




Article

# A Computational Approach to the Perimeter-Area Inequality in a Triangle

Tomás Recio <sup>1,†</sup> , Carlos Ueno <sup>2,†</sup>  and María Pilar Vélez <sup>1,\*,†</sup> 

<sup>1</sup> Departamento de Matemáticas y Física, Escuela Politécnica Superior, Universidad Antonio de Nebrija, 28015 Madrid, Spain; trecio@nebrija.es

<sup>2</sup> Centro de Educación a Distancia Profesor Félix Pérez Parrilla, 35005 Las Palmas de Gran Canaria, Spain; cuenjac@cead-laspalmas.net

\* Correspondence: pvelez@nebrija.es

† These authors contributed equally to this work.

**Abstract:** This paper explores the application of automated reasoning tools, specifically those implemented in GeoGebra Discovery, to the perimeter-area inequality in triangles. Focusing on the computational complex and real algebraic geometry methods behind these tools, this study analyzes a geometric construction involving a triangle with arbitrary side lengths and area, investigating the automated derivation of the relationship between the area and perimeter of a triangle, and showing that only equilateral triangles satisfy the exact perimeter-area equality. The main contribution of this work is to describe the challenges, and potential ways to approach their solutions, still posed by the use of such automated, symbolic computation, methods in dynamic geometry, in particular concerning the discovery of loci of points that satisfy specific geometric conditions, suggesting relevant improvements for the future development of these symbolic AI-based educational tools in geometry.

**Keywords:** perimeter-area inequality; triangle inequality; GeoGebra Discovery; Maple; computer algebra system; automatic reasoning tools; locus computation; dynamic geometry software

**MSC:** 51M04; 14Q10; 14P05



Academic Editors: Ruizhi Yang and Yining Xie

Received: 1 December 2024

Revised: 30 December 2024

Accepted: 2 January 2025

Published: 5 January 2025

**Citation:** Recio, T.; Ueno, C.; Vélez, M.P. A Computational Approach to the Perimeter-Area Inequality in a Triangle. *Axioms* **2025**, *14*, 40. <https://doi.org/10.3390/axioms14010040>

**Copyright:** © 2025 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

The aim of this work is to present and analyze some of the issues that arise when approaching Euclidean geometry problems by means of the computational algebraic geometry algorithms that are behind the automated reasoning (aka ART, in what follows) tools of the popular, freely available, dynamic mathematics software GeoGebra (see [1]).

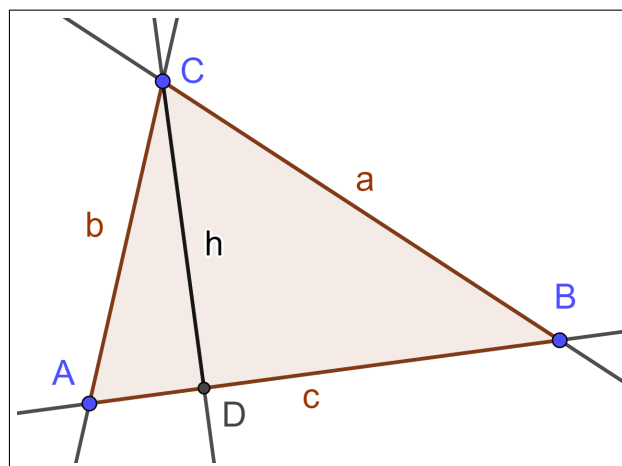
We start with a quite simple construction: a triangle  $ABC$ , with side lengths  $a = BC$ ,  $b = AC$ , and  $c = AB$ . In addition, we consider the length  $h = CD$  as the altitude of the triangle from the vertex  $C$  to the opposite side  $AB$ , where point  $D$  is the foot of the perpendicular drop from  $C$  to  $AB$  (Figure 1). Our investigation will revolve around the following problem:

**Problem 1.** *What is the relationship between the area and the perimeter of a generic triangle?*

More precisely, in this work we address, through the automated reasoning tools we have developed in GeoGebra, and expanded in the fork, experimental version GeoGebra

Discovery (aka GGD, in what follows) (see [2]), the discovery of the general inequality that holds between these two magnitudes (area, perimeter), that is, one that applies to any Euclidean triangle, and, as our main objective, to analyze for which triangles the corresponding equality holds.

Let us emphasize here that the goal of our work is not to contribute to the solution of Problem 1, a classical, well-known question, as we will document in the next paragraphs, but rather to use this elementary context to experiment, understand, and exhibit mathematical solutions to singular issues that arise in some special cases, concerning the performance of these GGD automated reasoning tools.



**Figure 1.** The basic construction for a generic triangle in GeoGebra.

Tools that are summarily introduced in Sections 2 and 2.1, mentioning that presently they include not only computer algebra algorithms to approach geometric queries through complex algebraic geometry methods, as is the case of the basic GeoGebra-ART tools [1], but also real geometry algorithms that can deal with algebraic inequalities through real quantifier elimination. In particular, we will exemplify in that Section their performance by asking GGD to answer Problem 1. GGD immediately outputs (less than 10 s using the GGD app GGD–2024Nov05 version, based on GeoGebra Classic 5.0.641.0-d and on the software Tarski [3], version 1.37, 19 October 2023), on a personal laptop, that  $(a + b + c)^2 \geq 6\sqrt{3} ch$ , see Figure 1) is the perimeter-area inequality.

We consider that this inequality is both non-trivial and well known to geometers. Thus, it appears as inequality 4.2 in the classical book [4] (where we refer the interested reader to find a “human-made” approach, and further references on this topic), which corresponds to the perimeter-area inequality that GGD ART has quickly discovered. In this book [4], the inequality is traced back to H. Hadwiger [5] and to L.A. Santaló [6]. But in the realm of classic plane geometry, it is often difficult to pursue a geometric statement up to its authentic origins. Indeed, [5] is not exactly the perimeter-area inequality, but a refinement, and [6] provides proof of this inequality in the context of the isoperimetric inequality, which is, in fact, another form of stating the perimeter-area inequality for triangles.

Detailed, basic information about the classic isoperimetric inequality, formulated, for example, as “among all curves in the plane with a given perimeter, find the one enclosing the greatest area”, appears in Chapter 18, pp. 104–108 (<https://doi.org/10.5948/UPO9780883859537.020> (accessed on 1 January 2025)) of [7]. Likewise, Chapter 3, pp. 13–17 (<https://doi.org/10.5948/UPO9780883859537.005> (accessed on 1 January 2025)), of the same book, brings information about this inequality for the case of triangles. Moreover, we can remark here that the isoperimetric inequality has attracted a lot of interest in various forms from ancient times, and we can connect the specific case of triangles to the work by

Zenodorus on the polygons that maximize their areas for a fixed perimeter [8]. All these illustrious precedents contribute, in some sense, to support our appreciation of the interest in exemplifying the approach to this inequality with automated reasoning tools.

Although there is now an extensive body of literature on computational methods for proving geometric statements (see, for example, [9,10]), our contribution stands out in two key aspects. First, we address the perimeter-area inequality and explore the equality case entirely through automated means. Second, we achieve this using GeoGebra Discovery, a widely accessible application available even on smartphones. Second, we analyze the mathematical, intrinsic, reasons related to the algebraic geometry framework that are behind the apparent inconsistencies or counterintuitive outputs that these automated methods produce in some singular statements, such as in this perimeter-area equality.

This second goal can be perceived as part of the ongoing work to enhance the algorithmic approaches implemented in GeoGebra Discovery, trying to improve the outputs that can appear when dealing with inputs that involve real algebraic geometry issues (as in this case, dealing with segment lengths), while addressing them with much better performing, complex algebraic geometry algorithms that GeoGebra Discovery uses for locus computation.

In fact, it is worth noting that, in general, GGD-ART uses the `Relation(F, G)` command for finding the ratio  $m = F/G$  between two geometric objects or expressions, and this ratio is obtained by performing first a standard elimination over the  $m$  variable of the ideal generated by the construction equations and by  $F - m \cdot G = 0$ . The output of such elimination is a collection of equations generating the elimination ideal. This set of equations defines the constraints that must be satisfied by the possible values of the ratio  $m$  (e.g.,  $m^2 - 1 = 0$ , thus  $m = 1$  or  $m = -1$ ). But, if the elimination does not yield a sound result, a real quantifier elimination (RQE) is performed (see [11] for details), to detect a possible interval describing  $m$ . As ideal elimination is usually much quicker than RQE, choosing elimination first is a kind of heuristics, but a good performing one (see, for instance, the comparison benchmark at [https://prover-test.risc.jku.at/job/GeoGebra\\_Discovery-comparetest/lastSuccessfulBuild/artifact/fork/geogebra/test/scripts/benchmark/compare/html/all.html](https://prover-test.risc.jku.at/job/GeoGebra_Discovery-comparetest/lastSuccessfulBuild/artifact/fork/geogebra/test/scripts/benchmark/compare/html/all.html) (accessed on 1 January 2025)), for a large list of successful outputs of the `Relation` command, each formulating a non-trivial equality holding between two given expressions involving side lengths of polygons.

Finally, the advantages of using elimination vs. RQE are also behind the current GeoGebra and GGD algorithms for `LocusEquation`, which find the locus of a certain point subject to some constraints, only through the elimination of the ideal generated by these conditions, over the variables representing the coordinates of the selected point. The absence of an alternative RQE algorithm for computing the `LocusEquation` when real algebraic geometry is clearly in context (e.g., when dealing with lengths) is not just a question of speed, but it is also related to the potential complexity and difficulty of interpretation of the output of an RQE in a dimension two space.

In summary, our main topic of study here is not exactly related to the perimeter-area inequality, but to investigate with GGD, and thus through `LocusEquation()` and elimination algorithms, in a complex algebraic geometry context, the cases in which this inequality becomes an equality, trying to gain understanding that could be exported to other instances, about the way to interpret complex algebraic geometry outputs to real geometry locus equation inputs.

The structure of this paper is as follows: In Section 2 we introduce GGD-ART and describe in some detail, through simple examples, the inherent difficulties concerning locus computation, both for the complex and the real geometry settings. In Section 3 we initiate the approach to the perimeter-area equality locus computation in the standard computer

algebra way, i.e., with the coordinates of the vertices as main variables of our computation, yielding quite unexpected results that are thoroughly analyzed there. In Section 4 we develop an alternative approach, still using elimination algorithms, in a more coordinate-free way, focusing on the sides of  $\triangle ABC$  as main working variables. An approach that ends up being related to the MEP polynomials we will mention in Section 2.2. Thus, our work contributes to supporting the relevance of MEP as a major tool for the interpretation of outputs obtained within a complex algebraic geometry framework when searching for real geometry locus regarding equality of expressions dealing with interval lengths. The last Section of conclusions includes some final reflections on the new ways that ART, such as those implemented in GGD, can play when tackling geometry problems of this kind, either in a research, or in an educational environment.

## 2. GeoGebra Discovery and Locus Computation

### 2.1. GeoGebra Discovery

GeoGebra Discovery (GGD) is a freely available program that can be used on a wide variety of devices such as smartphones, tablets, laptops, computers, etc. GGD is accessible in several offline release versions [12] for different OS (Mac, Windows, Linux, RaspberryPi), as well as in online versions [13]. As a fork version of GeoGebra (that already integrates a Dynamic Geometry (DGS) and a Computer Algebra System (CAS), see Section 3 in [14]), it is a sort of “dynamic mathematics” software, including dynamic geometry, calculus, plotting, statistics, etc. features, and, more relevant here, embedding the Giac computer algebra system [15] (to deal with polynomial equalities in a complex algebraic setting) and the Tarski system (to deal with inequalities, and possibly also with equalities from the real geometry point of view) which incorporates real algebraic geometry tools such as real quantifier elimination RQE and cylindrical algebraic decomposition (CAD). See [16] and the references therein for more details on the integration of Tarski and GGD.

Both CAS, Giac and Tarski, are behind GeoGebra Discovery automated reasoning tools (ART) for dealing with different questions involving elements on a construction implemented through the dynamic geometry features of GGD, and displayed in the graphics window of the program. A typical session using GGD-ART begins by drawing in the GGD graphics window a geometric construction, and then dragging the free objects created while observing some geometric facts that seem to hold in all cases, or while trying to find a specific position for some points in the figure for a certain property to be verified. Then, as a result of this inspection, the user is prone to formulate the following questions:

- Does this conjectured property hold between these elements?
- What property, if any, holds between these elements?
- Show properties that hold true, involving this particular element of the construction.
- Show all properties of a certain kind (e.g., parallelism) that are satisfied by all the elements of the figure.
- Where should I place this point, so that a certain property holds true on the resulting figure?

To deal with such queries, the algorithms behind the automated reasoning tools in GeoGebra Discovery begin, very roughly speaking, by assigning to the initially given points some symbolic coordinates, and then translating the different steps of the construction (hypotheses) and the posed questions (thesis) into algebraic equations with these symbolic coordinates as variables. The corresponding polynomials are then considered as generators of certain commutative algebraic ideals (hypotheses, thesis ideals). Subsequently, the solution of the above-mentioned questions is formulated in terms of the projection/elimination of certain variables in ideals that are built performing some operation with the hypotheses

and thesis ideals (e.g., adding both ideals, or adding to the hypotheses ideal the negation of the thesis, etc.).

More specifically, GGD includes the following ART tools and commands, that we enumerate here in an order corresponding to the above proposed questions:

- **Prove** and **ProveDetails** commands: to prove the truth or failure of a conjectured or given statement formulated as an algebraic equality or inequality. It includes the automated formulation of non-degeneracy conditions that are required for the statement to be true, obtained (roughly speaking) by elimination of the ideal of hypotheses plus the negation of the thesis over the set of independent variables with respect to the hypotheses ideal, i.e., the free coordinates of the points in the construction. The enhanced statement with the non-degeneracy conditions is, then, true over the components of the hypotheses ideal where the independent variables remain so, that are considered as collecting the non-degenerate instances of the statement. See [17], subsection 1.3, and [18], chapter 3, section 3 (“Formulation F3”), for detailed discussions about the elusive (and many times not-intuitive) concept of degeneracy, precise definitions, algorithmic proposals, and proofs of the related mathematical statements.
- **Relation** tool and command: to automatically discover the relation holding among some concrete elements of the given figure. For example, this point lies in the line defined by these other two, these two lines are perpendicular, etc. Or, these two expressions involving quantities are equal, or they are not equal but they have such ratio between them, or their ratio verifies this inequality, etc. The first output of the command (expressing the relation holding at least numerically) is followed by the verification of the symbolic, general truth (except for degenerate cases) of the formulated statement, via the internal performance of algorithms similar to those of the **Prove**, **ProveDetails** commands.
- **Discover** tool and command: to automatically find all statements (of a certain type, such as equality of lengths, perpendicularity, etc.) holding true and involving one element in the figure selected by the user.
- **ShowProof** command: it displays in the GeoGebra Discovery CAS window the different algebraic steps internally performed by the program to confirm or deny a given statement. It also provides a *measure of the complexity* of the proved statement. Currently available only for statements involving equalities.
- **LocusEquation** command (with two variants): roughly speaking, its output describes the equations that should be verified by a selected point on a figure that it is assumed to be placed corresponding either to the different positions of another point in the figure, or in such a way that the figure observes a given property. In this latter case, the output (that can be interpreted as yielding necessary conditions) contributes to the discovery of some sufficient conditions for the truth of such property. For example, given an arbitrary triangle  $\triangle ABC$ , find where to place  $C$  so that the triangle is right at  $C$ . Further details about this command are provided in the next Section 2.2.

If the commands **Relation** or **Discover** provide (possibly) unknown thesis actually holding on a figure verifying certain hypotheses (represented by the constraints involved in the steps of its geometric construction), and **Prove** or **ProveDetails** confirm (through a proof-certificate exhibited by the **ShowProof** command) or deny thesis conjectured by humans, we can say that the **LocusEquation** suggests new hypotheses needed for the truth of a thesis that humans want to achieve. In this way, the output of the **LocusEquation** command should be part of the formulation of a new statement, whose truth has to be verified using the **Prove** or **ProveDetails** commands.

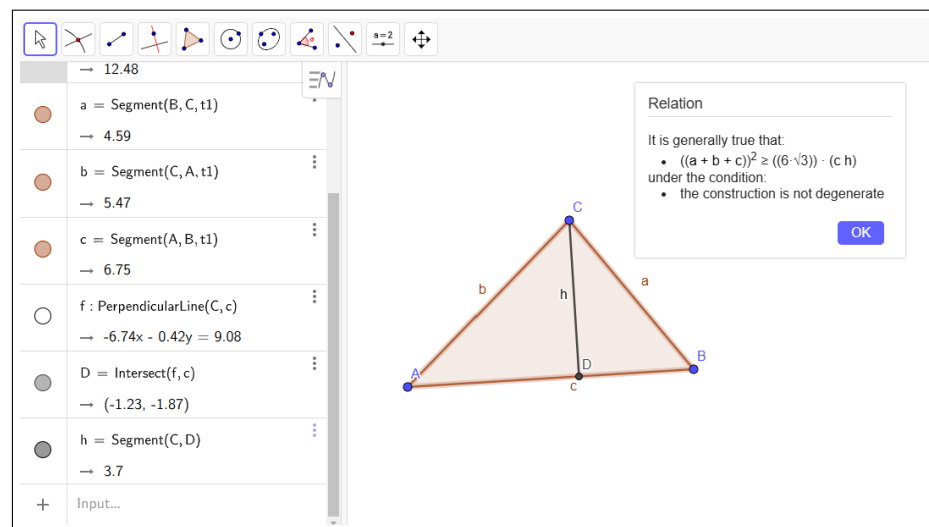
In summary: **Relation** and **Discover** provide new thesis derived from a given set of hypotheses, while the **LocusEquation** proposes new hypotheses contributing to the

possible truth of a considered thesis over the figure satisfying such hypotheses, a truth that should be confirmed using Prove or ProveDetails. Let us finally remark that, since the LocusEquation suggests changes in the initial construction, and the concept of degeneracy of a statement is bound to the final geometric figure, and the independent variables of its algebraic description, the LocusEquation does not provide any non-degeneracy information, although the user can intuitively consider that some of the conditions described by the LocusEquation output could be regarded as degenerate cases (e.g., equation  $g_1$  in Section 3).

The reader is referred to [17,19] for the foundational algorithmic protocols involved in these GGD tools in the realm of computational complex algebraic geometry and to [11,16] for some recent developments in the context of real algebraic geometry, which allows the manipulation of geometric inequalities. Further references related to each specific command can be found in the documentation located on the main developer's GitHub project site [2]).

## 2.2. Locus Computation

In order to illustrate the performance of these GGD-ART in the specific issues we are dealing with in this paper, let us address GGD-ART Problem 1, about finding, if there is one, the relationship between the area and the perimeter of a generic triangle. A quick consideration of the list of tools and commands described above, leads us to conclude that the Relation command seems a good way to discover the existence of any possible relation. Thus, we start by drawing a triangle  $ABC$  with side lengths  $a, b, c$  and its altitude  $h$  (see Figure 1), then we introduce in GeoGebra Discovery command line the text `Relation( $a + b + c, c \cdot h$ )`, and we obtain the well-known inequality between the area and the perimeter  $(a + b + c)^2 \geq 6\sqrt{3} \cdot c \cdot h$  (see Figure 2).

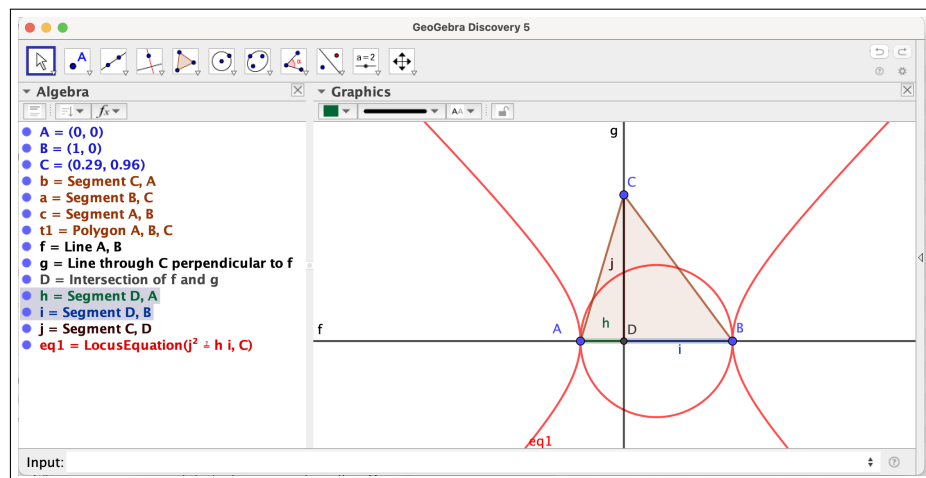


**Figure 2.** The relationship between the area and the perimeter of a generic triangle.

Yet, as mentioned in the Introduction Section in this paper we are particularly interested in the LocusEquation command, for several reasons. It is well known that locus computation is a characteristic feature of Dynamic Geometry programs, much before the consideration of this command as part of automated reasoning tools. See [20] or [21], for a survey of the locus computation historical relation with DGS, as well as a detailed description of current issues concerning DGS and CAS cooperation for locus computation. As already mentioned, locus computation is available in GGD through the LocusEquation command which calculates the equation of the locus, and plots it as an implicit curve. It includes two kinds of approaches:

- The classical, or explicit, locus computation (the syntax is `LocusEquation(<Tracer point Q>, <Mover point P>)`), that computes the locus by using as input a number of different positions of  $P$  within the construction, and the corresponding positions of  $Q$  (a point depending on  $P$ ), and adjusting—as output—an implicit curve that includes all the obtained positions of  $Q$ , a curve that, because of this sampling methodology, cannot guarantee the inclusion of all the possible locations of the tracer for any mover point input within the constraints of the figure where both points lay.
- The implicit one, that calculates the constraints that a selected point  $Q$  must verify, assuming that a certain boolean condition holds in the given construction. The syntax is `LocusEquation(<Boolean expression>, <Point Q>)`.

As already described in the previous section, this second locus computation command is a basic ingredient in the automated reasoning tools. Again, as a simple but illustrative example of this feature, let us consider a triangle  $\triangle ABC$ , with point  $D$  as the feet of the altitude from vertex  $C$  (see Figure 3). Such point  $D$  yields three segments: altitude  $j = CD$ , and segments  $h = AD, i = DB$ , from  $D$  to the extremes of the side  $AB$ . Now, it might occur that the user vaguely remembers that it might be true that  $j^2 = h \cdot i$  (the Altitude Theorem). But neither the `Relation( $j^2, h \cdot i$ )` nor the `Prove( $j^2 = h \cdot i$ )` confirm such conjecture. As a consequence, the user asks GGD where to place point  $C$  so that the thesis  $j^2 = h \cdot i$  holds, through the `LocusEquation( $j^2 == h \cdot i, C$ )` command. The answer is displayed in the red, degree-four curve in Figure 3. But its interpretation, from the human point of view, in terms of triangles  $\triangle ABC$ , is not immediate; we also suggest the reader, as a reasonable, but not obvious task, to describe, for example, the triangles that happen when placing  $A$  in the output of `LocusEquation( $j^2 == h \cdot i, A$ )`. See [22] for some curious answers.



**Figure 3.** Finding with GeoGebra Discovery the locus of  $C$  such that the Altitude Theorem holds, i.e.,  $CD^2 = DB \cdot DA$ .

In this context some subtle (and not yet well—solved) issues arise:

- Does it mean that if the thesis  $j^2 == h \cdot i$  holds, vertex  $C$  must belong to the curve  $eq1$ ?
- Does it mean that placing vertex  $C$  in any/all points of the curve  $eq1$ , the thesis  $j^2 = h \cdot i$  holds?
- The standard mathematical interpretation of the statement behind a locus computation (i.e., “if a triangle  $\triangle ABC$  verifies such and such property, then the altitude from  $C$  is the geometric mean of line segments formed by altitude on the opposite side”) considers it holds for all triangles with generic vertices  $A = (a1, a2), B = (b1, b2)$  verifying the discovered requirements, not just to the triangles with vertices  $A = (0, 0), B = (1, 0)$ , as displayed in the figure. How to handle this generic vs specialized locus computation? Should we perform elimination considering coordinates of free points as parameters

(i.e., as elements of the coefficient field), or as variables that should be eliminated to yield the locus? And, what is the relation of specializing the result of the elimination by assigning some specific values to the free points (as happens when one displays a figure in GGD) compared with performing the elimination starting with specialized values of the free points?

- Locus computation deals with the positions of  $C$  in the figure displayed in the complex?/real? plane, so that the corresponding thesis  $j^2 == h \cdot i$  holds, for complex?/real? values of  $j, h, i$ , or for real values with some other restrictions  $j > 0, h > 0, i > 0$ ?

Let us briefly mention, referring the reader to the citations mentioned in the previous subsection for further details, that in GGD the `LocusEquation` is performed through an elimination algorithm that outputs the generators of the ideal of polynomials, exclusively in the variables of the locus point  $C$ , that belong to the ideal of equations involved in the construction of the figure and of the sought thesis. So, from the complex algebraic point of view, the output curve or algebraic variety of the `LocusEquation` command is the Zariski closure (see The Closure Theorem, p. 123 of [23]) of the set of positions for  $C$  where the imposed thesis holds.

As already mentioned, we must recall that such output does not take into consideration any non-degeneracy conditions, among other more subtle reasons, because these kind of conditions could involve their formulation of other points of the construction (e.g., vertex  $C$  of a triangle should not be aligned with the other two vertices  $A, B$ ), but no other variables than the ones of the selected locus point are regarded in the current, straightforward elimination algorithm that performs the `LocusEquation`. See [20] for a much more precise protocol, involving both the elimination over the coordinates of the chosen point and, for each element in this projection, a new elimination of the construction ideal, specialized in this element, over the remaining variables. This is achieved through a Gröbner Cover algorithm [24], which, due to its demanding performance and complexity, is not yet implemented in the GGD-CAS.

Even disregarding the omission of degeneracy conditions, other relevant problems arise from the strict dependency of the `LocusEquation` command on a simple elimination algorithm. Indeed, by the Closure Theorem, the output of this command is the closure of the projection of the complex positions of points  $C$  verifying the construction plus the thesis, so it could happen that the “real locus” (real points of the projection of real points verifying the thesis) is just a subset of such curve. The alternative for dealing with these issues, quite relevant regarding our main problem, is, on the one hand, to perform a “real projection” by means of a quantifier elimination, as follows (for the locus in Figure 3):

```
> with(RegularChains):
> with(SemiAlgebraicSetTools):
> p:=&E[j,h,i],((j^2-y^2=0) &and (h^2-(x^2)=0) &and (i^2-(x-1)^2=0)
&and (j^2-i*h=0) &and (y<>0) &and (j>0) &and (i>0) &and (h>0)):
> out := QuantifierElimination(p);
```

$$\text{out} := \&\text{or}((x < 0) \&\text{and} (x^2 - y^2 - x = 0), (0 < x - 1) \\ \&\text{and} (x^2 - y^2 - x = 0), \&\text{and}(0 < x, x < 1, x^2 + y^2 - x = 0))$$

yielding the displayed circle and the parabola. But, as already remarked, the quantifier elimination is much more costly than elimination, so an alternative, practical approach, is to start with the GGD command `LocusEquation` and analyze carefully its output from the real point of view, as we will explore in the next sections for which triangles the inequality in Problem 1 is an equality.



This analysis involves, in particular, avoiding the use of square roots for describing segment lengths or distances (to keep in the realm of polynomial ideals) and, thus, trying to formulate expressions that handle lengths or distances using only even powers. This is an issue that has been already approached in [25], where they define the so-called MEP (Minimal Euclidean Polynomials) of a given polynomial expression, the smallest multiple of this polynomial but involving the special variables only with even powers, and where an algorithm for computing the MEP is provided, addressing the interpretation of the MEP zero set over the reals, through the concepts of “degenerate” and “essential conditions” (see [25] for details and examples). An algorithm that we have mimicked in Section 4 and that can be described as a result of multiplying the given polynomial by itself, but with different signs in the special variables, although it can be also computed via an elimination algorithm, introducing new variables representing the squares of the special ones.

A final source of problems for interpreting the GGD output for the LocusEquation in our problem, has to do, as already mentioned, with the consistency of the obtained result for generic inputs. Indeed, it is quite straightforward to approach problems formulated for all triangles, such as Problem 1, by introducing free variables associated to the coordinates of the three vertices  $A, B, C$ , and to develop the locus computation in this general context. But, as the elimination complexity depends on the number of involved variables, GGD usually begins by internally assigning (although this fact does not affect the displayed plots), without loss of generality, some specific numerical values to some vertices, such as  $A, B$ , and then performs an elimination involving only the coordinates of  $C$ . In order to address and compare both approaches, in the next sections we have turned to Maple© Computer Algebra System (CAS) (version 2022, Maplesoft, at Waterloo, ON, Canada) to obtain a generic result, for arbitrary values of  $A, B$ , and then confirm that the specialization of the elimination of the generic result coincides with the elimination of the specialization. See Lemma 2.3 and 2.4 in [21] for some technical details and general results in the same direction.

### 3. The Equality Case: A Coordinate Approach

Now let us turn our attention to the key issues concerning these GGD-ART tools for the main problem that we would like to address in this paper, namely:

**Problem 2.** *For what triangles  $ABC$  the perimeter-area inequality becomes the equality*

$$(a + b + c)^2 = 6\sqrt{3}ch ? \quad (1)$$

The automated solution to Problem 2 can be split into two steps, using a kind of “abductive” reasoning (The general form of an abduction is: “a fact  $A$  is observed; if  $C$  was true, then  $A$  would certainly be true; so, it is reasonable to assume  $C$  is true.” [26], p. 372), which is quite popular in mathematical exploration [27]. In the first step, we look for families of triangles that satisfy the proposed equality; in the second step, we try to decide if these families are the only ones, which involves, strictly speaking, a locus computation. And, as in many cases where triangle inequalities are investigated, the first natural candidates for checking whether the corresponding equality holds are equilateral triangles. Now, let us assume our triangle  $\triangle ABC$  is equilateral and let us ask GGD about the relationship between its area and perimeter. Figure 4 shows that equality  $ch = \frac{\sqrt{3}}{18}(a + b + c)^2$  is the output after introducing the command `Relation(c*h, a+b+c)` in the algebraic view of GGD. This confirms our suspicion for equilateral triangles, as these satisfy the equality.

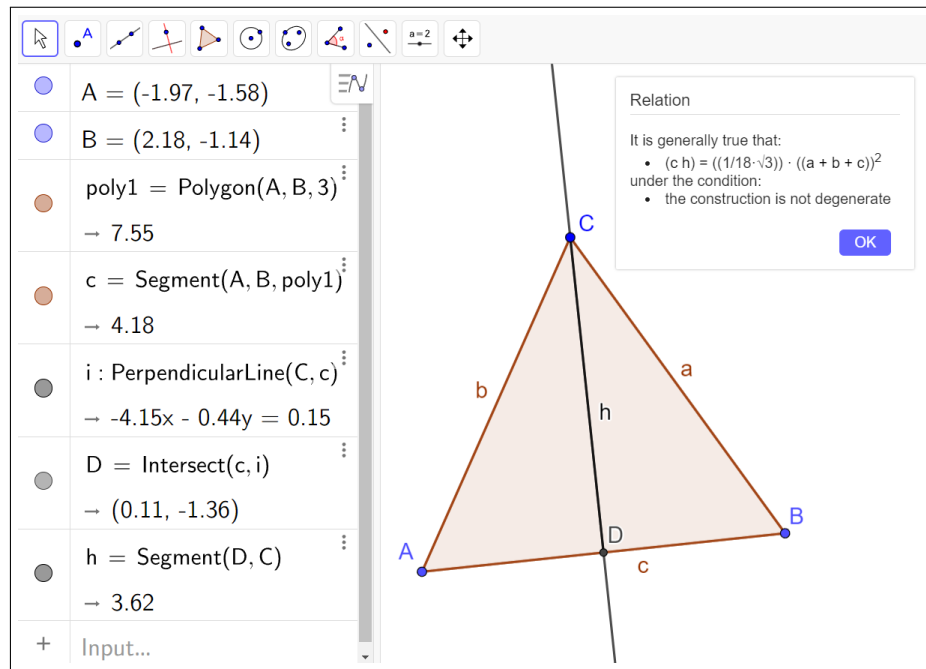


Figure 4. The case of an equilateral triangle.

Then we go for the second step of our approach, and we start by wondering whether equilateral triangles are the only ones answering positively Problem 2 (abductive thinking: if equality (1):  $ch = \frac{\sqrt{3}}{18}(a + b + c)^2$  is assumed, as we know that “equilateral triangles” verify such equality, it is reasonable to consider that if the equality holds on a triangle, the triangle must be equilateral). How could we rigorously prove such an assertion?

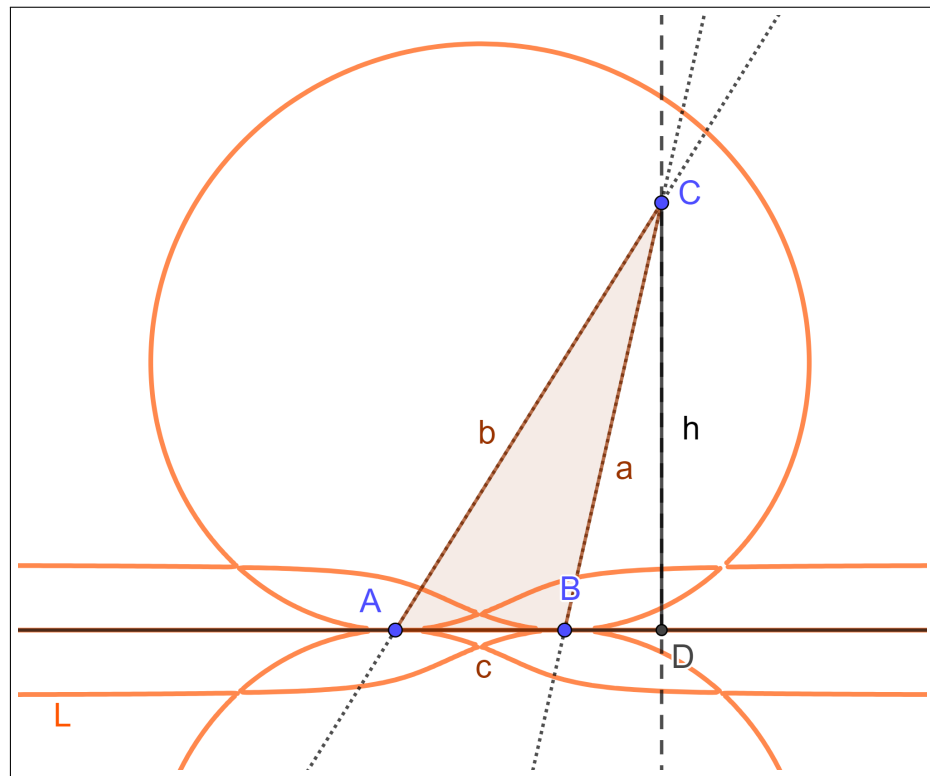
With the help of GGD and one of its ART commands we have a direct way to confront this question: Let us assume that vertices  $A, B$  are fixed. Then the command `LocusEquation(<Boolean Expression>, <Point>)` allows us to determine the geometric locus of all points satisfying the given Boolean Expression. In particular, we can set as condition the equality  $ch = \frac{\sqrt{3}}{18}(a + b + c)^2$  and as the distinguished point the vertex  $C$ , obtaining as output the locus of all possible positions of  $C$  satisfying the equality. In case only equilateral triangles comply with it we should only see two symmetric points with respect to the line  $AB$ , conforming the four possible equilateral triangles with  $AB$  as one of the sides. Now, because of technical reasons to avoid the presence of square roots in our expressions and in order to deal only with polynomials, as GGD-ART `LocusEquation` algorithms are complex algebraic geometry based, we rather introduce the following command, after squaring both sides of the equality:

$$\text{LocusEquation}(c^2 \cdot h^2 == (3/18^2) (a+b+c)^4, C)$$

and we examine the unexpected result, both visually (see Figure 5) and algebraically.

Indeed, for  $A = (0, 0)$  and  $B = (1, 0)$  we obtain the algebraic curve  $L$  with equation

$$\begin{aligned} I(x, y) := & 729x^8y^3 - 108x^8y - 2916x^7y^3 + 432x^7y + 2916x^6y^5 + 4050x^6y^3 - 648x^6y \\ & - 8748x^5y^5 - 1944x^5y^3 + 432x^5y + 4374x^4y^7 - 702x^4y^5 + 1107x^4y^3 \\ & - 108x^4y - 8748x^3y^7 + 15984x^3y^5 - 2376x^3y^3 + 2916x^2y^9 - 13986x^2y^7 \quad (2) \\ & - 27810x^2y^5 + 4266x^2y^3 - 2916xy^9 + 18360xy^7 + 18360xy^5 - 2916xy^3 \\ & + 729y^{11} - 9126y^9 + 8851y^7 - 9126y^5 + 729y^3 = 0. \end{aligned}$$



**Figure 5.** The locus obtained by GGD.

Where does this 11th-degree curve come from? Notice also that in the graphic view (see Figure 5) GGD does show none of the expected points we were considering, including those that correspond to the positions of  $C$  that make  $\triangle ABC$  equilateral. For those acquainted with computing the equations of geometric loci with a real algebraic geometry background, by using algebraic methods, the appearance of high-degree polynomials can be not such a big surprise, but it is always a challenge to give a geometric significance to the computational outputs obtained, especially when they differ substantially from the intuitively expected results, such as here, where we expected to obtain just a couple of points as the sought locus. See a simpler, but illustrative example, of the approach through a locus computation to the discovery of the Altitude Theorem, that we have developed in the previous section.

The following describes our approach to gaining some understanding of this situation, an approach that could establish the guidelines for a more systematic and general protocol. Firstly, we resort now to carry out personally the algebraic computations that are supposedly performed internally by GGD, but this time with the aid of Maple. Let us set coordinates  $A = (a_1, a_2)$ ,  $B = (b_1, b_2)$ ,  $C = (c_1, c_2)$ . Then we have the relations

$$a^2 = (c_1 - b_1)^2 + (c_2 - b_2)^2 \tag{3}$$

$$b^2 = (c_1 - a_1)^2 + (c_2 - a_2)^2 \tag{4}$$

$$c^2 = (b_1 - a_1)^2 + (b_2 - a_2)^2 \tag{5}$$

The point  $D$  where the altitude through  $C$  intersects line  $AB$  is given by the relations

$$(d_1 - a_1)(b_2 - a_2) - (d_2 - a_2)(b_1 - a_1) = 0 \tag{6}$$

$$(d_1 - c_1)(b_1 - a_1) + (d_2 - c_2)(b_2 - a_2) = 0 \tag{7}$$

and the height  $h$  satisfies

$$h^2 = (d_1 - c_1)^2 + (d_2 - c_2)^2 \tag{8}$$

Relations (3)–(8) lead to the polynomial ideal associated to the geometric construction

$$\begin{aligned} CS := & \langle a^2 - ((c_1 - b_1)^2 + (c_2 - b_2)^2), b^2 - ((c_1 - a_1)^2 + (c_2 - a_2)^2), \\ & c^2 - ((b_1 - a_1)^2 + (b_2 - a_2)^2), h^2 - ((d_1 - c_1)^2 + (d_2 - c_2)^2), \\ & (d_1 - a_1)(b_2 - a_2) - (d_2 - a_2)(b_1 - a_1), (d_1 - c_1)(b_1 - a_1) + (d_2 - c_2)(b_2 - a_2) \rangle, \end{aligned}$$

which is contained in  $\mathbb{C}[a_1, a_2, b_1, b_2, c_1, c_2, d_1, d_2, a, b, c, h]$ . Notice here that  $CS$  can not distinguish between positive/negative values of the lengths  $a, b, c$ , and  $h$ , as they appear always squared in the generators for this ideal. This is a key observation that explains the fact that the ideal  $CS$  presents certain symmetries, in the sense that if a polynomial  $p(a_1, a_2, b_1, b_2, c_1, c_2, d_1, d_2, a, b, c, h) \in CS$ , then the polynomials  $p(a_1, a_2, b_1, b_2, c_1, c_2, d_1, d_2, \pm a, \pm b, \pm c, \pm h)$  also belong to  $CS$ . Notice also that  $CS$  is symmetrical, and does not change if  $A, B$  are switched, or if  $C$  is replaced by its symmetrical regarding the line  $AB$ , since the perimeter or the height from vertex  $C$  to side  $AB$  do not change by permuting  $A$  by  $B$  or  $C$  by its opposite with respect to line  $AB$ . If we ask Maple for the prime components of its radical ideal that strictly contain the ideal  $CS$ , we obtain 16 prime components (see Table 1) that in fact contain redundancies, which reduce to just four of them, so that

$$\sqrt{CS} = CS_1 \cap CS_2 \cap CS_3 \cap CS_4.$$

Geometrically speaking, these components can be associated with the geometric configurations described below. In order to give these descriptions in such a way that the difference among the components  $CS_1 - CS_4$  becomes clearer, we assume implicitly that the distances  $a, b, c$ , and  $h$  appearing in the expressions can have arbitrary real values, including negative ones. This assumption will be used along the text whenever it helps distinguish, from a geometrical perspective, the algebraic expressions under study.

- (i)  $CS_1$  corresponds to constructions where the signs of  $ch$  and the signed area (we define here signed area of a triangle with vertices  $A = (a_1, a_2), B = (b_1, b_2), C = (c_1, c_2)$  by the expression  $[(b_1 - a_1)(c_2 - a_2) - (b_2 - a_2)(c_1 - a_1)]/2 = (a_1b_2 - a_1c_2 - a_2b_1 + a_2c_1 + b_1c_2 - b_2c_1)/2$ ) of triangle  $ABC$  coincide;
- (ii)  $CS_2$  corresponds to constructions where the signs of  $ch$  and the signed area of triangle  $ABC$  are opposite;
- (iii)  $CS_3$  corresponds to degenerate constructions where vertices  $A, B$  coincide and lengths  $a, b$  have the same sign and point  $D$  is arbitrary;
- (iv)  $CS_4$  corresponds to degenerate constructions where vertices  $A, B$  coincide and lengths  $a, b$  have opposite signs and point  $D$  is arbitrary.

Descriptions (i) and (ii) come from realizing, by inspecting the generators of  $CS_1$  and  $CS_2$ , that any construction corresponding to a point in  $V(CS_1)$  has a specular one in  $V(CS_2)$  with the sign of its height  $h$  reversed (and vice versa). In particular, the seventh generator of the corresponding ideals in Table 1 helps to provide the aforementioned characterization with respect to the signed areas. Descriptions (iii) and (iv) are straightforwardly deduced from the list of generators of  $CS_3$  and  $CS_4$  (see [28] for further evidence).

**Table 1.** Maple output for prime components of  $\sqrt{CS}$ .

Ideals	Set of Generators
CS <sub>1</sub>	$\langle -a_1h + b_1h - cc_2 + cd_2, a_2h - b_2h - cc_1 + cd_1, -a_1b_2 + a_1d_2 + a_2b_1 - a_2d_1 - b_1d_2 + b_2d_1, a^2 - b_1^2 + 2b_1c_1 - b_2^2 + 2b_2c_2 - c_1^2 - c_2^2, -a_1^2 + 2a_1c_1 - a_2^2 + 2a_2c_2 + b^2 - c_1^2 - c_2^2, -a_1^2 + 2a_1b_1 - a_2^2 + 2a_2b_2 - b_1^2 - b_2^2 + c^2, -a_1b_2 + a_1c_2 + a_2b_1 - a_2c_1 - b_1c_2 + b_2c_1 + ch, a_1c_1 - a_1d_1 + a_2c_2 - a_2d_2 - c_1d_1 - c_2d_2 + d_1^2 + d_2^2, a_1c_1 - a_1d_1 + a_2c_2 - a_2d_2 - b_1c_1 + b_1d_1 - b_2c_2 + b_2d_2, a_1c_1 - a_1d_1 + a_2c_2 - a_2d_2 - c_1^2 + c_1d_1 - c_2^2 + c_2d_2 + h^2, a_1^2h - 2a_1b_1h - a_1b_2c + a_1cc_2 + a_2^2h + a_2b_1c - 2a_2b_2h - a_2cc_1 + b_1^2h - b_1cc_2 + b_2^2h + b_2cc_1 \rangle$
CS <sub>2</sub>	$\langle a_1h - b_1h - cc_2 + cd_2, -a_2h + b_2h - cc_1 + cd_1, -a_1b_2 + a_1d_2 + a_2b_1 - a_2d_1 - b_1d_2 + b_2d_1, a^2 - b_1^2 + 2b_1c_1 - b_2^2 + 2b_2c_2 - c_1^2 - c_2^2, -a_1^2 + 2a_1c_1 - a_2^2 + 2a_2c_2 + b^2 - c_1^2 - c_2^2, -a_1^2 + 2a_1b_1 - a_2^2 + 2a_2b_2 - b_1^2 - b_2^2 + c^2, a_1b_2 - a_1c_2 - a_2b_1 + a_2c_1 + b_1c_2 - b_2c_1 + ch, a_1c_1 - a_1d_1 + a_2c_2 - a_2d_2 - c_1d_1 - c_2d_2 + d_1^2 + d_2^2, a_1c_1 - a_1d_1 + a_2c_2 - a_2d_2 - b_1c_1 + b_1d_1 - b_2c_2 + b_2d_2, a_1c_1 - a_1d_1 + a_2c_2 - a_2d_2 - c_1^2 + c_1d_1 - c_2^2 + c_2d_2 + h^2, -a_1^2h + 2a_1b_1h - a_1b_2c + a_1cc_2 - a_2^2h + a_2b_1c + 2a_2b_2h - a_2cc_1 - b_1^2h - b_1cc_2 - b_2^2h + b_2cc_1 \rangle$
CS <sub>3</sub>	$\langle c, b - a, b_1 - a_1, b_2 - a_2, a^2 - a_1^2 + 2a_1c_1 - a_2^2 + 2a_2c_2 - c_1^2 - c_2^2, -c_1^2 + 2c_1d_1 - c_2^2 + 2c_2d_2 - d_1^2 - d_2^2 + h^2 \rangle$
CS <sub>4</sub>	$\langle c, a + b, b_1 - a_1, b_2 - a_2, a^2 - a_1^2 + 2a_1c_1 - a_2^2 + 2a_2c_2 - c_1^2 - c_2^2, -c_1^2 + 2c_1d_1 - c_2^2 + 2c_2d_2 - d_1^2 - d_2^2 + h^2 \rangle$
CS <sub>5</sub>	$\langle c, a + b, b_1 - a_1, b_2 - a_2, d_1 - a_1, a^2 - a_1^2 + 2a_1c_1 - a_2^2 + 2a_2c_2 - c_1^2 - c_2^2, -a_1^2 + 2a_1c_1 - c_1^2 - c_2^2 + 2c_2d_2 - d_2^2 + h^2 \rangle$
CS <sub>6</sub>	$\langle c, a + b, b_1 - a_1, b_2 - a_2, d_1 - c_1, -c_2 + d_2 + h, a^2 - a_1^2 + 2a_1c_1 - a_2^2 + 2a_2c_2 - c_1^2 - c_2^2 \rangle$
CS <sub>7</sub>	$\langle c, a + b, b_1 - a_1, b_2 - a_2, d_1 - c_1, c_2 - d_2 + h, a^2 - a_1^2 + 2a_1c_1 - a_2^2 + 2a_2c_2 - c_1^2 - c_2^2 \rangle$
CS <sub>8</sub>	$\langle c, a + b, b_1 - a_1, b_2 - a_2, d_2 - a_2, a^2 - a_1^2 + 2a_1c_1 - a_2^2 + 2a_2c_2 - c_1^2 - c_2^2, -a_2^2 + 2a_2c_2 - c_1^2 + 2c_1d_1 - c_2^2 - d_1^2 + h^2 \rangle$
CS <sub>9</sub>	$\langle c, a + b, b_1 - a_1, b_2 - a_2, d_2 - c_2, -c_1 + d_1 + h, a^2 - a_1^2 + 2a_1c_1 - a_2^2 + 2a_2c_2 - c_1^2 - c_2^2 \rangle$
CS <sub>10</sub>	$\langle c, a + b, b_1 - a_1, b_2 - a_2, d_2 - c_2, c_1 - d_1 + h, a^2 - a_1^2 + 2a_1c_1 - a_2^2 + 2a_2c_2 - c_1^2 - c_2^2 \rangle$
CS <sub>11</sub>	$\langle c, b - a, b_1 - a_1, b_2 - a_2, d_1 - a_1, a^2 - a_1^2 + 2a_1c_1 - a_2^2 + 2a_2c_2 - c_1^2 - c_2^2, -a_1^2 + 2a_1c_1 - c_1^2 - c_2^2 + 2c_2d_2 - d_2^2 + h^2 \rangle$
CS <sub>12</sub>	$\langle c, b - a, b_1 - a_1, b_2 - a_2, d_1 - c_1, -c_2 + d_2 + h, a^2 - a_1^2 + 2a_1c_1 - a_2^2 + 2a_2c_2 - c_1^2 - c_2^2 \rangle$
CS <sub>13</sub>	$\langle c, b - a, b_1 - a_1, b_2 - a_2, d_1 - c_1, c_2 - d_2 + h, a^2 - a_1^2 + 2a_1c_1 - a_2^2 + 2a_2c_2 - c_1^2 - c_2^2 \rangle$
CS <sub>14</sub>	$\langle c, b - a, b_1 - a_1, b_2 - a_2, d_2 - a_2, a^2 - a_1^2 + 2a_1c_1 - a_2^2 + 2a_2c_2 - c_1^2 - c_2^2, -a_2^2 + 2a_2c_2 - c_1^2 + 2c_1d_1 - c_2^2 - d_1^2 + h^2 \rangle$
CS <sub>15</sub>	$\langle c, b - a, b_1 - a_1, b_2 - a_2, d_2 - c_2, -c_1 + d_1 + h, a^2 - a_1^2 + 2a_1c_1 - a_2^2 + 2a_2c_2 - c_1^2 - c_2^2 \rangle$
CS <sub>16</sub>	$\langle c, b - a, b_1 - a_1, b_2 - a_2, d_2 - c_2, c_1 - d_1 + h, a^2 - a_1^2 + 2a_1c_1 - a_2^2 + 2a_2c_2 - c_1^2 - c_2^2 \rangle$

We now proceed to consider the condition of the perimeter-area equality, which can be stated in the following form:

$$f := (18ch)^2 - 3(a + b + c)^4 = 0. \tag{9}$$

This equation is not symmetrical with respect to the variables  $a, b$ , and  $c$ , a fact that will be taken into consideration later. Imposing this condition (9) translates algebraically into the ideal  $CD = CS + \langle f \rangle$ . In order to see for what triangles  $ABC$  the equality (9) holds, we should begin by computing the elimination ideal of  $CD$  with respect to the variables involved in the coordinates of  $A, B$ , and  $C$ . The output is a very large 16th-degree polynomial with thousands of monomials, in the six variables  $\{a_1, a_2, b_1, b_2, c_1, c_2\}$ , so that its zeroes seem impossible to analyze in detail.

Thus, a reasonable alternative could be to proceed with the elimination of the ideals associated with the components of the variety  $CS$ , as they represent just smaller “parts” of the ideal  $CS$ . We noticed that, in general, it is not true (a trivial counterexample:  $(\langle x \rangle \cap \langle y \rangle) + \langle y - x^2 \rangle = \langle x * y, y - x^2 \rangle$ , while  $(\langle x \rangle + \langle y - x^2 \rangle) \cap (\langle y \rangle + \langle y - x^2 \rangle) = \langle y, x^2 \rangle$ ). that the ideal  $\sqrt{CS} + \langle f \rangle = (CS_1 \cap CS_2 \cap CS_3 \cap CS_4) + \langle f \rangle$  is equal to  $(CS_1 + \langle f \rangle) \cap (CS_2 + \langle f \rangle) \cap (CS_3 + \langle f \rangle) \cap (CS_4 + \langle f \rangle)$ .

Luckily, at an algebraic geometry level, considering the set of zeroes (denoted by  $V(I)$  of the corresponding ideal  $I$ ), we have  $V(\sqrt{CS} + \langle f \rangle) = V(\sqrt{CS}) \cap V(\langle f \rangle) = V(CS_1 \cap CS_2 \cap CS_3 \cap CS_4) \cap V(\langle f \rangle) = (V(CS_1) \cup V(CS_2) \cup V(CS_3) \cup V(CS_4)) \cap V(\langle f \rangle) = (V(CS_1) \cap V(\langle f \rangle)) \cup (V(CS_2) \cap V(\langle f \rangle)) \cup (V(CS_3) \cap V(\langle f \rangle)) \cup (V(CS_4) \cap V(\langle f \rangle)) = V(CS_1 + \langle f \rangle) \cup V(CS_2 + \langle f \rangle) \cup V(CS_3 + \langle f \rangle) \cup V(CS_4 + \langle f \rangle)$ . Moreover, the projection of a union of varieties is the union of the projections, and the Zariski closure of a finite union is the union of the corresponding Zariski closures.

Thus, since the variety  $V(Elim(I))$  of zeroes of the elimination of an ideal  $I$  (keeping just some special variables), is the Zariski closure of the projection of  $V(I)$  over the space of distinguished variables, we conclude that, from a geometric point of view, the locus variety defined by projecting  $V(CD) = V(CS + \langle f \rangle)$  can be studied just by considering the loci associated to  $CS_i + \langle f \rangle$ , by projecting the corresponding varieties.

Following this approach, let us label as  $CF_i := (CS_i + \langle f \rangle) \cap \mathbb{C}[a_1, a_2, b_1, b_2, c_1, c_2]$  for  $i = 1, \dots, 4$ , to obtain the conditions that the coordinates of the vertices  $A, B$ , and  $C$  must satisfy so that the perimeter-area equality holds. Certainly, this procedure allows us to obtain some results about the sought locus. We refer the reader to the Maple code in [28], where the following facts are verified:

- $CF_3 = \langle b_1 - a_1, b_2 - a_2, ((c_1 - a_1)^2 + (c_2 - a_2)^2)^2 \rangle$ . The triangles in this ideal reduce to the case  $A = B = C$ .
- $CF_4 = \langle b_1 - a_1, b_2 - a_2 \rangle$  holds, so that all configurations with  $A = B$  and  $a + b = 0$  satisfy the equality.
- The equality  $CF_1 = CF_2 = \langle g \rangle$  holds, where  $g \in \mathbb{C}[a_1, a_2, b_1, b_2, c_1, c_2]$  is a pretty large 16th degree polynomial.

In fact, the inclusions  $CF_1 \subset CF_4 \subset CF_3$  hold, so the projection of  $V(CD)$  is equal to  $V(CF_1) \cup V(CF_2) \cup V(CF_3) \cup V(CF_4) = V(CF_1)$  and the ideal relevant to us is  $CF_1$ . Unfortunately, in this case, we observe that splitting the computation of the projection between the components of  $CS$  does not simplify the final output, as computing directly the elimination ideal  $CD \cap \mathbb{C}[a_1, a_2, b_1, b_2, c_1, c_2]$ , we obtain again  $CF_1$ , so our “first find the components, then eliminate” approach is not always operative. Thus, in this case, we are forced to pay closer attention to the generator  $g$  of this last ideal. A first observation is that the polynomial  $g$  factorizes in the form  $g = g_1^2 g_2$ , where

$$g_1 = a_1 b_2 - a_1 c_2 - a_2 b_1 + a_2 c_1 + b_1 c_2 - b_2 c_1$$

$$g_2 = \text{A big polynomial of degree 12.}$$

Notice here that the polynomial  $g_1$  corresponds to the collinearity condition of points  $A, B$ , and  $C$ , a case of degeneracy (according to human intuition, not through any algorithmic definition) that is displayed in this particular case since we are computing the joint locus of the three vertices, whose coordinates are the only free variables of the whole construction. This means that if the equality (9) holds, then either the triangle collapses to a line, or it verifies  $g_2 = 0$ .

But even this partial and trivial interpretation of the output does not seem to help much concerning the initial question about which triangles verify equality (9). For example, after adding  $g_1 = 0$  to the generators of ideal  $CS$ , we observe (see computational details in [28]) that it is not true that, under the extended set of hypotheses, the equality (9) always holds. Indeed, a detailed analysis of this situation shows that the new ideal of hypotheses, i.e., the degenerate triangles that verify  $V(CS + \langle g_1 \rangle)$ , decompose into 29 prime ideals, of which only the first two are non-degenerate (i.e., with the same set of free variables than that of the ideal  $CS + \langle g_1 \rangle$ ) and the thesis (9) holds on the first one, but not on the second. The (algebraic) reason behind this fuzzy behavior is that in the first component,  $c = a$  as  $ABC$  are aligned, and  $a$  and  $b$  take opposite values, something not intuitive, since both are lengths, although algebraically possible, since our set of hypotheses described  $a$ ,  $b$ , and  $c$  as square roots, without fixing any sign for them. And, on the other hand, on the second component  $c = 0$ ,  $a = b$ , so the equality (9) is  $a = 0$ , but the length  $a$  of segment  $AB$  does not have to be always zero when  $ABC$  are aligned.

Because of this failure concerning  $g_1 = 0$ , and it seems impossible to interpret geometrically, the triangles whose vertices verify the equation  $g_2 = 0$ , the standard protocol turns, without loss of generality, to search for answers in a simpler context.

Thus, we now specialize the polynomial  $g$  setting  $A = (0, 0)$ ,  $B = (1, 0)$ , which, in this geometric context about perimeter and areas of triangles, keeps in all generality the analysis of the locus of triangles verifying the equality. The resulting polynomial

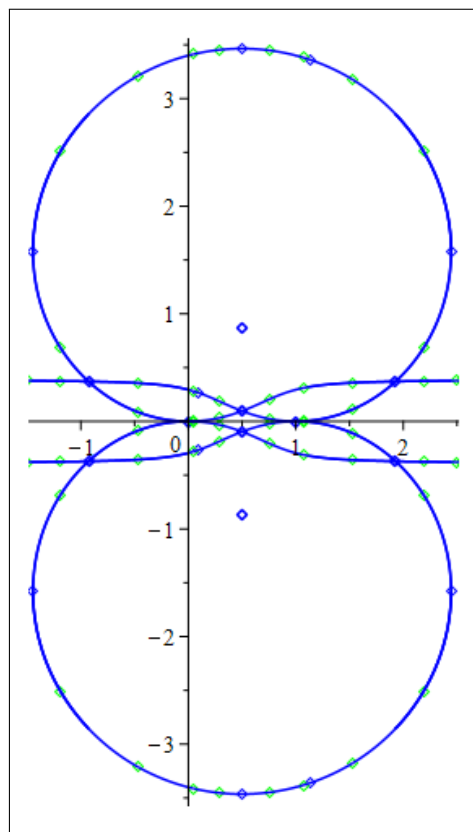
$$gg(c_1, c_2) := g(0, 0, 1, 0, c_1, c_2) = c_2 l(c_1, c_2),$$

coincides (here with coordinates  $\{c_1, c_2\}$  instead of  $\{x, y\}$ ) with the same GGD polynomial (see Formula (2)) we found by means of the `LocusEquation(, )` command for this particular case, an output that GGD obtains by performing the elimination on the specialized version of the input ideal  $(CS + \langle f \rangle)$ , as already pointed out in Section 2.2.

We can use `Maple` to obtain a graphical representation of the algebraic curve defined by  $gg(c_1, c_2) = 0$  to compare it with the `GeoGebra` output. Figure 6 shows how the locus includes two isolated points that correspond to  $(\frac{1}{2}, \pm \frac{\sqrt{3}}{2})$ , that is, the points that correspond to the equilateral triangles  $ABC$ .

Notice that, as expected, the curve in Figure 6 is symmetrical with respect to both the  $x$ -axis, i.e., the  $AB$  line, and with respect to the bisector line  $x = 1/2$  of the segment  $AB$ , since such symmetries for positioning vertex  $C$  do not affect the perimeter or area of the corresponding triangle.

Now we can proceed as previously, analyzing the statement that has as hypotheses the generators of  $CD$  and one of the two factors of the polynomial  $gg(c_1, c_2) = c_2 l(c_1, c_2) = 0$ , and as thesis the isoperimetric equality (9). We compute the dimension and free variables for each of the two hypotheses ideals  $CD + \langle c_2 \rangle$ ,  $CD + \langle l(c_1, c_2) \rangle$ , and then we observe, after computing in both cases a prime decomposition, and after checking that all their components are non-degenerate (in the context of the new ideals of hypotheses), that the corresponding statement is only true on some prime components of the considered ideal, and fails in some others. It is easy to observe that some “non-real” issues are involved in the components (e.g., negative lengths, sum of lengths equal zero, etc.). See the `Maple` worksheet available in [28] for details.



**Figure 6.** Maple image for the geometric locus when  $A = (0,0), B = (1,0)$ .

Therefore, despite all the developed work, both with and without specializing the input vertices  $A, B$ , the main question remains, that is, which other real triangles, if any, in addition to the equilaterals, satisfy  $g = 0$  (or  $gg = 0$ , after specialization)? We have seen that placing vertex  $C$  in such a curve does not imply that the isoperimetric equality is generally true, only true on some components and false in others. Which of these are real? It is hard to continue in this direction without becoming involved in the use of real quantifier elimination algorithms (that in this specialized context are able to solve in a reasonable time the locus problem, see [28]), something not available, for practical issues, in the current `LocusEquation` commands in GGD.

At least we have learned about the key role of the signs of the lengths, namely  $a, b$ , and  $c$ , for the interpretation of the results. Perhaps the reason behind the failure of the thesis (equality (9)) for the triangles with vertex  $C$  verifying  $gg = 0$ , is due to the need to extend the thesis, by considering not only the given Equation (9), but the product of variants of such equation for different choices of signs for  $a, b$ , and  $c$ , that is, the associated MEP polynomial [25]. Thus, we will proceed in the next section to approach the problem by changing our perspective, and working with the lengths of sides  $a, b, c$  as main variables, instead of the coordinates of vertices  $A, B$ , and  $C$ .

#### 4. The Equality Case: A Coordinate-Free Approach

In this section, we deal with the problem from a more intrinsic perspective, focusing on the variables  $a, b$ , and  $c$  that correspond to the sides of triangle  $\triangle ABC$  and ruling out the specific coordinates of the vertices of the triangle. This translates algebraically into dealing with the elimination ideals  $CG_i := (CS_i + \langle f \rangle) \cap \mathbb{C}[a, b, c], i = 1, \dots, 4$  (recall that  $f$ , defined in (9), represents the perimeter-area equality condition). In this case, we obtain with `Maple` the following results:



- $CG_3 = \langle c, b^4, -b + a \rangle$ . The triangles in this component reduce to those with  $a = b = c = 0$ .
- $CG_4 = \langle c, a + b \rangle$ . The triangles are those with  $c = 0$  and with  $a = -b$ .
- $CG_1 = CG_2 = \langle r(a, b, c) \rangle$ ,

where

$$r(a, b, c) = 7a^4 + a^3b + a^3c - 12a^2b^2 + 3a^2bc - 12a^2c^2 + ab^3 + 3ab^2c + 3abc^2 + ac^3 + 7b^4 + b^3c - 12b^2c^2 + bc^3 + 7c^4.$$

Maple can verify that  $\sqrt{CG_1} \subset \sqrt{CG_4} \subset \sqrt{CG_3}$ , and therefore, the relevant ideal here is  $CG_1$ . Let us now focus on the polynomial  $r(a, b, c)$ , which can be factorized and rewritten in the following form:

$$r(a, b, c) = (a + b + c)(-6(-a + b + c)(a - b + c)(a + b - c) + (a^3 + b^3 + c^3) + 3abc).$$

This last expression is interesting, because it can be related to the well known Heron’s formula for the area of a triangle in terms of its sides. Indeed,

$$r(a, b, c) = -108\sqrt{\frac{(a + b + c)(-a + b + c)(a - b + c)(a + b - c)}{16}} + \frac{(a + b + c)^4}{4},$$

and so

$$r(a, b, c) = -108(ch/2)^2 + \frac{(a + b + c)^4}{4}.$$

Thus, except for a constant,  $r(a, b, c)$  can be interpreted geometrically as an equivalent form of Formula (9),  $f(a, b, c, h) = (18ch)^2 - 3(a + b + c)^4 = 324c^2h^2 - 3(a + b + c)^4 = -12r(a, b, c)$ , expressed just in terms of the sides of  $\triangle ABC$ . This can be considered both negatively, since it means that the computed polynomial  $r$  is just another version of the starting locus constraint, or positively, since it is easier to interpret geometrically as it deals with only three variables, as we will show next. For example, the triangles that satisfy the equality given by  $f = 0$  must satisfy  $r(a, b, c) = 0$ . From this, two possibilities arise: either  $a + b + c = 0$ , or

$$r_1(a, b, c) := -6(-a + b + c)(a - b + c)(a + b - c) + (a^3 + b^3 + c^3) + 3abc = 0. \tag{10}$$

The first option leads to the unique possibility  $a = b = c = 0$  (considering all sides of the triangle non-negative). The second option drives us to find out which real triangles satisfy (10). Here, Maple gives us a clean response. By typing

```
> r1 := 7*c^3 - 6*c^2*b - 6*c^2*a - 6*c*b^2 + 15*c*b*a
      - 6*c*a^2 + 7*b^3 - 6*b^2*a - 6*b*a^2 + 7*a^3;
> solve([r1=0, a>=0, b>=0, c>=0], [a, b, c]);
```

we readily obtain

$$[[b = a, c = a, 0 \leq a]],$$

so that equilateral triangles are the only ones that satisfy (10).

What is the connection between the polynomial  $r(a, b, c)$ , of degree 4, and the polynomial  $g(a_1, a_2, b_1, b_2, c_1, c_2)$ , of degree 16? Let us set

$$s_a := \sqrt{(c_1 - b_1)^2 + (c_2 - b_2)^2} \tag{11}$$

$$s_b := \sqrt{(c_1 - a_1)^2 + (c_2 - a_2)^2} \tag{12}$$

$$s_c := \sqrt{(b_1 - a_1)^2 + (b_2 - a_2)^2}. \tag{13}$$

From the definition of our variables  $a_1, a_2, b_1, b_2, c_1, c_2, a, b, c$  in (3)–(5) we have the relations

$$a = \pm s_a, \quad b = \pm s_b, \quad c = \pm s_c$$

and we claim now that the following identity holds:

$$16g(a_1, a_2, b_1, b_2, c_1, c_2) = r(s_a, s_b, s_c)r(-s_a, s_b, s_c)r(s_a, -s_b, s_c)r(s_a, s_b, -s_c). \tag{14}$$

We can verify this statement with the following sequence of commands in Maple:

```
> r:=7a^4+a^3b+a^3c-12a^2b^2+3a^2bc-12a^2c^2+ab^3
+3ab^2c+3abc^2+ac^3;
> sa := sqrt((c1 - b1)^2 + (c2 - b2)^2);
> sb := sqrt((c1 - a1)^2 + (c2 - a2)^2);
> sc := sqrt((a1 - b1)^2 + (a2 - b2)^2);
> rs1 := subs([a = x, b = y, c = z], r);
> rs2 := subs([a = -x, b = y, c = z], r);
> rs3 := subs([a = x, b = -y, c = z], r);
> rs4 := subs([a = x, b = y, c = -z], r);
> rs:=expand(rs1*rs2*rs3*rs4);
> rg:=expand(subs([x=sa,y=sb,z=sc],rs));
> rg-16g;
```

The output of the last prompt is zero, confirming the validity of the identity.

We can interpret this equality by considering  $r(a, b, c)$  as an equivalent of the locus condition  $f$ , and then proceeding to compute—by multiplying different versions of  $r(a, b, c)$  changing signs for the involved variables—the smallest multiple of such condition involving only even powers of  $a, b$ , and  $c$ , i.e., the already mentioned in Section 2.2 MEP polynomial. An alternative method to compute  $rg$ , following the elimination algorithms described in [25], is performed in [28]. We refer the reader to [25] for details on how algebraic geometry provers implemented in software such as GGD work with Euclidean positive measures (distances, lengths) by computing the corresponding MEP to handle better the problems with signs, while keeping complex algebraic geometry computations, usually much simpler than the real geometry counterparts.

Indeed, the identity (14) helps to provide a full understanding of the locus we are dealing with, since it decomposes the locus into a union of four different semialgebraic subsets, each with a distinctive geometric meaning:

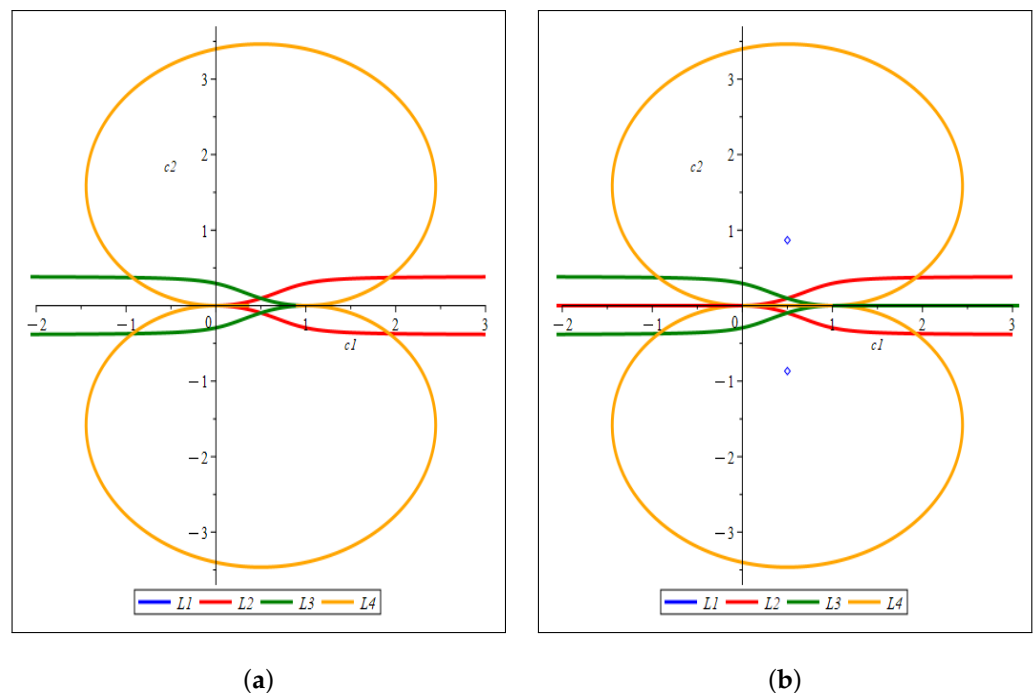
- The set  $L_1$  of points with  $r(s_a, s_b, s_c) = 0$  just consists of two points that correspond to the positions of  $C$  that form equilateral triangles with  $A, B$ .
- The set  $L_2$  of points with  $r(-s_a, s_b, s_c) = 0$  correspond to the triangles satisfying the perimeter-area equality, assuming  $a \leq 0$  and  $b, c \geq 0$ .
- The set  $L_3$  of points with  $r(s_a, -s_b, s_c) = 0$  correspond to the triangles satisfying the perimeter-area equality, assuming  $b \leq 0$  and  $a, c \geq 0$ .

- The set  $L_4$  of points with  $r(s_a, s_b, -s_c) = 0$  correspond to the triangles satisfying the perimeter-area equality, assuming  $c \leq 0$  and  $a, b \geq 0$ .

To visualize better the situation we can specialize these subsets in the case  $A = (0, 0)$  and  $B = (1, 0)$  by setting

$$\hat{s}_a := \sqrt{(c_1 - 1)^2 + c_2^2}, \quad \hat{s}_b := \sqrt{c_1^2 + c_2^2} \quad \text{and} \quad \hat{s}_c := 1.$$

Figure 7 shows the result in this particular case. It is worth point out here the difficulties that the software has in representing with precision these semialgebraic sets. In particular, nor  $L_1$  nor the segments  $\{c_1 \leq 0, c_2 = 0\}$  in  $L_2$ ,  $\{c_1 \geq 1, c_2 = 0\}$  in  $L_3$  and  $\{0 \leq c_1 \leq 1, c_2 = 0\}$  in  $L_4$  appear in the Maple graphical output when using the `implicitplot()` command.



**Figure 7.** The partition of  $L$  into  $L_1 \cup L_2 \cup L_3 \cup L_4$ . Maple version versus real version. (a) Maple output. (b) Corrected locus.

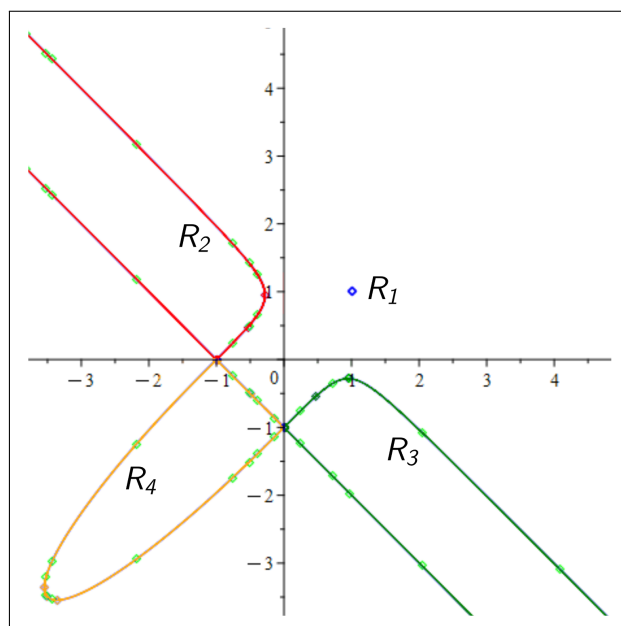
As a final verification, let us notice that, in this specialized situation in which  $A = (0, 0)$ ,  $B = (1, 0)$  so that  $c = 1$ , our polynomial  $r(a, b, c)$  becomes  $\hat{r}(a, b) := r(a, b, 1)$ , and we can plot the algebraic curve  $R$  with equation  $\hat{r}(a, b) = 0$  describing the set of triangles (with  $c = 1$ ) satisfying the perimeter-area equality in terms of their (signed) sides  $a$  and  $b$ , obtaining Figure 8. Then, let us set

$$\begin{aligned} R_1 &= R \cap \{a \geq 0, b \geq 0\}, & R_2 &= R \cap \{a \leq 0, b \geq 0\}, \\ R_3 &= R \cap \{a \geq 0, b \leq 0\}, & R_4 &= R \cap \{a \leq 0, b \leq 0\}. \end{aligned}$$

If we think now about the maps  $\mathbb{R}^2 \rightarrow \mathbb{R}^2$  given by

$$\begin{aligned} h_1 &: (c_1, c_2) \mapsto (\hat{s}_a, \hat{s}_b), & h_2 &: (c_1, c_2) \mapsto (-\hat{s}_a, \hat{s}_b), \\ h_3 &: (c_1, c_2) \mapsto (\hat{s}_a, -\hat{s}_b), & h_4 &: (c_1, c_2) \mapsto (-\hat{s}_a, -\hat{s}_b), \end{aligned}$$

we can verify that actually  $h_i^{-1}(R_i) = L_i$  (see [28] for more details on this verification).



**Figure 8.** Algebraic curve  $R = R_1 \cup R_2 \cup R_3 \cup R_4$  given by  $r(a, b, 1) = 0$ .

## 5. Conclusions

This study has explored the application of automated tools in investigating relevant geometric statements, such as the perimeter-area inequality in triangles. As a pertinent test for detecting different issues that could be required to improve GGD automated reasoning tools, we attempted to confirm with GGD and Maple, but always with the concurrence of the symbolic computation algorithms that are behind such GGD-ART, that the equality case holds exclusively for equilateral triangles. The approaches undertaken, both coordinate-based and coordinate-free, provided complementary insights into the problem, showcasing both the remaining difficulties and the versatility of computational methods in tackling geometric inequalities.

We consider that this work contributes to a deeper understanding of the perimeter-area inequality and highlights the evolving role of computational tools in mathematical research and education. In fact, although our work here is not directly related to mathematics education, it is clearly driven by the relevant role of education in the development of the software GeoGebra, a freely available, dynamic geometry, algebra, spreadsheets, graphing, statistics, calculus, etc. software, accessible over a variety of devices (laptops, smartphones, tablets. . .) and operating systems, on and offline. It is known to have more than 100 million users worldwide (data confirmed by GeoGebra’s CEO, personal communication, June 2024), mostly students or teachers, and it is, in many countries, the “de facto” standard in classrooms to visualize mathematics in a dynamic way. A software whose characteristics (type of users, multiplicity and simplicity of some supporting devices) have constrained our context for performing the isoperimetric–equality locus analysis, looking for approaches that could be more efficient and adequate to be implemented on future versions of GGD-ART than those directly requiring real quantifier elimination, as argued in the Introduction Section.

However, as we have illustrated in this paper, an innocent and natural geometry question such as Problem 1, can pose a series of challenging issues to the currently quite performing GGD-ART, a collection of tools capable to successfully deal, as can be perceived in the previously mentioned GGD-ART benchmark, with geometric statements with a much more complex formulation. Challenging questions that have required human intervention, with the help of computer software, to be visualized and fully understood. Of course, we

consider that these human-driven techniques we have exhibited in this paper and that we have created along our development of GGD-ART (see [17,19,25] for some basic references): computing a generic locus through elimination; checking truth, or truth on parts, by prime decomposition; performing a specialization of the problem and repeating the same steps over this less complex formulation; addressing the locus interpretation through the minimum euclidean polynomial of the sides-only version of the isoperimetric-equality, etc., could be applied to other geometric situations where a locus is involved.

However, the study also reveals certain existing limitations of automated tools for dealing with geometric loci involving real algebraic geometry. In particular, the deficiencies observed in the graphic outputs of mathematical software can be attributed to the difficulty of devising algorithms to display faithfully real algebraic plane curves (see [29]). Thus, we have highlighted and applied using this example some commutative algebra issues that, although theoretically developed, are still not fully implemented in GGD, such as the handling of specializations, the use of component-wise loci protocols, and the advantages of performing some sign-sensitive avoidance computations by translating the locus constraint to one with even powers when dealing with distances. Other issues, such as interpreting high-degree polynomial loci and ensuring precise graphical representations, underscore the need for continued refinement of these systems. In addition, human insight remains indispensable for interpreting results, defining problems, and contextualizing findings within larger mathematical theories.

But we are not sure if, when the algebraic complexity of the statement grows, even sophisticated software such as Maple, which has been the key here to apply the mentioned techniques and to interpret the GGD-ART output of the perimeter-area equality, could become stuck in the involved computations. For a simple, yet seemingly challenging example, we dare to propose the reader to consider extending our Problem 2, asking the following parametric variant:

**Problem 3.** *Let  $k$  be a real number with  $k \geq 12\sqrt{3}$ . For what triangles  $ABC$  the relation  $(\text{Perimeter}(ABC))^2 = k \cdot \text{Area}(ABC)$ ?*

Future studies could extend this approach to other geometric inequalities or problems involving higher-dimensional figures. Moreover, the integration of advanced algorithms for symbolic and numerical computation in the automated reasoning tools, such as the ones we have exhibited when dealing with this example, could further enhance their efficiency and scope, paving the way for more widespread adoption in classrooms and research environments.

In general terms, we consider the achieved results to emphasize the growing importance of dynamic geometry software integrated with computer algebra systems. Such tools not only streamline complex algebraic manipulations but also allow users to visualize and interact with geometric configurations, thereby enhancing intuition and understanding. For example, the ability of GGD to compute loci and handle inequalities demonstrates its potential as an educational and research tool, enabling users to explore mathematical relationships that might otherwise remain inaccessible.

Ultimately, this investigation underscores the synergistic relationship between human intuition and computational power. As tools like GGD become increasingly sophisticated, they will continue to complement traditional approaches, fostering innovation and discovery in both theoretical and applied mathematics.

**Author Contributions:** The three authors contributed equally to this work. All authors have read and agreed to the published version of the manuscript.

**Funding:** First and third authors are partially supported by the project “Inteligencia aumentada en educación matemática mediante modelización, razonamiento automático e inteligencia artificial (IAXEM-CM)” (PHS-2024/PH-HUM-383, granted by the Comunidad de Madrid-Consejería de Educación, Ciencia y Universidades-Programas de Actividades de I+D entre Grupos de Investigación de la Comunidad de Madrid, Convocatoria Procesos Humanos y Sociales).

**Data Availability Statement:** The computations performed with Maple computer algebra system for the development of this work are available at <https://github.com/carlosueno/GGD-and-Maple-files-A-computational-approach-to-the-perimeter-area-inequality-in-a-triangle> (accessed on 1 January 2025).

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

## References

1. GeoGebra. Available online: <https://www.geogebra.org/> (accessed on 2 October 2024).
2. GeoGebra Discovery. Available online: <https://github.com/kovzol/geogebra-discovery> (accessed on 2 October 2024).
3. Tarski. Available online: <https://www.usna.edu/Users/cs/wcbrown/tarski/index.html> (accessed on 18 November 2024).
4. Bottema, O.; Djordjević, R.Ž.; Janič, R.R.; Mitrinović, D.S.; Vasic, P.M. *Geometric Inequalities*; Wolters-Noordhoff Publishing: Groningen, The Netherlands, 1969.
5. Hadwiger, H. Ergänzungen zu zwei Defizitformeln des Dreiecks. In *Jahresbericht der Deutschen Mathematiker Vereinigung*; Sperner, E., Ed.; B.G. Teubner: Leipzig/Berlin, Germany, 1939; pp. 35–39.
6. Santaló, L.A. Algunas desigualdades entre los elementos de un triángulo. *Math. Notae* **1943**, *3*, 65–73.
7. Chang, G.; Sederberg, T.W. *Over and over Again*; Anneli Lax New Mathematical Library; Mathematical Association of America: Washington, DC, USA, 1997; Volume 39, ISBN 978-0-88385-953-7.
8. Heath, T. *A History of Greek Mathematics*; Oxford University Press: Oxford, UK, 1921; pp. 206–213.
9. Yang, L.; Hou, X.; Xia, B. Automated Discovering and Proving for Geometric Inequalities. In *Automated Deduction in Geometry. ADG 1998*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 1999; Volume 1669. [\[CrossRef\]](#)
10. Chou, S.-C.; Gao, X.-S.; Zhang, J.-Z. *Machine Proofs in Geometry: Automated Production of Readable Proofs for Geometry Theorems*; Series on Applied Mathematics; World Scientific: Singapore, 1994; Volume 6.
11. Brown, C.; Kovács, Z.; Recio, T.; Vajda, R.; Vélez, M.P. Is Computer Algebra Ready for Conjecturing and Proving Geometric Inequalities in the Classroom? *Math. Comput. Sci.* **2022**, *16*, 31. [\[CrossRef\]](#)
12. GeoGebra Discovery Releases. Available online: <https://github.com/kovzol/geogebra/releases> (accessed on 12 November 2024).
13. GeoGebra Discovery Online. Available online: <https://www.autgeo.online/geogebra-discovery/> (accessed on 12 November 2024).
14. Quaresma, P. Evolution of Automated Deduction and Dynamic Constructions in Geometry. In *Mathematics Education in the Age of Artificial Intelligence, Mathematics Education in the Digital Era 17*; Richard, P.R.F., Vélez, M.P., Van Vaerenbergh, S., Eds.; Springer: Cham, Switzerland, 2022; pp. 3–22. [\[CrossRef\]](#)
15. Kovács, Z.; Parisse, B. Giac and GeoGebra—Improved Gröbner basis computations. In *Computer Algebra and Polynomials*; Lecture Notes in Computer Science; Gutierrez, J., Schicho, J., Weimann, M., Eds.; Springer: Cham, Switzerland, 2015; Volume 8942, pp. 126–138. [\[CrossRef\]](#)
16. Kovács, Z.; Brown, C.; Recio, T.; Vajda, R. Computing with Tarski formulas and semi-algebraic sets in a web browser. *J. Symb. Comput.* **2024**, *120*, 102235. [\[CrossRef\]](#)
17. Recio, T.; Vélez, M.P. Automatic discovery of theorems in elementary geometry. *J. Autom. Reason.* **1999**, *23*, 63–82. [\[CrossRef\]](#)
18. Chou, S.C. *Mechanical Geometry Theorem Proving*; D. Reidel Publishing Co.: Dordrecht, The Netherlands, 1987.
19. Kovács, Z.; Recio, T.; Vélez, M.P. Automated reasoning tools with GeoGebra: What are they? What are they good for? In *Mathematics Education in the Age of Artificial Intelligence*; Richard, P.R., Vélez, M.P., van Vaerenbergh, S., Eds.; Series: Mathematics Education in the Digital Era; Springer: Cham, Switzerland, 2022; pp. 23–44. [\[CrossRef\]](#)
20. Abánades, M.; Botana, F.; Montes, A.; Recio, T. An algebraic taxonomy for locus computation in dynamic geometry. *Comput. Aided Des.* **2014**, *56*, 22–33. [\[CrossRef\]](#)
21. Dana-Picard, T.; Recio, T. From Thales theorem to octic curves. *Maple Trans.* **2024**, *4*, 18022. [\[CrossRef\]](#)
22. Etayo-Gordejuela, F.; de Lucas-Sanz, N.; Recio, T.; Vélez, M.P. Inventando teoremas con GeoGebra: Un nuevo Teorema de la Altura. *Boletín Soc. Puig Adam* **2021**, *111*, 8–27. (In Spanish)
23. Cox, D.A.; Little, J.; O’Shea, D. *Ideals, Varieties and Algorithms*, 1st ed.; Undergraduate Texts in Mathematics; Springer: New York, NY, USA, 1992. [\[CrossRef\]](#)

24. Montes, A. Automatic Deduction of Geometric Theorems. In *The Gröbner Cover; Algorithms and Computation in Mathematics*; Springer: Cham, Switzerland, 2018; Volume 27. [[CrossRef](#)]
25. Kovács, Z.; Recio, T.; Sólyom-Gecse, C. Rewriting input expressions in complex algebraic geometry provers. *Ann. Math. Artif. Intell.* **2019**, *85*, 73–87. [[CrossRef](#)]
26. Peirce, C.S. *Collected Papers II, Elements of Logic*; University Press: Harvard, UK, 1960.
27. Arzarello, F.; Andriano, V.; Olivero, F.; Robutti, O. Abduction and Conjecturing in Mathematics. *Philosophica* **1998**, *61*, 77–94. [[CrossRef](#)]
28. Recio, T.; Ueno, C.; Vélez, M.P. GGD and Maple Files: A Computational Approach to the Perimeter-Area Inequality in a Triangle. Available online: <https://github.com/carlosueno/GGD-and-Maple-files-A-computational-approach-to-the-perimeter--area-inequality-in-a-triangle> (accessed on 1 January 2025).
29. Labs, O. A List of Challenges for Real Algebraic Plane Curve Visualization Software. In *Nonlinear Computational Geometry*; Emiris, I., Sottile, F., Theobald, T., Eds.; The IMA Volumes in Mathematics and its Applications; Springer: New York, NY, USA, 2009; Volume 151, pp. 137–144. [[CrossRef](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.