough’s statement, see Figure 3, an affirmative answer is obtained, together with a list (too long in this case to be fully displayed in the Figure) of construction instances that should be avoided for the truth of the statement: degenerate cases such as when the triangle collapses and, also, variants of the thesis statement that hold just on the primary components of the hypotheses where the given thesis is false.

This is related to the possibility, in this case, to modify the thesis so that it collects all possible sign choices for  $l, m, n, p$ , by multiplying  $(l + m + n) - 3/2 \cdot p$  by all possible variants of such formula, changing signs for  $l, m, n, p$ , such as  $((l - m + n) - 3/2 \cdot p)$ ,  $((l - m + n) + 3/2 \cdot p)$ , etc. See [21] for a detailed study of this way of understanding real statements in a complex geometry context.





**Figure 3.** (a) Asking *GeoGebra Discovery* to prove that the sum  $l + m + n$  is equal to the semiperimeter  $3/2f$ . (b) *GeoGebra Discovery* replies that the statement is true except in some degenerate cases (described in the list after true in the last line of the Algebra window).

Of course, the reason for choosing a complex algebraic geometry approach for *GeoGebra* automated reasoning tools is because algorithms based on complex geometry are generally much more efficient than their real counterparts, and this aspect is of special relevance for the implementation of GARTs in software with educational purposes.

- (ii) *Non-degeneracy conditions*: In Geometry statements, whenever we mention objects such as line segments, polygons, circles..., we implicitly assume certain conditions on the points that are used to define those figures and render a theorem as *true*. As an example, in a theorem on triangles we always assume that its vertices are *non-collinear* points of the plane. These conditions are not easy to interpret by the GATPs and require special care when dealing with them. The problem of non-degeneracy conditions, which depends upon the choice of independent variables in the construction, the way the construction itself is carried out, or the interpretation we can give to what a degenerate construction is, has been studied extensively in the literature and the reader is referred to [12,22–25] or [26] to gain a better understanding of the intricacies surrounding it.
- (iii) *True on parts, false on parts statements*: As already explained in item i), quite often the hypothesis variety that is obtained from the algebraization process (via complex geometry algorithms which avoid the computationally costly decomposition of the hypothesis variety into irreducible components) can have irreducible, non-degenerate components of maximal dimension, such that the thesis holds on some of the components, while not on others. This situation was thoroughly treated in [27], and gave rise to a new category of *true on parts, false on parts* geometric statements.

However, our latest developments want to go further, providing tools that actually discover geometric theorems, not just prove them. Some of the strategies behind the algebraic approach to this endeavor (see also [12,24,28]) are briefly described below:

**Strategy 1.** We want to show the implication  $\{H \implies T\}$ . As seen before, we proceed to check that the ideal  $\langle H \rangle + \langle Tt - 1 \rangle = \langle 1 \rangle = K[u_1, \dots, u_n, t]$ . What if we do not get  $\langle 1 \rangle$  as response to our query? Let us assume that our CAS system produces an output such as this:

$$\langle H \rangle + \langle Tt - 1 \rangle = \langle g_1(u_1, \dots, u_n, t), \dots, g_l(u_1, \dots, u_n, t) \rangle =: J.$$

If we set  $J' = J \cap K[u_1, \dots, u_d] \neq 0$ , where the variables  $\{u_1, \dots, u_d\}$  form a maximal set of independent variables in  $\langle H \rangle$ , it is easy to see that the points in the variety  $V(J') \subset L^n$  cannot satisfy the thesis condition, and must be excluded from the hypothesis variety in order to obtain a valid statement, which would take the form

$$H \wedge (g_1 \neq 0 \vee \dots \vee g_l \neq 0) \implies T.$$

Of course, a new check should be made to confirm the validity and interest of this new formulation.

In other words, an algorithm has provided us *automatically* with the missing hypothesis that turn our initial statement into a *true* one. Depending on the form and complexity that the polynomials  $g_i$  take, the interpretation of these new conditions in geometric terms can be more or less affordable. In fact, some of them are closely related to the non-degeneracy conditions we already mentioned.

**Strategy 2.** We have obtained from a geometric construction our hypothesis ideal  $H$ , and we are curious about when a certain property holds among some of its elements. If we express this property through an algebraic relation  $f(u_1, \dots, u_n) = 0$ , then we can think of the ideal

$$J = \langle H \rangle + \langle f(u_1, \dots, u_n) \rangle,$$

and if the variables  $\{u_1, \dots, u_d\}$  form a maximal set of independent variables in  $\langle H \rangle$  we can eliminate the variables  $\{u_{d+1}, \dots, u_n\}$  to obtain an ideal

$$J' = (\langle H \rangle + \langle f(u_1, \dots, u_n) \rangle) \cap K[u_1, \dots, u_d]$$

that provides the conditions that the independent variables must fulfill in the construction in order to obtain the wanted property.

**Strategy 3.** Another possibility for implementing discovering tools which combine a numerical and a symbolic approach can proceed as follows: Given a geometric configuration, proceed firstly to search (by using fast performing numerical computations) for properties held among the elements of the construction (parallelism, concurrency, collinearity, concyclicity, ...) and set them as possible conjectures. In a second stage, these conjectures will be handled by the algebraic prover to decide whether they can be settled as true statements or rejected.

GeoGebra makes use of these techniques in several commands, some of them already implemented in the official releases of GeoGebra (for technical details concerning their inner workings, see [28,29]):

- **Relation(<list of objects>):** This higher level command performs two distinct operations: In the first place it proceeds to make a numerical comparison of the objects, and outputs an answer concerning geometric properties such as equality/inequality, parallelism, collinearity, concurrency, etc. Together with this numerical output, the user finds a "More..." button that, when clicked, gives way to a deeper, symbolic analysis of the numerical result previously obtained, resorting to the commands that mention below and proving or rejecting its truthfulness. See Figure 2.
- **Prove(<Boolean Expression>):** This command is the basic symbolic automatic theorem prover in GeoGebra, and its output can be (generally) true, (generally) false or undefined/? (when GeoGebra cannot decide the question). In order to introduce easily the Boolean expressions of geometric nature, GeoGebra offers a number of different commands such as **AreParallel(<line>, <line>)**, **AreCollinear(<Point>, <Point>, <Point>)**, **AreConcyclic(<Point>, <Point>, <Point>, <Point>)**, etc.
- **ProveDetails(<Boolean Expression>):** Improved version of the previous command, offering as output more information on degeneracy conditions whenever they are easy to translate into a simple statement. See Figure 3.
- **LocusEquation(<Arguments>):** This command offers the possibility of finding the algebraic equation for a geometric locus, obtained also by symbolic means. Besides, the command shows on the graphic window the curve corresponding to the equation. Among the several options for choosing the arguments, one which will be useful later to our purposes is **LocusEquation(<Boolean expression>, <Point>)**; this command will find the equation for the locus of points satisfying the given Boolean expression.
- **Discover(<Point>):** This command, one of the latest additions in GeoGebra Discovery, provides as output from a geometric construction a whole set of geometric statements concerning properties about the lines, segments and other geometric ele-

ments related to the point provided as argument. The inner workings of the command follow the idea exposed in Strategies 1 and 3 above.

### 3. New Tools in GeoGebra Discovery: Handling Geometric Inequalities

In the previous paragraphs we summarily described how automated reasoning tools have been implemented in GeoGebra for proving geometric equalities. However, many classic theorems and results in Geometry are not stated in the form of an equality, but through the use of inequalities—and immediately comes to mind as an essential example the *triangle inequality*. However, working with inequalities forces us to enter into the realm of Real Algebraic Geometry and its corresponding set of instruments, so that the methods based on polynomials in closed algebraic fields do not suffice to cover this interesting family of geometric results.

In this respect, we believe that at present GeoGebra is pioneering the implementation of automated reasoning tools for proving and discovering geometric inequalities in a DGS environment, and aims at being able to produce proofs in a reasonable timeframe for a wide collection of results.

These new tools are currently in the development stage, and have not yet been implemented in official releases of GeoGebra. Nevertheless, they can be accessed through an experimental version of GeoGebra denominated *GeoGebra Discovery* ([30]). The (real) algebraic methods that are at the root of dealing with this new set of geometric propositions are based on real quantifier elimination and a derived technique denominated Cylindrical Algebraic Decomposition (CAD). The following is a brief description of the basic ideas involved when performing a CAD.

We start with a given geometric construction that, in a similar way as in the previous case when considering geometric equalities, is defined by a collection of polynomial relations. However, now we introduce two main differences: On the one hand, we restrict our variables to take values in the field  $\mathbb{R}$ , so that we focus our attention in the real plane. On the other hand, we allow them to be only positive or negative, or to hold inequality relations among them. In this way, we obtain a better, more precise description of the set of instances available for our Euclidean construction. Therefore, instead of working with complex varieties, our set of constructions can be represented by a subset of  $\mathbb{R}^n$  of a type that generalizes that of a variety, called a *semialgebraic set*. These sets can be expressed (see [31]), (Chapter 3) as a union of sets of the form

$$H = \{(x_1, \dots, x_n) \in \mathbb{R}^n : h_i(x_1, \dots, x_n) = 0, h_j(x_1, \dots, x_n) > 0, \\ i = 1, \dots, d, j = d + 1, \dots, m\}.$$

The possibility of working in a real geometry environment allows us to consider details that escaped our previous complex geometry approach. To see how naturally semialgebraic sets appear in geometric constructions, let us translate into algebraic form the statement

*Given two points in the plane, take a point P in the interior of the line segment AB.*

If we set coordinates  $A(u_1, u_2)$ ,  $B(v_1, v_2)$  and  $P(x, y)$ , then the algebraic translation would give the following semialgebraic set of points  $(u_1, u_2, v_1, v_2, x, y) \in \mathbb{R}^6$  defined by the following equations and inequalities:

$$H = \{(x - u_1)(y - v_2) - (x - v_1)(y - u_2) = 0, (x - u_1)(v_1 - x) > 0\} \\ \cup \{(x - u_1)(y - v_2) - (x - v_1)(y - u_2) = 0, (y - u_2)(v_2 - y) > 0\}. \quad (4)$$

Here is where a CAD becomes handy. A CAD is a partition of  $\mathbb{R}^n$  in semialgebraic sets. Let us consider first the projections  $\pi_k : \mathbb{R}^n \rightarrow \mathbb{R}^k$ ,  $(u_1, \dots, u_n) \mapsto (u_1, \dots, u_k)$ ,  $1 \leq k \leq n - 1$ . Given a set  $S \subset \mathbb{R}^{n-1}$ , a *cylinder* is a set of the form  $S \times \mathbb{R}$ , and a *stack* is a



partition of a cylinder of the form  $S \times \mathbb{R} = \bigsqcup_{j=1}^{2k+1} S_j$ , induced by a sequence of continuous functions  $f_0 := -\infty < f_1 < \dots < f_k < +\infty =: f_{k+1}$  on  $S$  so that:

- If  $j$  is odd, then

$$S_j = \{(u_1, \dots, u_n) : (u_1, \dots, u_{n-1}) \in S \wedge f_{(j-1)/2}(u_1, \dots, u_{n-1}) < u_n < f_{(j+1)/2}(u_1, \dots, u_{n-1})\}.$$

- If  $j$  is even, then

$$S_j = \{(u_1, \dots, u_n) : (u_1, \dots, u_{n-1}) \in S \wedge u_n = f_{j/2}(u_1, \dots, u_{n-1})\}.$$

Notice that we have  $\pi_{n-1}(S_j) = S$  for each  $j$ . Now we can define a CAD inductively as follows:

- For  $n = 1$ , a CAD of  $\mathbb{R}$  is a finite partition  $\mathbb{R} = \bigsqcup_{j=1}^{2k+1} S_j$  of the real line in  $k$  points and the  $k + 1$  open intervals which constitute their complement (it can be viewed as a stack over a point). A point is called a 0-cell and an open interval a 1-cell of the CAD.
- For  $n > 1$ , a CAD of  $\mathbb{R}^n$  is a partition of  $\mathbb{R}^n = \bigsqcup_{i=1}^r (T_i \times \mathbb{R}) = \bigsqcup_{i=1}^r (\bigsqcup_{j=1}^{2k_i+1} S_{ij})$  in stacks  $T_i \times \mathbb{R} = \bigsqcup_{j=1}^{2k_i+1} S_{ij}$ , whose elements are semialgebraic sets, and such that the collection  $\{T_i\}$ ,  $i = 1, \dots, r$  is a CAD of  $\mathbb{R}^{n-1}$  (here we have renamed and reindexed the elements of the  $(n - 1)$ -dimensional CAD to avoid accumulation of subindices). If  $T_i$  is a  $k$ -cell, then  $S_{ij}$  is either a  $(k + 1)$ -cell (if  $j$  is odd) or a  $k$ -cell (if  $j$  is even). Besides,  $\pi_{n-1}(S_{ij}) = T_i$  and, by the recursive nature of the construction, the collection of sets  $\{\pi_k(S_{ij})\}$  (with repeated elements) forms a CAD of  $\mathbb{R}^k$ .

Now, given a finite set of polynomials  $P = \{p_i\}$ , there are algorithms whose output is a CAD, together with a sample point in each of its elements, and so that for all the points in each of the elements of the partition and each  $p_i \in P$  only one of the relations  $p < 0$ ,  $p = 0$ ,  $p > 0$  holds. We then say that the CAD is *compatible* with  $P$  (see Figure 4 for an example).

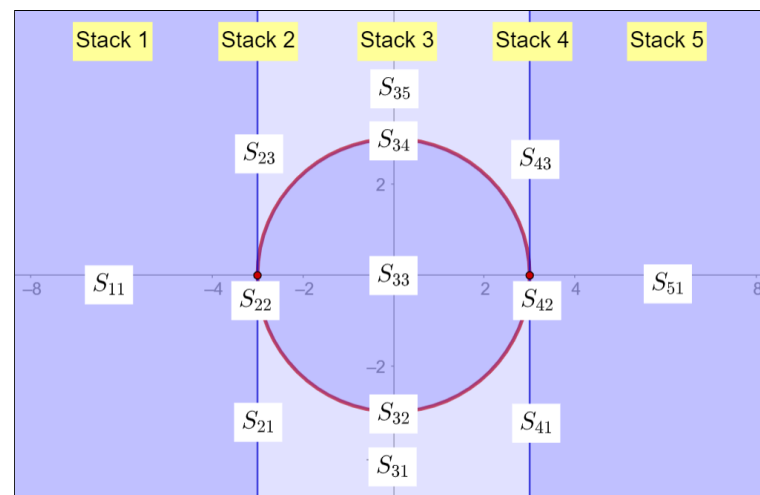


Figure 4. CAD compatible with the polynomial  $x^2 + y^2 - 9$ .

Therefore, if we describe a geometric construction by the hypothesis semialgebraic set  $H \subset \mathbb{R}^n$  and our thesis is represented by an inequality  $T = \{T(u_1, \dots, u_n) \geq 0\} \subset \mathbb{R}^n$ , we can construct a CAD for the collection of all polynomials defining the hypothesis and thesis sets. For each set  $S_{ij}$  in the resulting partition, either  $S_{ij} \subset H$  or  $S_{ij} \cap H = \emptyset$ . Since the CAD provides also a complete sample of points  $\{q_{ij}\}$  for all the pieces  $S_{ij}$  and the sign of  $T$  on each piece does not change on each of them, we can verify whether the inequality  $T \geq 0$  is satisfied on all the sets  $S_{ij}$  that constitute  $H$  by verifying that  $T(q_{ij}) \geq 0$  for each of them, and this would mean that our geometric statement is *true*. Unfortunately, the computational

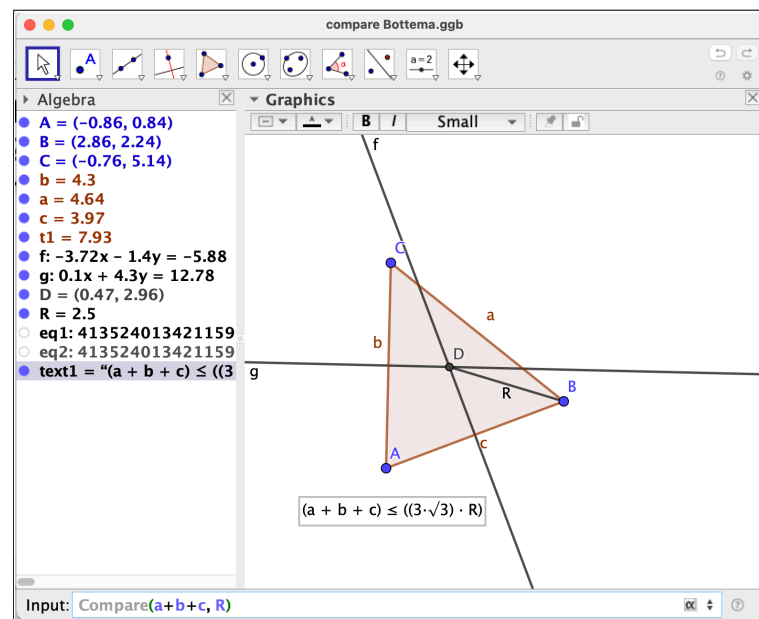
complexity of performing a CAD is very high, being doubly exponential on the number of variables (see [32]), and this renders the process unpractical in complex configurations. Nonetheless, some clever implementations, combined with other algebraic techniques that simplify the process, have been already successful, time ago, to automatically prove a wide number of classic, non-trivial results ([33–35]).

However, we can take advantage of the semialgebraic set  $H$  in order to *discover* new inequalities as well. Let us assume that we want to investigate the relationship between two quantities  $u > 0, v > 0$  that have a geometric meaning in our construction, and find out whether there is a constant  $p$  such that an inequality of the form  $u \leq pv$  (or a similar one) holds. Then we can include  $\{p, u, v\}$  in the space of variables where our Hypothesis semialgebraic set lies ( $p$  as first coordinate), together with the defining polynomials for  $u$  and  $v$  and the equation  $u - pv = 0$  that defines  $p$  as the ratio of the quantities  $u, v$ . Thus, we obtain a hypothesis set in some  $\mathbb{R}^n$  which also contains as coordinates the variables  $\{p, u, v\}$ . Recall that *each point of  $H$  corresponds to an instance of our geometric construction*. Now, if we proceed to construct a CAD and use it to project  $H$  on the coordinate  $p$ -axis we will get a semialgebraic subset in this axis, which is the collection of intervals or points for which there exists geometric constructions providing the corresponding value  $p$ . The CAD algorithm facilitates the computation of this projection set, and if we obtain, for example, an interval of the form  $0 \leq p \leq M$ , this leads to a proof of the inequality  $u \leq Mv$ . Thus, this algorithm turns out to be a suitable tool for the discovery of new geometric facts.

A detailed description of the algebraic and geometric aspects of the construction of a CAD can be found in [36,37]. There are implementations of the corresponding algorithms in several software programs such as Mathematica, Maple, Reduce or QEPCAD, and GeoGebra Discovery, in its most recent versions, has incorporated them through its *realgeom* extension, see [38]. In its most recent release, GeoGebra Discovery embeds the Tarski/QEPCAD B program (see [39–41]) to perform these CADs, opening the possibility of proving geometric inequalities without resorting to external programs. The commands in GeoGebra that put in action this new set of tools are the following:

- `Compare(<Expression, Expression>)`: This command performs a sequence of tasks that can be briefly described as follows (see [40]):
  1. The first two points of the construction are identified with (0,0) and (1,0). This reduces the number of variables required, and avoids some degenerate configurations.
  2. By using only the equalities defining the hypothesis, there is a first trial to check for an equality relation between the input expressions. This is handled by the Giac CAS embedded in GeoGebra.
  3. If equality relations are not found, the set of equations in the hypothesis is processed, eliminating unnecessary variables and generating a quantified formula.
  4. Now, via the Tarski/QEPCAD B software, this quantified formula is converted into a quantifier-free formula, by means of a CAD algorithm that outputs an inequality for the quotient of the input expressions.
  5. This inequality is translated and displayed into a simple form, easy to understand by the user.

See Figure 5 for an example of the performance of the `Compare` command regarding the statement of Proposition 1.



**Figure 5.** Discovering the inequality between the perimeter and the radius of the circumcenter of a triangle.

- **Relation(<list of objects>):** As mentioned before, this higher level command performs two distinct operations ([28]): first, proceeds to make a numerical comparison of the objects, and outputs an answer concerning geometric properties, including—this is the more recent improvement—inequalities, etc. Then the user finds a “More...” button that, if clicked, starts a deeper, symbolic analysis of the numerical result previously obtained, proving or rejecting its truthfulness. A detailed example is developed in Section 5, and shown in Figures 10 and 11.

#### 4. Visualization of Real Curves and the Dynamic Color Scanning Method

Nowadays, computer visualization of mathematical objects is an active area of research, which finds applications in a great variety of disciplines. In particular, the graphic representation of real algebraic curves (especially in the neighborhood of a singularity) poses a number of challenges that, although have been faced by numerical and also by symbolic methods (see [42] and its many references, as well as [43] as a more recent contribution), still lack accuracy in current computer software implementations, even in apparently innocent-looking examples.

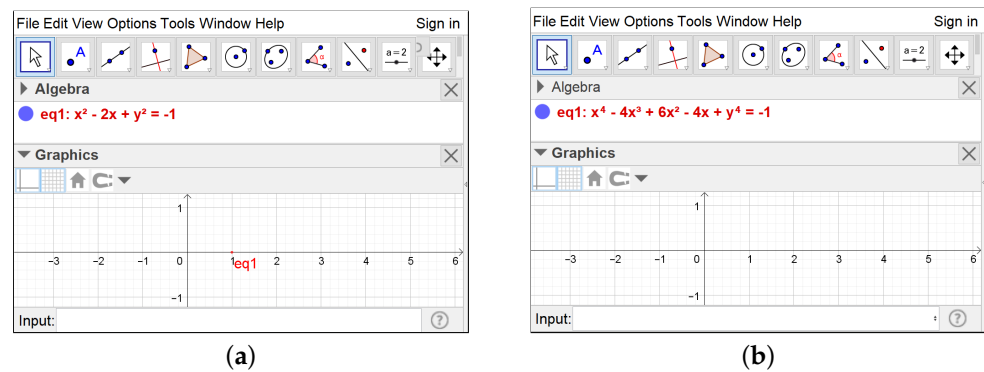
**Example 1.** Within GeoGebra, introduce in the input line the command

```
ImplicitCurve((x-1)^2+y^2),
```

whose output corresponds to the curve  $C$  with equation  $(x-1)^2 + y^2 = 0$ . It is obvious that the real part of  $C$  contains only the point  $(1, 0)$ . GeoGebra will display correctly the curve, as shown in Figure 6a. If we want to visualize now the curve  $C'$  given by  $(x-1)^4 + y^4 = 0$ , whose real part coincides with that of  $C$ , and introduce

```
ImplicitCurve((x-1)^4+y^4),
```

GeoGebra will display nothing on its graphic view, as shown in Figure 6b. These issues can also affect the command *LocusEquation()*, since it makes use of *ImplicitCurve()* to represent loci graphically.



**Figure 6.** GeoGebra output for the implicit curves  $(x - 1)^2 + y^2 = 0$  (success, subfigure (a)) and  $(x - 1)^4 + y^4 = 0$  (failure, subfigure (b)).

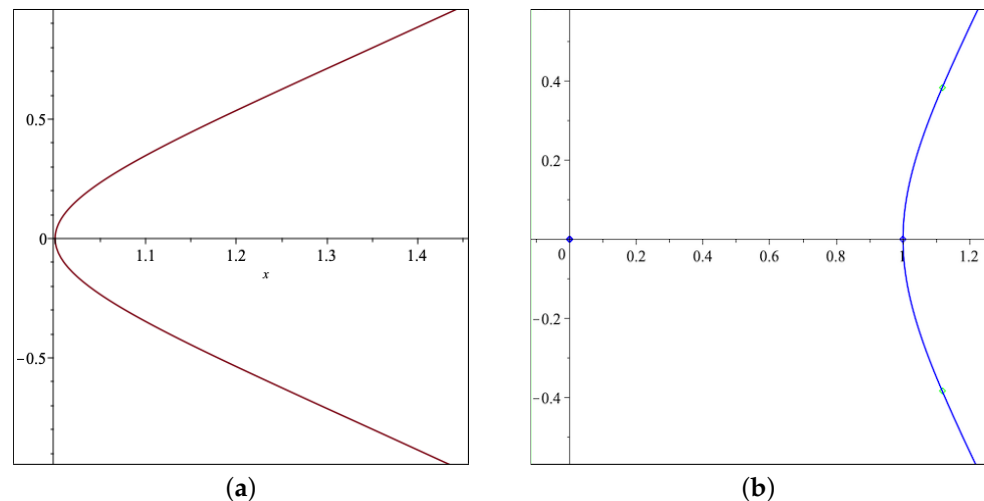
**Example 2.** Even advanced mathematical software packages such as *Maple* face similar problems with its plotting commands. The graphic output of the set of instructions

```
with(plots);
implicitplot(x^2+y^2-x^3, x=-1..1, y=-1..1);
```

will miss the real point  $(0,0)$  that belongs to this cubic curve, see Figure 7a. This is due to the fact that the *implicitplot* command uses a sample-based method which is not able to detect accurately isolated points. However, *Maple* includes other representation options that allow a more precise output for the case of real algebraic curves, by means of the package *algcurves*. Executing now

```
with(algcurves);
plot_real_curve(x^2+y^2-x^3, x, y);
```

our previous missing point will appear on screen, as shown in Figure 7b.

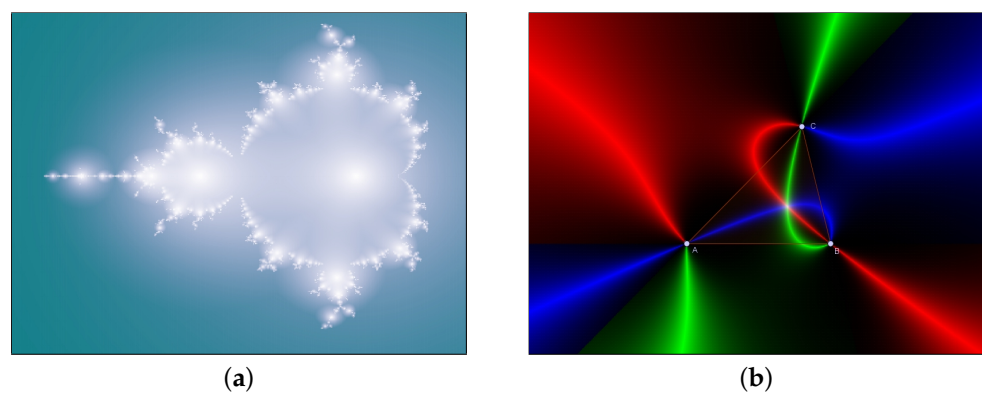


**Figure 7.** Graphic output of *Maple* for the curve  $x^2 + y^2 - x^3 = 0$  (a) with the *implicitplot()* command; (b) with the *plot\_real\_curve()* command.

Being the case that important characteristics of algebraic curves can be lost when making use of these plotting commands, it is interesting to explore other ways to approach the problem of visualizing (real) algebraic curves. This is the moment to take a closer look at some of GeoGebra's special functionalities. Since this program was mainly designed as a software to be used for the teaching and learning of Mathematics, it incorporates a set of tools that facilitates the visualization of geometric constructions and their properties, making them more attractive and comprehensible to students. It turns out that these tools

can also be employed as a powerful resource to explore diverse problems where complex geometric configurations arise. In particular, the combined use of the properties *Show Trace* and *Dynamic color* that can be assigned to objects in the graphic window of GeoGebra creates a scanning tool capable of showing hidden relations among the elements of a given construction. We refer to this technique as *the dynamic color scanning method*.

The basic idea behind this method lies in the ability to assign a color to each point in the plane depending on its position in relation to a geometric construction, element or property we have to deal with. Whether working with distances and other measures, with operations performed on a set of these quantities, or with equations corresponding to geometric loci, the scanning method in GeoGebra is easy to implement, reasonably fast and reliable, and produces outstanding visual information that can provide insight in a complex scenario. Pioneer work in the use of this technique in the GeoGebra environment can be attributed to the second author, who has a nice collection of applications in [44] (see Figure 8).



**Figure 8.** (a) The classic Mandelbrot set, created with the dynamic color scanning method. (b) An application to investigate the Fermat point of an arbitrary triangle.

Let us describe briefly how this visual method works by showing a simple example (for those unacquainted with GeoGebra, its reference manual can be found at [45]). Let us assume that we have three given points  $A, B, C$  on the plane, that we can display in the graphic window of GeoGebra. Let us denote the distance between two points  $P$  and  $Q$  by  $d(P, Q)$ .

**Problem 1.** Find the geometric locus of points  $P$  such that  $d(P, C) = d(P, A) + d(P, B)$ .

This locus is easy to display with the `ImplicitCurve` command in GeoGebra, see Figure 9a. Now, in order to exemplify how to visualize this locus through our dynamic color scanning method approach, let us start associating some colors to point  $P(x, y)$ , as follows. First, we remark that GeoGebra allows to set independently dynamic values of the three color channels RGB that colorize an object in GeoGebra. The numeric value for each channel ranges between 0 (no color) and 1 (full color), and for other values  $z$  outside this range GeoGebra applies a periodic pattern given by

$$\begin{cases} z - \lfloor z \rfloor, & \text{if } \lfloor z \rfloor \text{ is even} \\ 1 - (z - \lfloor z \rfloor), & \text{if } \lfloor z \rfloor \text{ is odd.} \end{cases}$$

In particular, if we set the values (R,G,B) to integers of the form (even, even, even) we get black, and by setting them to integers (odd, odd, odd) we obtain white. Besides, if all three channels have equal values, a color in the scale of greys is produced. Now, for instance, let us set dynamic values for the point  $P$  as follows:

$$\begin{aligned} \text{Red:} & \quad c/s \\ \text{Green:} & \quad 1 - \text{abs}(s - c) \end{aligned}$$



Blue :  $s/c$ ,

where  $c = \text{Distance}(P, C)$  and  $s = \text{Distance}(P, A) + \text{Distance}(P, B)$ . Observe that we will get for the point  $P$  the white color only when  $c = s$ , since values assigned to red and blue channels are positive and mutually inverse, and all three channels will give odd numbers (in fact, 1) only in this case.

Now, if we activate the trace for the point  $P$  and drag it, after covering the region we are interested in, we obtain the color scanning of the area, and we will have a visualization of the locus (the set of points white-colored) we are looking for. It is interesting to make here several remarks concerning the output image:

- The periodic patterns that frequently arise in form of concentric rings of similar colors are due to the periodic behavior of the dynamic colors when the values introduced in the RGB channels move outside the  $[0, 1]$  range, as previously mentioned (see Figure 9b).
- A way to restrict the values for the dynamic color channels in the range  $[0, 1]$  is obtained by using exponential expressions of the form

$$\begin{aligned}\text{Red :} & e^{(-\text{abs}(f(P)))} \\ \text{Green :} & e^{(-\text{abs}(g(P)))} \\ \text{Blue :} & e^{(-\text{abs}(h(P)))},\end{aligned}$$

so that only when simultaneously the functions  $f, g, h$ , depending on point  $P$ , are equal to 0 the white color will be assigned to  $P$ . In our example, by assigning to the color channels the values

$$\begin{aligned}\text{Red :} & e^{(-\text{abs}(1 - c/s))} \\ \text{Green :} & e^{(-\text{abs}(s - c))} \\ \text{Blue :} & e^{(-\text{abs}(1 - s/c))},\end{aligned}$$

we obtain Figure 9c.

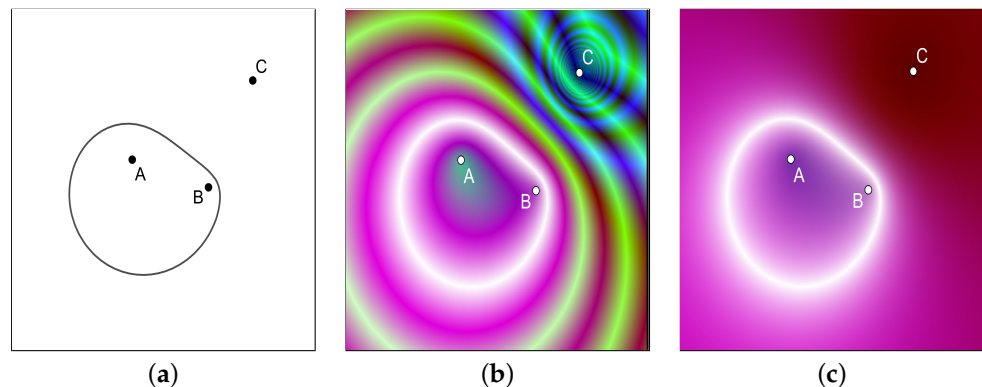
Besides, if we substitute in a given RGB channel, for instance the red one, the expression in the exponent  $\text{abs}(f(P))$  by  $(\text{abs}(k f(P)) - \text{floor}(k \text{abs}(f(P))))$ , where  $k$  is a convenient positive constant, we get the mantissa of  $k \text{abs}(f(P))$ , which drops abruptly to zero when  $f(P)$  crosses integer values, producing distinct lines because of the sudden color change in the red channel. This trick leads to the appearance of level curves connecting points  $P$  with the same  $f(P)$  value, which will be more densely distributed by choosing a greater value for the constant  $k$ . We will make use of this technique later in Section 6.1.

- Since we have three different channels that can hold independent values, by playing wisely with them, we have at our disposal up to *three different properties* to visualize in our scanning method. An illustration of how to take advantage of this feature is shown in Figure 8b, where an investigation on the Fermat's Point is carried out (see the section on "Escáner del Punto Fermat" in [44], for details on the construction).

Now, to improve the output of the process and avoid the cumbersome dragging of point  $P$  all over the graphic view we can benefit from other resources included in GeoGebra and proceed as follows:

1. Set a rectangular region to be scanned. Let us say it has as upper left corner the point  $(x_0, y_0)$ , and dimensions  $w \times h$ .
2. Shrink the size of the radius for the point  $P$  to its minimum size.
3. Create a slider  $t$  that will be used as the  $x$ -coordinate of  $P$ , and runs from  $x_0$  to  $x_0 + w$ .
4. Set the coordinates of  $P$  to  $(t, y_0)$  and rename the point to  $A1$ . This trick allows us to automatically assign the point  $P$  to the cell  $A1$  of the GeoGebra spreadsheet. Now we can create a list of points  $Ai$ ,  $1 \leq i \leq n$ , by taking advantage of the spreadsheet features, starting first by adding the point  $A2 = A1 - (0, \epsilon)$ , with the same properties as  $A1$ , and dragging downwards (as it is usually done when working with spreadsheets) the cell  $A2$ . This column of points (all of them inheriting the properties we set up for  $A1$ ) translates in the graphic window into a scanning vertical segment, which

will sweep the region as the slider  $t$  runs through its defined range. A recommended value for  $\epsilon$  is 0.02 in order to obtain a good quality of the scanning image. Therefore, for covering our rectangular region of height  $h$  we should create around  $n \approx 50 h$  points.



**Figure 9.** Locus of  $P$  such that  $d(P, A) + d(P, B) = d(P, C)$ , displayed (a) with the traditional GeoGebra tools, (b,c) with the dynamic color scanning method.

We will see below how this method proves useful when trying to understand geometric configurations that lead to demanding computational tasks that cannot be handled by the GeoGebra CAS. Other variations of these dynamic coloring methods in GeoGebra that show their flexibility and visualization potential can be found in [46] (see also [44,47]).

**Remark 2.** Similar visualization techniques to those presented here can be also employed in other DGS programs such as Cinderella [48] and its companion open project CindyJS [49], some of them inspired in Losada's scanning method for GeoGebra (see for instance [50]).

## 5. A Case Study: Relation between the Perimeter and Circumradius of a Triangle

Now we center our attention on Proposition 1, and let us see how GeoGebra can help us discover this proposition automatically, without having to provide any clue whatsoever as an input, apart from the basic construction required to set the scenario for our investigation.

We start by constructing the three vertices  $A, B, C$  of a triangle. Without loss of generality, we can assume  $A$  and  $B$  to be the points  $A(0,0)$  and  $B(1,0)$ . The third vertex  $C$  has free coordinates  $(x, y)$ . Now we draw the segments  $BC, CA$  and  $AB$ , with respective lengths  $a, b, c$ , and the perpendicular bisectors  $l$  and  $m$  to  $CA$  and  $AB$  respectively. Let us denote by  $D(d_1, d_2)$  the point of intersection of the two previous bisectors. The segment  $AD$  of length  $R$  is the circumradius of the triangle  $\triangle ABC$ . Our construction is all set. Now we introduce in the GeoGebra input line the command `Relation( $a + b + c, R$ )` and we obtain (see Figure 10) an initial, numerical output indicating that the quantities  $a + b + c$  and  $R$  do not coincide. Not very informative indeed, but if we press the “More...” button (see Figure 11) the *realgeom* extension enters in action and quickly shows that the relation  $a + b + c \leq 3\sqrt{3} R$  holds.

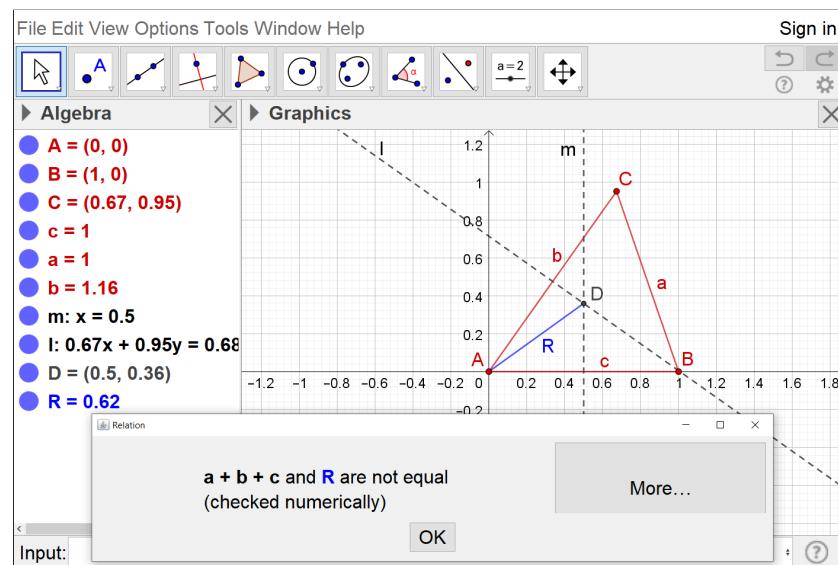


Figure 10. Numerical output of the Relation tool.

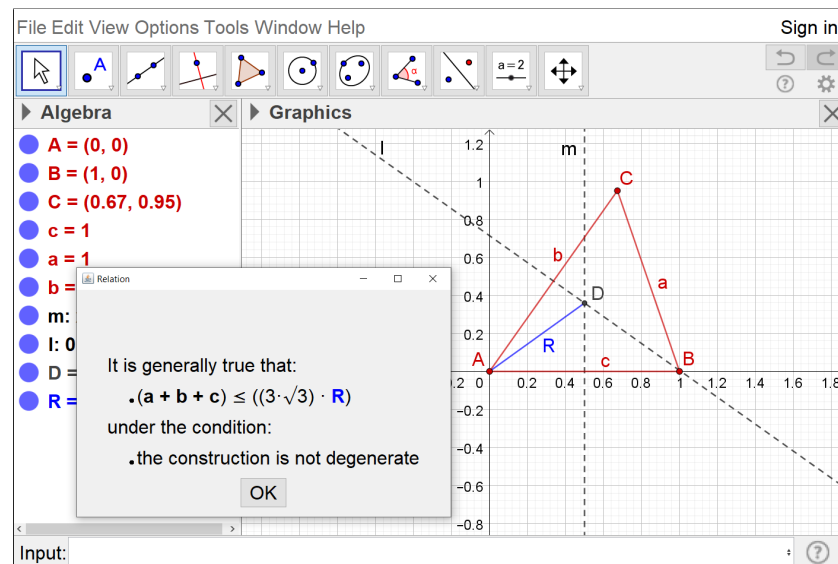


Figure 11. Symbolic output of the Relation tool.

However, Proposition 1 not only asserts that this inequality holds. It also states that the corresponding equality  $a + b + c = 3\sqrt{3} R$  is true if and only if the triangle  $\triangle ABC$  is equilateral. How can we prove this with our set of tools? Well, for this task the command `LocusEquation(<Boolean Condition>, <Point>)` becomes handy, since if we ask GeoGebra to provide us with the geometric locus of the points in the plane satisfying the equality relation  $a + b + c = 3\sqrt{3} R$  we should obtain an algebraic curve giving us information on the set of points we are looking for. Were Proposition 1 correct, this locus should reduce to the points  $C_1(\frac{1}{2}, \frac{\sqrt{3}}{2})$  and  $C_2(\frac{1}{2}, -\frac{\sqrt{3}}{2})$ , which are the only ones producing equilateral triangles together with A and B.

Having therefore a clear strategy to settle down the equality part of our statement, we type in the input bar `LocusEquation(a + b + c == 3*sqrt(3) * R, C)`, obtaining the following equation for this geometric locus:

$$\begin{aligned}
 P(x, y) := & 531441x^{16} + 3621672x^{14}y^2 + 10657980x^{12}y^4 + 17653464x^{10}y^6 \\
 & + 17955270x^8y^8 + 11448216x^6y^{10} + 4452732x^4y^{12} + 962280x^2y^{14} \\
 & + 88209y^{16} - 4251528x^{15} - 25351704x^{13}y^2 - 63947880x^{11}y^4
 \end{aligned}$$

$$\begin{aligned}
& - 88267320x^9y^6 - 71821080x^7y^8 - 34344648x^5y^{10} - 8905464x^3y^{12} \\
& - 962280x^{14}y^4 + 14880348x^{14} + 77551020x^{12}y^2 + 167328828x^{10}y^4 \\
& + 191561004x^8y^6 + 123219252x^6y^8 + 42802020x^4y^{10} + 6689844x^2y^{12} \\
& + 204228y^{14} - 29760696x^{13} - 135733968x^{11}y^2 - 250455240x^9y^4 \\
& - 236640096x^7y^6 - 118283976x^5y^8 - 28362960x^3y^{10} - 2237112x^{12}y^{12} \quad (5) \\
& + 37200870x^{12} + 149197140x^{10}y^2 + 235946682x^8y^4 + 183225240x^6y^6 \\
& + 69737274x^4y^8 + 10694484x^2y^{10} + 236134y^{12} - 29760696x^{11} \\
& - 105973272x^9y^2 - 144481968x^7y^4 - 92158128x^5y^6 - 26125848x^3y^8 \\
& - 2237112x^{10}y^{10} + 14880348x^{10} + 47790324x^8y^2 + 56867832x^6y^4 \\
& + 30035016x^4y^6 + 6281388x^2y^8 + 204228y^{10} - 4251528x^9 \\
& - 12597120x^7y^2 - 13401936x^5y^4 - 6018624x^3y^6 - 962280x^{12}y^8 \\
& + 531441x^8 + 1495908x^6y^2 + 1485702x^4y^4 + 609444x^2y^6 \\
& + 88209y^8 = 0
\end{aligned}$$

The degree of this polynomial equation is 16 and, unfortunately, GeoGebra is not capable of producing any points of the corresponding curve in the graphical window (see Figure 12, where all elements apart from the locus have been hidden to allow a better visualization). Our strategy fails! In principle, it could be the case that the curve had no real points, but a quick inspection of the polynomial  $P(x, y)$  is enough to see that at least the point  $(0, 0)$  must belong to it. Therefore, something is going wrong here... It is time to look for another way to find out what is the real aspect of this curve.

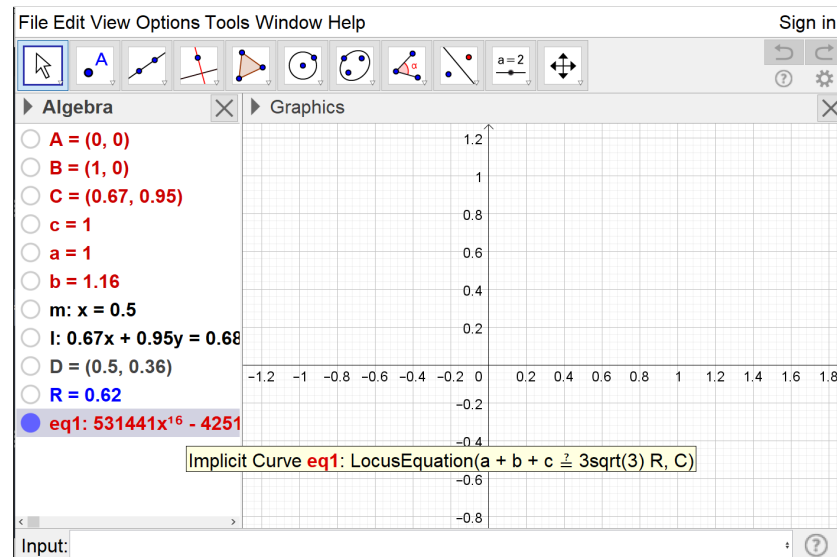


Figure 12. The algebraic curve  $P(x, y) = 0$  does not appear on-screen.

## 6. Combining Visual and Symbolic Strategies to Overcome Difficulties

### 6.1. The Visual Approach Using the Dynamic Color Scanning Method

After the difficulties that we have encountered when trying to study the algebraic curve  $P(x, y) = 0$  by means of the `LocusEquation()` command, let us approach the problem from a more visual strategy, by using the dynamic color scanning method described in Section 4. Let us choose as scanning area the rectangle  $R$  with two opposite vertices located at  $(-2, -2)$  and  $(3, 2)$ , which contains points  $A, B$  of our construction. We have to assign dynamic values to the three RGB color channels of the scanning points, and these values should depend on some kind of measure of the “distance” of the point to the curve  $P(x, y) = 0$ . As an initial choice, we could just choose the values

$(e^{-|P(x,y)|}, e^{-|P(x,y)|}, e^{-|P(x,y)|})$  for each point  $(x, y)$ , but the high values that the polynomial  $P$  attains on the considered region (for instance,  $P(2, 2) = 448988778496$ ) produces undesirable effects in the visualization process, so that it is better to adjust these values in order to obtain a better display. A good choice to get a neat output is obtained by assigning to each point  $C_i$  of the scanning segment the following values (see Figures 13 and 14):

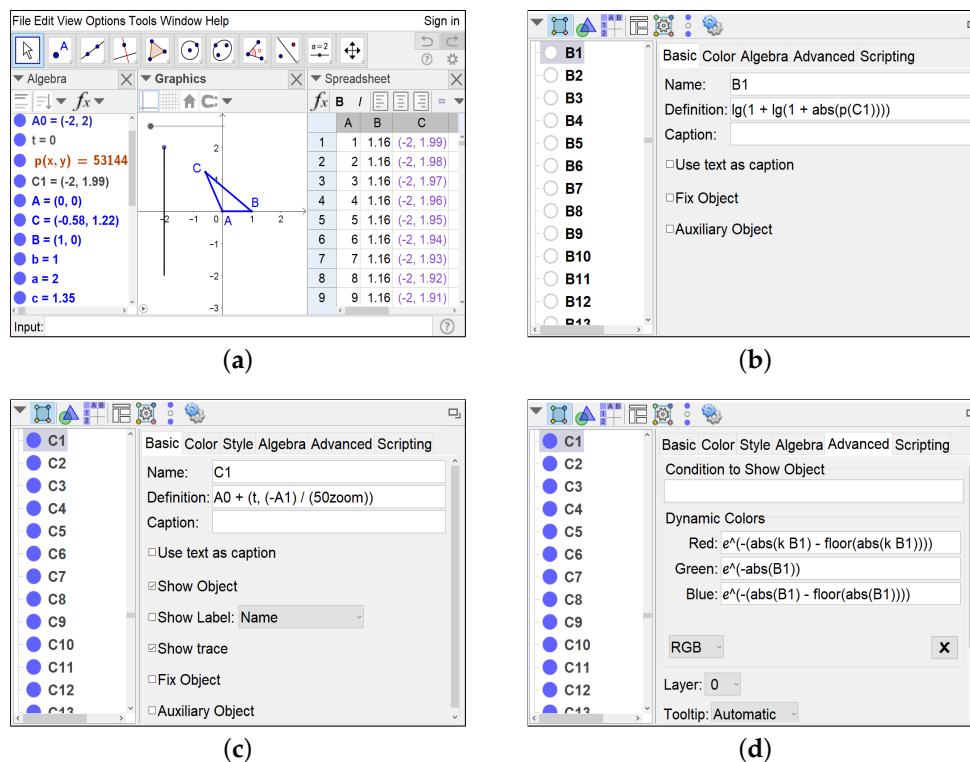


Figure 13. (a) Setting up the dynamic color scanner. (b,c) Basic parameters of the scanning points. (d) Parameter values for the RGB channels.

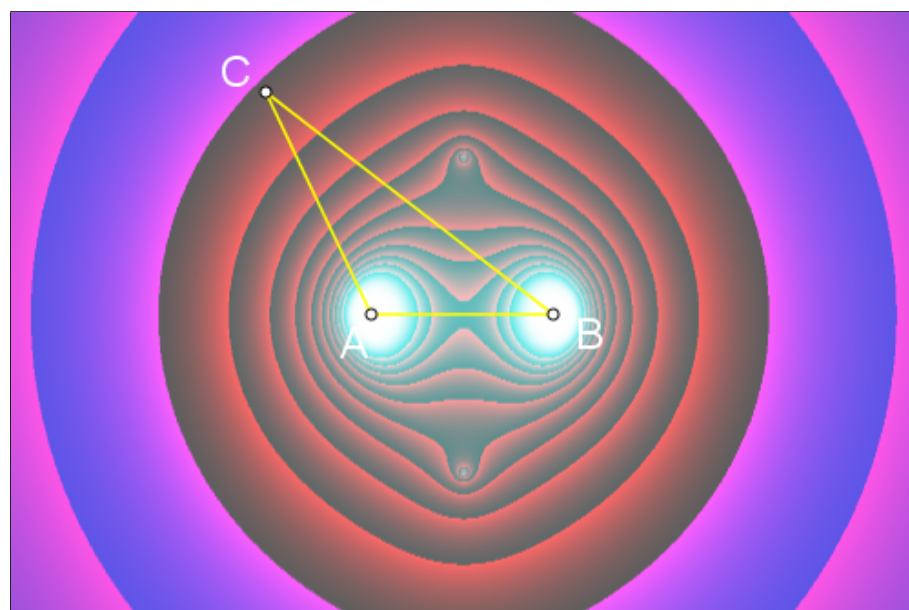


Figure 14. Output after a dynamic color scanning of the plane for the locus  $P(x, y) = 0$ .

Red:  $e^{-(\text{abs}(k \cdot B_i) - \text{floor}(\text{abs}(k \cdot B_i)))}$   
 Green:  $e^{-\text{abs}(B_i)}$



Blue :  $e^{-(\text{abs}(Bi) - \text{floor}(\text{abs}(Bi)))}$ ,

Here, recommended values for the values  $k, Bi$  are  $k = 10$  and  $Bi = \log(1 + \log(1 + \text{abs}(P(Ci))))$ .

As already seen, the “curve” of white points in the scanned region would represent the part of the geometric locus  $a + b + c = 3\sqrt{3} R$  contained in  $R$ . However, instead of a curve we perceive four whitened areas: two of them, more visible, with centers located around  $(0, 0)$  and  $(1, 0)$ , and two smaller ones (these ones are trickier to spot, and the technique described in Section 4 for producing isolines, applied to the red and blue channels, helps detecting them) with centers lying on the line  $x = \frac{1}{2}$ . This leads us to state the following conjecture:

**Conjecture 1.** *The number of real points of the curve  $P(x, y) = 0$  is finite and equal to four. Besides, two of them have as x-coordinate the value  $x = \frac{1}{2}$ , while the two others have x-coordinates  $x = 0$  and  $x = 1$  respectively.*

So, the dynamic color scanning method has helped us approach the solution to our question on the conditions that correspond to the equality case in Proposition 1. Of course, this cannot be considered a formal proof, but the information that we have obtained from this visual exploration allows us to return to a symbolic approach, this time with a conjecture in mind, and with Maple and its algebraic toolkit as auxiliary assistant.

## 6.2. The Symbolic Approach Using Maple Commands

Thanks to the previous section we now have a better intuition about the geometric locus corresponding to the equality  $a + b + c = 3\sqrt{3} R$ , and we can try to resort to more powerful tools in order to definitely settle the question that has resisted our analysis so far. From now on, we will use Maple and a variety of its available tools to perform a thorough algebraic study of the situation in order to completely elucidate the set of real points in  $P(x, y) = 0$ .

In the first place, since we already have some familiarity with CADs, we will use higher level commands implemented in recent versions of Maple to quickly settle down Proposition 1. To achieve this, we need the RegularChains package and its commands to perform CADs:

```
> with(RegularChains):
with(ChainTools):
with(SemiAlgebraicSetTools):
R:=PolynomialRing([y,x])
cad:=CylindricalAlgebraicDecompose(P,R,method = recursive,
output=piecewise)
```

Here,  $P$  represents the polynomial detailed in the precedent Equation (5). Executing this group of instructions we get

$$cad := \left\{ \begin{array}{ll} 1 & x < 0 \\ \left\{ \begin{array}{l} 1 \quad y < 0 \\ 1 \quad y = 0 \\ 1 \quad y > 0 \end{array} \right. & x = 0 \\ 1 & 0 < x \text{ and } x < 1/2 \\ \left\{ \begin{array}{l} 1 \quad y < -1/2\sqrt{3} \\ 1 \quad y = -1/2\sqrt{3} \\ 1 \quad \text{and}(y > -1/2\sqrt{3}, y < 1/2\sqrt{3}) \\ 1 \quad y = 1/2\sqrt{3} \\ 1 \quad y > 1/2\sqrt{3} \end{array} \right. & x = 1/2 \\ 1 & 1/2 < x \text{ and } x < 1 \\ \left\{ \begin{array}{l} 1 \quad y < 0 \\ 1 \quad y = 0 \\ 1 \quad y > 0 \end{array} \right. & x = 1 \\ 1 & 1 < x \end{array} \right. ,$$

which graphically corresponds to the CAD shown in Figure 15.

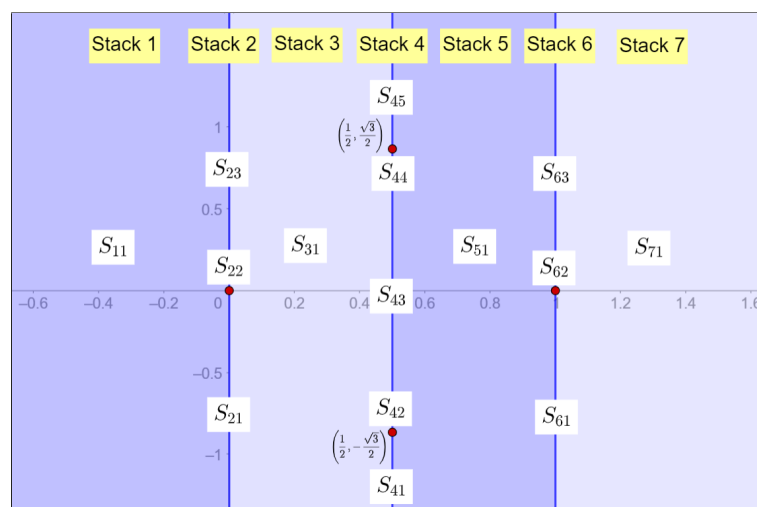


Figure 15. CAD compatible with the set of solutions of  $P(x, y) = 0$ .

This confirms at once our Conjecture 1, as well as the equality case in Proposition 1, and we are done.

However... what kind of symbolic calculations lie behind performing a CAD compatible with a set of given polynomials? In order to gain a better understanding of how Maple is capable of completing these tasks related to working with polynomials in real coefficients, in the sequel, we proceed to give a more guided, step-by-step treatment of the problem, by using lower level Maple commands that can also be translated to other software packages such as the Xcas/Giac environment.

We start by translating into algebraic form our initial construction (see Table 1).

**Table 1.** Algebraic translation of geometric construction in GeoGebra.

Construction Element	Variables	Polynomial Equation
Points $A(0,0)$ , $B(0,0)$ , $C(x,y)$	$\{x, y\}$	
Perpendicular Bisector $l$ to $AB$	$\{u\}$	$u - \frac{1}{2} = 0$
Perpendicular Bisector $m$ to $BC$	$\{x, y, u, v\}$	$(x-1)(u - \frac{x+1}{2}) + y(v - \frac{y}{2}) = 0$
Intersection of $l$ and $m$ is $D(u,v)$	$\{u, v\}$	
Segment $h = CD$	$\{x, y, u, v, h\}$	$h^2 - (x-u)^2 - (y-v)^2 = 0$
Segment $a = BC$	$\{x, y, a\}$	$a^2 - (x-1)^2 - y^2 = 0$
Segment $b = CA$	$\{x, y, b\}$	$b^2 - x^2 - y^2 = 0$
Segment $c = AB$	$\{x, y, c\}$	$c^2 - 1 = 0$
Square Root of 3	$\{r\}$	$r^2 - 3 = 0$
Locus Equation	$\{a, b, c, h, r\}$	$a + b + c - 3 \cdot r \cdot h = 0$

Notice that we take care of expressing all our relations concerning distances and other measures in polynomial form. This allows the use of the algebraic tools implemented in Maple. We define now the ideal containing our hypothesis, which we denote by *Hypo*:

```
> restart:with(PolynomialIdeals):Hypo:=PolynomialIdeal(u-(1/2),
(x-1)*(u-((x+1)/2))+y*(v-(y/2)),h^2-(x-u)^2-(y-v)^2,
a^2-(x-1)^2-y^2,b^2-x^2-y^2,c^2-1, variables={x,y,u,v,h,a,b,c});
```

$$Hypo := \langle 2u - 1, (x-1)\left(u - \frac{x}{2} - \frac{1}{2}\right) + y\left(v - \frac{y}{2}\right), \\ a^2 - (x-1)^2 - y^2, h^2 - (x-u)^2 - (y-v)^2, c^2 - 1, b^2 - x^2 - y^2 \rangle$$

The Hilbert dimension of this ideal gives us information on the number of maximal independent variables, giving us the *degrees of freedom* of the construction:

```
> HilbertDimension(Hypo); EliminationIdeal(Hypo, {x,y});
```

$$\begin{matrix} 2 \\ \langle 0 \rangle \end{matrix}$$

Therefore, we can consider as free variables the coordinates  $x, y$  of the point  $C$ , since we can drag this point on the plane and its position determines the whole construction. Now we introduce the polynomial related to the locus equation (together with the relation defining  $\sqrt{3}$ ) to obtain ideal  $T$ , which should decrease its Hilbert dimension by one:

```
> T:=PolynomialIdeal(u-(1/2),(x-1)*(u-((x+1)/2))+y*(v-(y/2)),
h^2-(x-u)^2-(y-v)^2,a^2-(x-1)^2-y^2,b^2-x^2-y^2,c^2-1,r^2-3,
a+b+c-3*r*h, variables={x,y,u,v,h,a,b,c,r}); HilbertDimension(T);
```

$$T := \langle 2u - 1, (x-1)\left(u - \frac{x}{2} - \frac{1}{2}\right) + y\left(v - \frac{y}{2}\right), a^2 - (x-1)^2 - y^2, \\ h^2 - (x-u)^2 - (y-v)^2, c^2 - 1, r^2 - 3, b^2 - x^2 - y^2, -3rh + a + b + c \rangle$$

$$1$$

Therefore, the variables  $\{x, y\}$  are not free anymore, and a polynomial relation among them must hold. If we type now

```
> P:=EliminationIdeal(T, {x,y});
```

we obtain the same polynomial  $P(x, y)$  appearing in Equation (5) (with a different order of terms), confirming the good job done by GeoGebra when computing the locus equation for the equality  $a + b + c = 3\sqrt{3} R$ . Our intuition, reinforced through our visualization

approach in Section 6.1, invites us to think that this algebraic curve should have only two real points (apart from degenerate cases that already have been considered). Therefore, our mission consists now in finding all the real points belonging to the curve  $P(x, y) = 0$ . Since Maple also includes graphic tools to plot real algebraic curves, in the same spirit as we tried with GeoGebra we can see whether Maple outputs (see Figure 16) successfully the real set of points for this curve:

```
> with(algcurves): plot_real_curve(P(x,y), x, y);
```

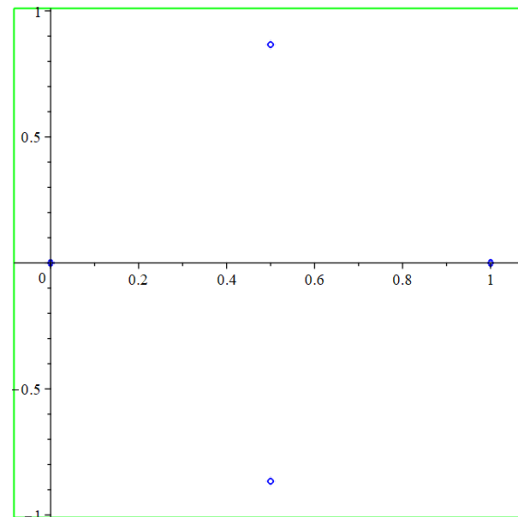


Figure 16. Maple output for locus equation  $P(x, y) = 0$ .

Indeed, it seems that this time the picture looks complete: The graphic output shows four points that certainly could correspond to the degenerate cases  $C_1(0, 0)$  and  $C_2(1, 0)$ , together with the elusive points  $C_3(\frac{1}{2}, \frac{\sqrt{3}}{2}) \approx (0.5, 0.866)$  and  $C_4(\frac{1}{2}, -\frac{\sqrt{3}}{2}) \approx (0.5, -0.866)$ , which correspond to the two possible equilateral triangle configurations. We are getting closer to our goal, but we have learned that, even though the visual approach has provided us with clues to settle the problem, at the same time it can be an inaccurate technique.

Now, in order to verify rigorously that these are the only real points of our curve, we will proceed as follows:

**Step 1:** Check that  $P(x, y) = 0$  for  $x \in \{0, \frac{1}{2}, 1\}$  has real solutions in  $y$ , and determine them all.

**Step 2:** Check that  $P(x, y) = 0$  for  $x \notin \{0, \frac{1}{2}, 1\}$  has not real solutions in  $y$ .

These two facts will lead to a full determination of the set of real points in our geometric locus.

**Step 1:** We first set  $x = 0$  and compute  $P(0, y)$ , factorizing the result.

```
> simplify(subs(x=0, P(x, y))); factor(%);
88209y16 + 204228y14 + 236134y12 + 204228y10 + 88209y8
y8(297y4 + 432y3 + 658y2 + 432y + 297)(297y4 - 432y3 + 658y2 - 432y + 297)
```

From the first factor it follows that there exists the real solution  $y = 0$ , and to exclude any other possibilities it is enough to count real roots by using the Sturm sequence of the second factor (since the third factor has symmetric real roots, being obtained from the second through the substitution  $y \rightarrow -y$ ).

```
> sturmseq(297*y^4 + 432*y^3 + 658*y^2 + 432*y + 297, y);
sturm(%, y, -infinity, infinity);
```

$$\left[ y^4 + \frac{16}{11}y^3 + \frac{658}{297}y^2 + \frac{16}{11}y + 1, y^3 + \frac{12}{11}y^2 + \frac{329}{297}y + \frac{4}{11}, -y^2 - \frac{2248}{2323}y - \frac{2835}{2323}, y - \frac{30348}{32993}, 1 \right]$$

For the value  $x = 1$ , we proceed similarly and obtain

> simplify (subs (x=1,P(x,y)));

$$88209y^{16} + 204228y^{14} + 236134y^{12} + 204228y^{10} + 88209y^8,$$

Since this polynomial coincides with the one in case  $x = 0$ , we again have that  $y = 0$  is the only real root for  $x = 1$ .

Finally, let us study the case  $x = \frac{1}{2}$ . Now,

> simplify (subs (x=1/2,P(x,y)));

$$\frac{(432y^4 + 152y^2 + 27)^2(1936y^4 + 744y^2 + 81)(4y^2 - 3)^2}{65536}$$

The last factor in this expression has roots  $y = \pm \frac{\sqrt{3}}{2}$ , while the others do not produce further real roots:

> Sturmseq(1936\*y^4 + 744\*y^2 + 81,y);  
Sturm(%y, -infinity, infinity);

$$\left[ y^4 + \frac{93}{242}y^2 + \frac{81}{1936}, y^3 + \frac{93}{484}y, -y^2 - \frac{27}{124}, y, 1 \right]$$

> Sturmseq(432\*y^4 + 152\*y^2 + 27,y);  
Sturm(%y, -infinity, infinity);

$$\left[ y^4 + \frac{19}{54}y^2 + \frac{1}{16}, y^3 + \frac{19}{108}y, -y^2 - \frac{27}{76}, y, 1 \right]$$

Therefore, after we finish Step 1 we have checked that the curve contains the real points  $A(0,0)$ ,  $B(1,0)$ ,  $C_1(\frac{1}{2}, \frac{\sqrt{3}}{2})$  and  $C_2(\frac{1}{2}, -\frac{\sqrt{3}}{2})$ . However... Are these the only real points?

**Step 1:** We start by considering again the polynomial  $P(x,y)$ , but now displayed as a polynomial in  $(\mathbb{R}[x])[y]$ , that is, as a polynomial in the variable  $y$  with coefficients in the domain  $\mathbb{R}[x]$ :

> P(y):= collect (P(x,y), y);  

$$P(y) := 88209y^{16} + (962280x^2 - 962280x + 204228)y^{14} + (4452732x^4 - 8905464x^3 + 6689844x^2 - 2237112x + 236134)y^{12} + (11448216x^6 - 34344648x^5 + 42802020x^4 - 28362960x^3 + 10694484x^2 - 2237112x + 204228)y^{10} + (17955270x^8 - 71821080x^7 + 123219252x^6 - 118283976x^5 + 69737274x^4 - 26125848x^3 + 6281388x^2 - 962280x + 88209)y^8 + (17653464x^{10} - 88267320x^9 + 191561004x^8 - 236640096x^7 + 183225240x^6 - 92158128x^5 + 30035016x^4 - 6018624x^3 + 609444x^2)y^6 + (10657980x^{12} - 63947880x^{11} + 167328828x^{10} - 250455240x^9 + 235946682x^8 - 144481968x^7 + 56867832x^6 - 13401936x^5 + 1485702x^4)y^4 + (3621672x^{14} - 25351704x^{13} + 77551020x^{12} - 135733968x^{11} + 149197140x^{10} - 105973272x^9 + 47790324x^8 - 12597120x^7 + 1495908x^6)y^2 + 531441x^{16} - 4251528x^{15} + 14880348x^{14} - 29760696x^{13} + 37200870x^{12} - 29760696x^{11} + 14880348x^{10} - 4251528x^9 + 531441x^8$$

Now we consider the discriminant of  $P(y)$ , i.e., the following polynomial in the variable  $x$ .



```
> F(x) := discrim(P(y), y);
```

$$\begin{aligned}
 F(x) := & 280067312748789736985963225604373352510564033142263971832537392349 \\
 & 415195652578325220678822080410345331615448815562735192521704455798784x^{72} \\
 & (x^8 - 8x^7 + 28x^6 - 56x^5 + 70x^4 - 56x^3 + 28x^2 - 8x + 1) \\
 & (3632485367808x^{56} - 159829356183552x^{55} + 3577163194302464x^{54} \\
 & - 54164413774561280x^{53} + 622443578726023168x^{52} - 5773582936701927424x^{51} \\
 & + 44897775920459808768x^{50} - 300247901712809459712x^{49} \\
 & + 1758114864489495724032x^{48} - 9135260149961008349184x^{47} \\
 & + 42551623871336184741888x^{46} - 179090221864084504117248x^{45} \\
 & + 685344164455520353296384x^{44} - 2396629632755743126634496x^{43} \\
 & + 7689454585845088640053248x^{42} - 22708981881839500024713216x^{41} \\
 & + 61891840837200517187347200x^{40} - 155990296354557949507587072x^{39} \\
 & + 364158999313469028710248320x^{38} - 788406917587255927405708416x^{37} \\
 & + 1584428222011265587900156720x^{36} - 2957569256832435823271357152x^{35} \\
 & + 5129945603504315688350052152x^{34} - 8269776590784139495965972872x^{33} \\
 & + 12390543211614717631909801257x^{32} - 17252401206283398073500981408x^{31} \\
 & + 22318296998768405837652400560x^{30} - 26814078429820941216636836640x^{29} \\
 & + 29905088583572672158777405176x^{28} - 30942227685324623761835121056x^{27} \\
 & + 29681091982410789831813469808x^{26} - 26374203265941990164579799232x^{25} \\
 & + 21689407819853155907386334332x^{24} - 16490199124372028963640724512x^{23} \\
 & + 11576884004652233065735193904x^{22} - 7494649995207776536812979488x^{21} \\
 & + 4467144551988536607861767208x^{20} - 2447145125704270768169593824x^{19} \\
 & + 1229604833215458358583229216x^{18} - 565390666420461765033763728x^{17} \\
 & + 237282486568592921740430646x^{16} - 90615812927915371616847840x^{15} \\
 & + 31379801331933253034228112x^{14} - 9814109135651240497580640x^{13} \\
 & + 2759065273605140504442696x^{12} - 693397355113836911985120x^{11} \\
 & + 154761785959667058874512x^{10} - 30436116451158737790592x^9 \\
 & + 5223919673230693179772x^8 - 773249983840487534432x^7 \\
 & + 97228055482879826960x^6 - 10181010035441972448x^5 \\
 & + 863987640180241896x^4 - 57116611454625216x^3 + 2762134609728168x^2 \\
 & - 87043594688424x + 1344301719849)^2
 \end{aligned}$$

The output is a factorized polynomial  $F(x)$  in  $\mathbb{R}[x]$  that we can express in the form  $F(x) = F_0(x) \cdot F_1(x) \cdot F_2^2(x)$ . The first factor  $F_0(x)$  is a monomial containing  $x^{72}$ , which gives the real root  $x = 0$ . The two other factors  $F_1(x)$ ,  $F_2(x)$  (one of them squared) need a more careful examination. Let us see first whether they can be further factorized.

```
> factor(F_1); factor(F_2);
```

$$(x - 1)^8$$

$$\begin{aligned}
 & (221709312x^{10} - 1108546560x^9 + 5547202304x^8 - 15537529856x^7 + 23360155968x^6 \\
 & - 20355008960x^5 + 10731606224x^4 - 3448222560x^3 + 658900764x^2 - 70266636x \\
 & + 3485889)(4x^2 + 23)^2(4x^2 - 8x + 27)^2(2x - 1)^6(x - 1)^{32}
 \end{aligned}$$

Now it is clear that  $F_1(x) = 0$  has only one real solution  $x = 1$  (with multiplicity 8). With respect to  $F_2(x)$ , the factors  $4x^2 + 23$  and  $4x^2 - 8x + 27$  do not contribute with real roots, and the factors  $2x - 1$  and  $x - 1$  contribute with real solutions  $x = \frac{1}{2}$  and  $x = 1$  (with multiplicities 6 and 32 respectively). Finally, we study the Sturm sequence for the larger factor in order to see whether we are missing other real roots of  $F_2(x)$ .

```
> sturmseq(221709312*x^10 - 1108546560*x^9 + 5547202304*x^8
- 15537529856*x^7 + 23360155968*x^6 - 20355008960*x^5
+ 10731606224*x^4 - 3448222560*x^3 + 658900764*x^2
- 70266636*x + 3485889,x);
sturm(%,x,-infinity , infinity );
```

$$\left[ x^{10} - 5x^9 + \frac{21668759}{866052}x^8 - \frac{15173369}{216513}x^7 + \frac{121667479}{1154736}x^6 - \frac{28913365}{314928}x^5 + \frac{670725389}{13856832}x^4 - \frac{11972995}{769824}x^3 + \frac{6100933}{2052864}x^2 - \frac{59147}{186624}x + \frac{11737}{746496}, x^9 - \frac{9}{2}x^8 + \frac{21668759}{1082565}x^7 - \frac{106213583}{2165130}x^6 + \frac{121667479}{1924560}x^5 - \frac{28913365}{629856}x^4 + \frac{670725389}{34642080}x^3 - \frac{2394599}{513216}x^2 + \frac{6100933}{10264320}x - \frac{59147}{1866240}, -x^8 + 4x^7 - \frac{152575271}{23851348}x^6 + \frac{123806941}{23851348}x^5 - \frac{105485273}{47702696}x^4 + \frac{5223331}{11925674}x^3 - \frac{6179517}{381621568}x^2 - \frac{1656315}{381621568}x + \frac{68607}{1526486272}, -x^7 + \frac{7}{2}x^6 - \frac{6032902658003487}{1200060777167108}x^5 + \frac{9163449689593045}{2400121554334216}x^4 - \frac{7901689078731841}{4800243108668432}x^3 + \frac{3852119358162855}{9600486217336864}x^2 - \frac{246472468139943}{4800243108668432}x + \frac{52410697894269}{19200972434673728}, -x^6 + 3x^5 - \frac{1923113575493489063067}{544181504190477281099}x^4 + \frac{1125319630034591720639}{544181504190477281099}x^3 - \frac{2699431145391007808109}{4353452033523818248792}x^2 + \frac{374878642014550049949}{4353452033523818248792}x - \frac{129123207926862901389}{34827616268190545990336}, -x^5 + \frac{5}{2}x^4 - \frac{6120084968285261526685869}{5247608252952953776114636}x^3 - \frac{7877786359908984300515573}{10495216505905907552229272}x^2 + \frac{297440786451944785973148517}{440799093248048117193629424}x - \frac{113685261871877652900325591}{881598186496096234387258848}, x^4 - 2x^3 + \frac{83459519425974095718301535996311}{55852716316393303993363363418712}x^2 - \frac{27606803109580791724938172577599}{55852716316393303993363363418712}x + \frac{6814525055230976067861308675045}{111705432632786607986726726837424}, -x^3 + \frac{3}{2}x^2 - \frac{264697672012697346416360283393844501}{357631291633235791535031047332757050}x + \frac{21470506549019862662211189931866494}{178815645816617895767515523666378525}, -x^2 + x - \frac{27892930064141325504042418797332472427}{11908197738863253638405327183317787758}, x - \frac{1}{2}, -1 \right]$$

0

Therefore, no new real roots appear, and we conclude that the only real roots of the discriminant  $F(x)$  are 0,  $\frac{1}{2}$  and 1. These three values split  $\mathbb{R}$  into four intervals and on their interiors  $F(x)$  does not vanish. Now the following result concerning discriminants becomes handy (see for example [51], (pp. 320–321) for a proof of this fact):

**Proposition 2.** Between consecutive roots of the discriminant  $F(x)$  of a bivariate polynomial  $p(x, y)$  (regarded as a function of  $y$  alone), the number of real zeros of  $p(x, y)$  is independent of  $x$ .

Therefore, we can use again Sturm's sequences to check the number of real roots for the variable  $y$  for specifying particular values of  $x$  lying in each of the aforementioned intervals, let us say  $x_1 = -1$ ,  $x_2 = \frac{1}{3}$ ,  $x_3 = \frac{2}{3}$  and  $x_4 = 2$ .

> subs(x=-1,P(y)); Sturmseq(%,y): Sturm(%,y,-infinity , infinity );

$$88209y^{16} + 2128788y^{14} + 22521286y^{12} + 130093668y^{10} + 434474577y^8 + 846168336y^6 + 944574048y^4 + 559312128y^2 + 136048896$$

0  
> subs(x=1/3,P(y)); Sturmseq(%,y): Sturm(%,y,-infinity , infinity );

$$88209y^{16} - 9612y^{14} - 41114y^{12} - 892y^{10} + \frac{293419}{27}y^8 + \frac{1093808}{243}y^6 + \frac{630112}{729}y^4 + \frac{20224}{243}y^2 + \frac{256}{81}$$

0  
> subs(x=2/3,P(y)); Sturmseq(%,y): Sturm(%,y,-infinity , infinity );

$$88209y^{16} - 9612y^{14} - 41114y^{12} - 892y^{10} + \frac{293419}{27}y^8 + \frac{1093808}{243}y^6 + \frac{630112}{729}y^4 + \frac{20224}{243}y^2 + \frac{256}{81}$$

0  
> subs(x=2,P(y)); Sturmseq(%,y): Sturm(%,y,-infinity , infinity );

$$88209y^{16} + 2128788y^{14} + 22521286y^{12} + 130093668y^{10} + 434474577y^8 + 846168336y^6 + 944574048y^4 + 559312128y^2 + 136048896$$

Therefore, for any  $x_0$  outside the finite set  $\{0, \frac{1}{2}, 1\}$  there are no real solutions for  $P(x_0, y) = 0$ , and we can conclude that  $A(0, 0)$ ,  $B(1, 0)$ ,  $C_1(\frac{1}{2}, \frac{\sqrt{3}}{2})$  and  $C_2(\frac{1}{2}, -\frac{\sqrt{3}}{2})$  are the only real points of the curve  $P(x, y) = 0$ , and this concludes our scaffolded, computational-based proof of Proposition 1.

Certainly, there are still people loving human-made proofs. If the reader is among them and is curious about a more classical approach, see [10] or [52].

## 7. Conclusions and Further Research

We have seen that the current graphic and symbolic tools available in diverse mathematical software packages, due to high complexity issues, present some limitations for visualizing appropriately geometric objects such as loci defined by algebraic curves. In order to gain some insight on the geometry of these objects we have resorted to an alternative visualization technique, the dynamic color scanning method, which takes advantage of the possibility of working with dynamic colors in the GeoGebra graphic view. The visual guidance provided by this method allows us to approach the extreme case of a geometric inequality more effectively, this time using symbolic tools available in external software such as Maple.

The authors consider that future releases of GeoGebra could benefit from the inclusion of new commands that:

- (i) perform straightforwardly a dynamic color scanning, avoiding the setup process which is needed at present and improving its performance;
- (ii) utilize the Giac and Tarski/QEPCAD embedded systems to increase the power of the GART's tools present in GeoGebra, rendering unnecessary to make use of external software to visualize and study certain geometric problems (see for instance [53]) which can involve equalities or inequalities in their formulation.

We hope to accomplish these tasks in a near future, with the help of the many colleagues already involved in the development of the GeoGebra automated reasoning tools project.

**Author Contributions:** Writing—original draft, T.R., R.L., Z.K. and C.U. All authors contributed equally to this work. All authors have read and agreed to the published version of the manuscript.

**Funding:** First and third authors were partially supported by a grant PID2020-113192GB-I00 (Mathematical Visualization: Foundations, Algorithms and Applications) from the Spanish MICINN.

**Acknowledgments:** Thanks to Laureano González-Vega for some clever hints concerning Section 6.2.

**Conflicts of Interest:** The authors declare no conflict of interest.

### Abbreviations

The following abbreviations are used in this manuscript:

CAD	Cylindric Algebraic Decomposition
CAS	Computer Algebra System
DGS	Dynamic Geometry Software
GART	Geometry Automated Reasoning Tool
GATP	Geometry Automated Theorem Prover

### References

- Wu, W.T. On the decision problem and the mechanization of theorem-proving in elementary geometry. *Sci. Sin.* **1978**, *21*, 159–172. Reprinted in: *Automated Theorem Proving: After 25 Years*; Bledsoe, W.W., Loveland, D.W., Eds.; Contemporary Mathematics, 29; AMS: Providence, RI, USA, 1984; pp. 213–234.
- Quaresma, P. Thousands of Geometric problems for geometric Theorem Provers (TGTP). In *Automated Deduction in Geometry, 8th International Workshop, ADG 2010*; Schreck, P., Narboux, J., Richter-Gebert, J., Eds.; LNAI 6877; Springer: Berlin/Heidelberg, Germany, 2011; pp. 169–181.
- Chou, S.C. An Introduction to Wu’s Method for Mechanical Theorem Proving in Geometry. *J. Autom. Reason.* **1988**, *4*, 237–267. [CrossRef]
- Buchberger, B. Introduction to Gröbner Bases. In *Gröbner Bases and Applications*; Buchberger, B., Winkler, F., Eds.; LMSLN 251; Cambridge University Press: Cambridge, UK, 1998; pp. 3–31.
- Kapur, D. Using Gröbner bases to reason about geometry problems. *J. Symb. Comput.* **1986**, *2*, 399–408. [CrossRef]
- About GeoGebra. Available online: <https://www.geogebra.org/m/pR5DME5S#material/uy93nfzr> (accessed on 21 August 2021).
- Cevikbas, M.; Kaiser, G. A Systematic Review on Task Design in Dynamic and Interactive Mathematics Learning Environments (DIMLEs). *Mathematics* **2021**, *9*, 399. [CrossRef]
- Ondes, R.N. Research trends in dynamic geometry software: A content analysis from 2005 to 2021. *World J. Educ. Technol. Curr. Issues* **2021**, *13*, 236–260. [CrossRef]
- Kovács, Z.; Montag, A.; Vajda, R. On Euler’s inequality and automated reasoning with dynamic geometry. *arXiv* **2020**, arXiv:1708.02993.
- Bottema, O.; Djordjević, R.Ž.; Janić, R.R.; Mitrinović, D.S.; Vasić, P.M. *Geometric Inequalities*, 1st ed.; Wolters-Noordhoff Publishing: Groningen, Germany, 1969.
- Hohenwarter, M.; Kovács, Z.; Recio, T. Using GeoGebra automated reasoning tools to explore geometric statements and conjectures. In *Proof Technology in Mathematics Research and Teaching*; Hanna, G., de Villiers, M., Reid, D., Eds.; Mathematics Education in the Digital Era 14; Springer: Berlin/Heidelberg, Germany, 2019; pp. 215–236.
- Recio, T.; Vélez, M.P. Automated Discovery of Theorems in Elementary Geometry. *J. Autom. Reason.* **1999**, *23*, 63–82. [CrossRef]
- Botana, F.; Kovács, Z.; Recio, T. A Mechanical Geometer. *Math. Comput. Sci.* **2020**. doi: 10.1007/s11786-020-00497-7. [CrossRef]
- De Graeve, R.; Parisse, B. Giac/Xcas (v. 1.7.0). 2021. Available online: <https://www-fourier.ujf-grenoble.fr/~parisse/giac.html> (accessed on 29 July 2021).
- Kovács, Z.; Parisse, B. Giac and GeoGebra—Improved Gröbner Basis Computations. In *Computer Algebra and Polynomials*; Gutiérrez, J., Schicho, J., Weimann, M., Eds.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2015; pp. 12–138.
- Cox, D.A.; Little, J.; O’Shea, D. *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra*, 3rd ed.; Springer: Cham, Germany, 2015.
- Conti, P.; Traverso, C. Algebraic and semialgebraic proofs: Methods and paradoxes. In *Proc. 3rd International Workshop on Automated Deduction in Geometry (ADG 2000)*; Richter-Gebert, J., Wang, D., Eds.; LNAI, 2061; Springer: Berlin/Heidelberg, Germany, 2000; pp. 83–103.

18. Dolzmann, A.; Sturm, T.; Weispfenning, V. A New Approach for Automatic Theorem Proving in Real Geometry. *J. Autom. Reason.* **1998**, *21*, 357–380. [\[CrossRef\]](#)
19. De Villiers, M. Clough's conjecture: A Sketchpad investigation. In *Proceedings of the 10th Annual National Congress of the Association for Mathematics Education of South Africa*; Nieuwoudt, S., Froneman, S., Nkhoma, P., Eds.; AMESA: Potchefstroom, South Africa, 2004; Volume 2, pp. 52–56.
20. De Villiers, M. An illustration of the explanatory and discovery functions of proof. *Pythagoras* **2012**, *33*, 8. <http://dx.doi.org/10.4102/pythagoras.v33i3.193>. [\[CrossRef\]](#)
21. Kovács, Z.; Recio, T.; Sólyom-Gecse, C. Rewriting input expressions in complex algebraic geometry provers. *Ann. Math. Artif. Intell.* **2018**, *85*, 73–87. [\[CrossRef\]](#)
22. Chou, S.C. *Mechanical Geometry Theorem Proving*, 1st ed.; D. Reidel Publishing Company: Dordrecht, Holland, 1988.
23. Kovács, Z.; Recio, T.; Tabera, L.F.; Vélez, M.P. Dealing with Degeneracies in Automated Theorem Proving in Geometry. *Mathematics* **2021**, *9*, 1964. [\[CrossRef\]](#)
24. Kapur, D. A Refutational Approach to Geometry Theorem Proving. *Artif. Intell.* **1988**, *37*, 61–93. [\[CrossRef\]](#)
25. Bulmer, M.; Fearnley-Sander, D.; Stokes, T. The kinds of truth of geometric theorems in automated deduction in geometry. In *Proc. 3rd International Workshop on Automated Deduction in Geometry (ADG 2000)*; Richter-Gebert, J., Wang, D., Eds.; LNAI, 2061; Springer: Berlin/Heidelberg, Germany, 2000; pp. 129–142.
26. Guan, H.; Rao, Y.S.; Zhang, J.Z.; Cao, S.; Qin, X.L. Method for Processing Graph Degeneracy in Dynamic Geometry Based on Domain Design. *J. Comput. Sci. Technol.* **2021**, *36*, 910–921. [\[CrossRef\]](#)
27. Kovács, Z.; Recio, T.; Vélez, M.P. Detecting truth, just on parts. *Rev. Matemática Complut.* **2019**, *32*, 451–474. [\[CrossRef\]](#)
28. Kovács, Z. The Relation Tool in GeoGebra 5. In *Proceedings of the 10th International Workshop on Automated Deduction in Geometry (ADG 2014)*, Coimbra, Portugal, 9–11 July 2014; Botana, F., Quaresma, P., Eds.; Springer: Berlin/Heidelberg, Germany, 2015; pp. 53–71.
29. Botana, F.; Hohenwarter, M.; Janičić, P.; Petrović, I.; Recio, T.; Weitzhofer, S. Automated Theorem Proving in GeoGebra: Current Achievements. *J. Autom. Reason.* **2015**, *55*, 39–59. [\[CrossRef\]](#)
30. GeoGebra Discovery. Available online: <https://github.com/kovzol/geogebra-discovery> (accessed on 29 July 2021).
31. Bochnak, J.; Coste, M.; Roy, M.F. *Real Algebraic Geometry*, 1st ed.; Springer: Berlin/Heidelberg, Germany, 1998.
32. Davenport, J.; Heintz, J. Real quantifier elimination is doubly exponential. *J. Symb. Comput.* **1988**, *5*, 29–35. [\[CrossRef\]](#)
33. Xia, B.; Yang, L.; Hou, X. Automated discovering and proving for geometric inequalities. In *ADG'98: Proceedings of the Second International Workshop on Automated Deduction in Geometry, Beijing, China, 1–3 August 1998*; Gao, X.-S., Wang, D., Yang, L., Eds.; Springer: Berlin/Heidelberg, Germany, 1998.
34. Xia, B.; Yang, L. *Automated Inequality Proving and Discovering*; World Scientific Publishing Company: Hackensack, NJ, USA, 2017.
35. Yang, L. Recent advances in automated theorem proving on inequalities. *J. Comput. Sci. Technol.* **1999**, *14*, 434–446. [\[CrossRef\]](#)
36. Collins, G.E. Quantifier elimination for real closed fields by cylindrical algebraic decomposition. In *Proc. 2nd GI Conference on Automata. Theory and Formal Languages*; Brakhage, H., Ed.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 1975; Volume 33, pp. 134–183.
37. Chen, C.; Maza, M.M. Quantifier Elimination by Cylindrical Algebraic Decomposition based on Regular Chains. *J. Symb. Comput.* **2016**, *75*, 74–93. [\[CrossRef\]](#)
38. Kovács, Z.; Vajda, R. GeoGebra and the realgeom reasoning tool. In *Proceedings of the Fifth International Workshop on Satisfiability Checking and Symbolic Computation*, Paris, France, 5 July 2020; Fontaine, P., Korovin, K., Kotsireas, I.S., Rümmer, P., Tourret, S., Eds.; CEUR Workshop Proceedings; RWTH: Aachen, Germany, 2020; Volume 2752, pp. 204–219.
39. Brown, C.W. QEPCAD. Extended Tarski Formulas. Available online: <https://www.usna.edu/Users/cs/wcbrown/qepcad/B/user/ETF.html> (accessed on 12 August 2021).
40. Brown, C.W.; Kovács, Z.; Vajda, R. Supporting Proving and Discovering Geometric Inequalities in GeoGebra by Using Tarski. In *Proceedings of the 13th International Workshop on Automated Deduction in Geometry (ADG2021)*, Hagenberg, Austria, 15–17 September 2021.
41. Brown, C.W. An Overview of QEPCAD B: A Tool for Real Quantifier Elimination and Formula Simplification. *J. JSSAC* **2003**, *10*, 13–22.
42. Cheng, J.; Lazard, S.; Peñaranda, L.; Pouget, M.; Rouillier, F.; Tsigaridas, E. On the topology of real algebraic plane curves. *Math. Comput. Sci.* **2010**, *4*, 113–137. [\[CrossRef\]](#)
43. Chen, C.; Wu, W.; Feng, Y. Visualizing Planar and Space Implicit Real Algebraic Curves with Singularities. *J. Syst. Sci. Complex.* **2020**, *33*, 1252–1274. [\[CrossRef\]](#)
44. Losada, R. Color Dinámico (Spanish). Available online: <https://www.geogebra.org/m/d6j2nhYG> (accessed on 19 July 2021).
45. GeoGebra Manual. Available online: <https://wiki.geogebra.org/en/Manual> (accessed on 23 August 2021).
46. Losada, R. El Color Dinámico de GeoGebra (Spanish). *Gaceta De La Real Sociedad Matematica Espanola* **2014**, *17*, 525–547.
47. Losada, R.; Recio, T.; Valcarce, J.L. Equal Bisectors at a Vertex of a Triangle. In *Computational Science and Its Applications—ICCSA 2011*; Murgante, B., Gervasi, O., Iglesias, A., Taniar, D., Apduhan, B.O., Eds.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2011; Volume 6785, pp. 328–341.
48. Cinderella Homepage. Available online: <https://cinderella.de/tiki-index.php> (accessed on 20 August 2021).



- 
49. CindyJS. Available online: <https://cindyjs.org/> (accessed on 22 August 2021).
  50. Montag, A.; Richter-Gebert, J. Bringing Together Dynamic Geometry Software and the Graphics Processing Unit. *arXiv* **2018**, arXiv:1808.04579.
  51. Schwartz, J.T.; Sharir, M. On the “Piano Movers” Problem. II. General Techniques for Computing Topological Properties of Real Algebraic Manifolds. *Adv. Appl. Maths.* **1983**, *4*, 298–351. [[CrossRef](#)]
  52. Bottema, O. Inequalities for R, r and s. Publikacije Elektrotehničkog Fakulteta. *Seriya Matematika i Fizika* **1971**, *338/352*, 27–36.
  53. Kovács, Z.; Recio, T.; Vélez, M.P. Merging Maple and GeoGebra Automated Reasoning Tools. In *Maple in Mathematics Education and Research*; Corless, R.M., Gerhard, J., Kotsireas, I., Eds.; Communications in Computer and Information Science; Springer Nature: Cham, Switzerland, 2021.