

Some issues on the automatic computation of plane envelopes in interactive environments

Francisco Botana

*Departamento de Matemática Aplicada I, Universidad de Vigo, Campus A Xunqueira,
36005 Pontevedra, Spain*

Tomas Recio

*Departamento de Matemática Aplicada, Estadística y Computación, Universidad de
Cantabria, Avda. Los Castros, 39071 Santander, Spain*

Abstract

This paper addresses some concerns, and describes some proposals, on the elusive concept of envelope of an algebraic family of varieties, and on its automatic computation.

We describe how to use the recently developed Gröbner Cover algorithm to study envelopes of families of algebraic curves, and we give a protocol towards its implementation in dynamic geometry environments. The proposal is illustrated through some examples. A beta version of GeoGebra is used to highlight new envelope abilities in interactive environments, and limitations of our approach are discussed, since the computations are performed in an algebraically closed field.

Keywords: envelope, Dynamic Geometry, automatic computation, GröbnerCover algorithm

1. Introduction

This paper addresses some concerns, and describes some proposals, on the elusive concept of envelope of an algebraic family of varieties, and on its automatic computation.

Email addresses: `fbotana@uvigo.es` (Francisco Botana), `tomas.recio@unican.es` (Tomas Recio)

Preprint submitted to Elsevier

May 24, 2014

We will deal with these issues by restricting our framework to

- families of algebraic plane curves;
- in the context of dynamic geometry software (DGS).

Yet, as we will show below (see examples in this Section), even in this restricted setting we will need to reflect about the very basic concept of envelope, in order to be able to propose some sound algorithmic protocols for its computation.

1.1. *The manifold concept of envelope*

It is clear that the idea of envelope of a family of curves is an elementary differential geometry notion. Despite this elementary character, it is a definition that does not seem to generate a unanimous consensus on its basic terms. For instance, let us consider a family of curves $\{C_\alpha : F(x, y, \alpha) = 0\}$, where for each value α of the parameter we assume $F(x, y, \alpha) = 0$ to be the implicit polynomial equation of the curve C_α . In Wikipedia¹, the envelope of this family of curves is introduced as a certain curve which is tangent to each one of the C_α 's in the family. But, immediately after, four definitions are proposed and discussed concerning the very same concept:

1. the envelope as the set of all points (x, y) such that there is an α verifying $\{F(x, y, \alpha) = 0, \frac{\partial F(x, y, \alpha)}{\partial \alpha} = 0\}$, i.e., as the projection over the (x, y) -plane of the points, in the (x, y, α) -3 dimensional space, belonging to the surface $F(x, y, \alpha) = 0$ and having tangent plane parallel to the α -axis (or being singular points and, thus, not having tangent plane, properly speaking),
2. the envelope as the set of limit points of intersections of nearby curves C_α ,
3. (as previously introduced) the envelope as a curve tangent to all the given curves,
4. the envelope as the curve that bounds the planar region described by the points belonging to the curves in the family.

¹[https://en.wikipedia.org/wiki/Envelope_\(mathematics\)](https://en.wikipedia.org/wiki/Envelope_(mathematics))

Finally, the Wikipedia points out that these four definitions are not, in general, coincident; and that they yield to different envelope sets, E_i , for each definition $i = 1, 2, 3, 4$. But this situation is, by no means, a problem with the Wikipedia only. In a previous paper [4] we have reviewed different reputed sources, some classical and some very modern, all of them expressing the existence of a plurality of approaches to the concept of envelope, as well as describing the many subtle and difficult aspects involved in handling this apparently elementary notion. We also have summarily described in the same paper [4] the limitations for computing a specific envelope with general purpose and well-known DGS, such as Sketchpad, Cabri, Cinderella or GeoGebra. See Section 2 for a discussion of the state of the art. A rough and immediate conclusion from that survey is the need for improvement in different directions:

- extending the computation of envelopes to families of curves of much more general type (i.e. not limited to lines or circles, etc.);
- identifying the resulting envelope curve as an element of DGS (so that the system is able to do further operations with this curve) and not as a mere graphical display;
- improving the reliability and accuracy of the envelope computation (currently merely conjectural in most DGS, often erroneously including some extra components or omitting some true components with respect to the correct envelope).

1.2. On some difficulties when computing envelopes

As mentioned above, some of the main difficulties dealing with envelopes do not disappear even if we restrict ourselves to working in the frame of computing with families of planar curves build up by manipulating with DGS. Thus, in order to justify the use in this context of very powerful tools from computational algebraic geometry (such as Gröbner Cover, see Section 3), let us start by discussing the following simple examples, which can provide the reader an idea of the involved subtleties.

Example 1. The first example is about computing the envelope of a very simple family of lines in the (x, y) -plane. Here we consider the family of all lines parallel to the x -axis, say, described by $y = t$, for a single parameter t .

Then its envelope is –according to one of the classical definitions, but choosing any of them will yield here the same output– the result of eliminating t from $y = t, \partial(y - t)/\partial t$. Since $\partial(y - t)/\partial t$ is -1 , the system has no solution. There is no envelope.

Now let us make the following consideration. It is easy to imagine that someone working with a DGS could be making a construction, attempting to build up a certain family of curves, but –unfortunately– it could happen that the performed construction it is not necessarily the simplest possible one yielding the same family.

Bearing this remark in mind, let us discuss here an absurd, toy example, about such situation. For this purpose let us consider, again, a family of lines in the plane, each one going through a point of the curve $t_1^3 + t_2^2 = 1$ and so that they are all parallel to the x -axis. Each point (t_1, t_2) parametrizes the family of lines $y = t_2$. Geometrically, the fact that t_2 is the second coordinate of an arbitrary point in the curve does not play any special role, since the family of lines is identical to that previously introduced, described as $y = t$ for a single parameter t . Thus, it could happen that a naive dynamic geometry user could end up considering that computing the envelope for the constructed family $y = t_2$, where $t_1^3 + t_2^2 - 1 = 0$, will output the same envelope as for the family $y = t$, for a single parameter t .

But, what is the envelope of $y = t_2, t_1^3 + t_2^2 - 1 = 0$? These two equations define an algebraic variety V (a surface, indeed) in the (x, y, t_1, t_2) -affine space and we would like –following the classical definition number 4, as in 1.1– to compute the boundary of its projection onto the (x, y) -plane. That is, the points in this plane that lift up to points of V such that either they are singular or they have tangent plane parallel to the (t_1, t_2) -plane. Namely, we should consider the projection of the set of points in V where the Jacobian of $y = t_2, t_1^3 + t_2^2 - 1 = 0$, with respect to (t_1, t_2) , vanishes, i.e. where $3t_1^2 = 0$. The result of the elimination procedure is the couple of lines $y = 1, y = -1$, namely, the envelope curve for the same family of lines presented in this other way!

Obviously, this result can also be obtained and explained in a simpler way by considering $y = t_2, t_1^3 + t_2^2 - 1 = 0$ as $y = \sqrt{1 - t^3}$ for a single, independent parameter and then applying the classical elimination of t from the couple of equations $y = \sqrt{1 - t^3}, \partial(\sqrt{1 - t^3}/\partial t)$, yielding, again, the pair of lines $y = \pm 1$.

The lesson here (more or less well-known, see [6, p. 104] for the same kind of example, playing with a family of lines parallel to an axis, defined

as $y = f(t)$ for a certain function $f(t)$ different from t or from $\sqrt{1-t^3}$ is that computing the envelope of a family of curves strongly depends on the “algebraic” presentation of the family; the envelope is not about a geometric object, but about a dynamic one, related to the function that describes the family of lines and not only related to the geometric image of such function.

The lesson learnt, for dynamic geometry users, is that in the context of DGS, building up the “algebraic” presentation of the curve family is closely related to the construction protocol. In other words, one does not really compute the envelope of a tracer (an element of the family) determined by a point (as it would be the case in Cinderella or Cabri), but rather the envelope of the *constructed tracer* parametrized by the *constructed point*.

The following example exploits further the situation previously presented.

Example 2. Let us assume that we build up a family of lines parallel to the x -axis by a *very* redundant construction, namely,

- collecting all lines parallel to the x -axis and parametrized by points in the unit circle, i.e. $y = t_2, t_1^2 + t_2^2 - 1 = 0$;
- *moreover*, we choose another point (t_3, t_4) in each line (so $t_4 = t_2$) and we construct –redundantly– the line going through (t_1, t_2) and (t_3, t_4) , that is $(t_2 - t_4)(x - t_1) - (t_1 - t_3)(y - t_2) = 0$.

The result, given through the equation system,

$$y = t_2, t_1^2 + t_2^2 - 1 = 0, t_4 = t_2, (t_2 - t_4)(x - t_1) - (t_1 - t_3)(y - t_2) = 0,$$

describes a 3-dimensional variety in the $(x, y, t_1, t_2, t_3, t_4)$ -six dimensional space. This is, somehow, surprising: four equations in six variables should yield –if the equations were independent– a dimension 2 variety, but here it is not, because the last equation is a combination of $y = t_2, t_4 = t_2$, and can be removed and thus the dimension of the resulting variety should be $6-3=3$.

As a complementary argument in this direction we could consider that only two –say, t_2, t_3 – of the four parameters (t_1, t_2, t_3, t_4) are independent and that for each value of the pair (t_2, t_3) we are building an one dimensional object (a line). Thus, we could conclude the whole construction has dimension 3, since we have 2 (the degrees of freedom of the parameter space)+1 (the dimension of the geometric object built up for each parameter value) = 3. Obviously, this specific way of reasoning is hard to generalize or to implement as an algorithm over DGS.

The observation that one of the equations –say $E(x, y, t_1, t_2, t_3, t_4) = 0$ – is a combination of the other ones is the key point we would like to analyze here. In fact,

- noticing such combination means that we have, in some sense, discovered a statement that automatically holds in our construction, while we were thinking of $E(x, y, t_1, t_2, t_3, t_4) = 0$ as imposing a new, extra constraint;
- it is not possible, currently, to expect DGS to detect such facts automatically in most cases, or to require such detection as a previous step for envelope computation. Thus, we should not consider to have such information ready when aiming to compute an envelope;
- yet, it is crucial to clarify the global dimension of the involved variety of variables and parameters in order to obtain a correct envelope for the constructed family.

In fact, if we (wrongly) consider the four given equations, the vanishing of the Jacobian (with respect to the four parameters) is $-2t_1(y - t_2) = 0$, and the elimination of the system described by the equations and the Jacobian is 0, that is, the envelope in the (x, y) -plane turns to be the whole plane.

On the other hand, if we consider just the three independent equations and we impose the condition that the Jacobian of the three equations with respect to the four parameters is of rank 2, i.e., the new condition $-2t_1 = 0$, the elimination of the parameters in the system

$$y = t_2, t_1^2 + t_2^2 - 1 = 0, t_4 = t_2, -2t_1 = 0,$$

yields the equation $y^2 = 1$, that is, the previously obtained couple of lines, as the true envelope of the given family.

As a final remark about this example, notice that, even after deleting the depending equation, our family of lines is described by four parameters t_1, t_2, t_3, t_4 linked by two equations

$$t_1^2 + t_2^2 - 1 = 0, t_4 = t_2,$$

so that the whole family is defined by the 3-dimensional variety

$$y = t_2, t_1^2 + t_2^2 - 1 = 0, t_4 = t_2.$$

It is quite clear that we can take, say, t_2, t_3, x as independent variables ruling this 3-dimensional variety. On the other hand, the role of t_3 is null, it does not appear in any of the considered constraints. Thus, we could avoid including such parameter in our computations and, if considered in this way, the family of lines

$$y = t_2, t_1^2 + t_2^2 - 1 = 0, t_4 = t_2$$

over the (x, y, t_1, t_2, t_4) -space would have dimension 2 (one degree of freedom for the parameters t_1, t_2, t_4 , represented by the parameter t_2 , plus one degree of freedom for each line $y = t_2$ for each value of t_2), which is closer to our intuition than the dimension 3 obtained above. Moreover, eliminating this fake parameter t_3 would lead to easier envelope computations, since then we would have just to compute the vanishing of the Jacobian of three equations with respect to the three remaining parameters, and then to some elimination, yielding to the same result $y^2 = 1$. Of course, the problem here is how to automatize the discovery of such “fake” parameters. . .

As a conclusion, in the previous example the message was “computing envelopes in DGS depends on the construction protocol and not just on the geometric object”. Here, in this last example, the message could be summarized as stating that “it is crucial to control the algebraic data related to the construction protocol in order to correctly fix the settings for the envelope computation”.

The precedent example, one that includes some unnecessary parameters or equations, could be thought as artificial (in the dynamic geometry context). But we think it is not so.

Example 3. Consider the computation of the envelope of a family of parallel rays reflecting on the unit circle C , a caustic. At some point one needs to construct the symmetrical of a line L with respect to the axis S described by the center of C (the origin) and the point of intersection $C \cap L$.

Now, if such construction is not included as a macro in the DGS we could be using, one would need to make an auxiliary construction for the symmetrical of a line L with respect to some axis S . One possibility is to take a point $P \in L$ and then the perpendicular line, T , to S through P . Now compute the intersection M of the lines T and S . Then the symmetrical of P will be point Q such that $Q = M + \overrightarrow{PM}$. Finally one constructs, as the symmetrical of L , the line LL going through $L \cap S$ and Q .

Next, we remark that this procedure for obtaining the symmetrical of some given line L with respect to some given axis S should yield a unique result. But it is clear that the above method includes one irrelevant degree of freedom (namely, from choosing point $P \in L$). Obviously, the fact that dragging point P along the line L yields, in the above construction, always the same line LL , is an interesting (and well known) geometric property. But it can illustrate our claim that, in general, a user does not have to be aware of such facts when using some construction for envelope computations.

1.3. Our goal

A general conclusion, from the precedent section, is the presence of many, unexpected difficulties for envelope computation when attempting to implement such features in DGS. In fact, users in such context are prone to identify what they visualize to what they have actually computed. . . Here the challenge is to be able, through the inclusion of automatic reasoning features, to open a dialog window with the user, warning him/her about the different subtleties of his/her performed construction, if an envelope command is used in this setting. Although we have been working for some time in this direction ([7, 13, 14]), there remain many pending issues for the full performing inclusion of reasoning features in DGS.

For all these reasons, in this paper we focus on a very specific part of the envelope computation problem. Namely, we will assume the algebraic setting correctly represents what the user wants to compute, and we will concentrate on another difficult problem: how to proceed with the envelope computation as the projection of a given algebraic variety in the space of variables and parameters.

2. Envelope computation in current dynamic geometry environments

There is a wide consensus on requiring abilities related to the determination of geometric loci for an interactive geometric environment to be considered as a standard one. See, for instance, [8]. For example, the locus item is a distinguished one in the Wikipedia article² listing 36 computer programs categorized as plane DGS. Nevertheless, envelopes are not mentioned as a

²http://en.wikipedia.org/wiki/List_of_interactive_geometry_software

relevant topic in such article, and most DGS lack a specific command to deal with.

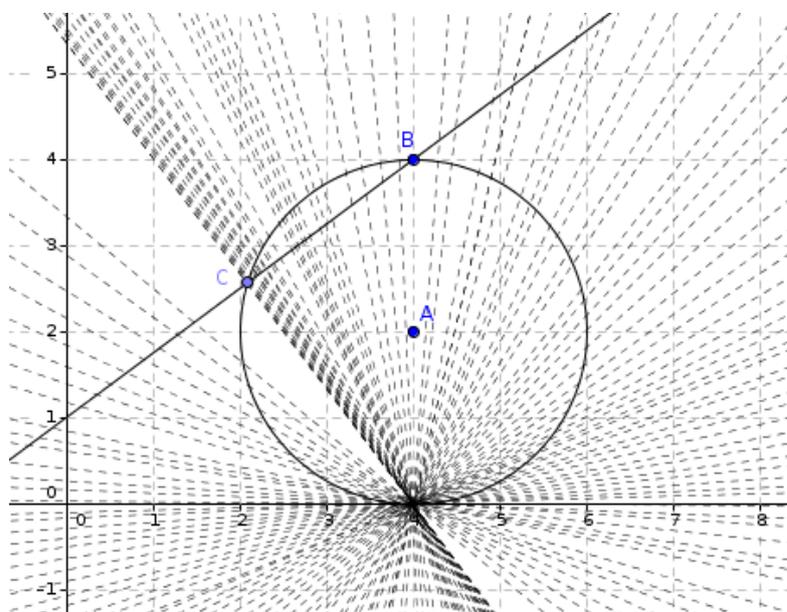


Figure 1: An envelope as displayed by tracing an element of the family.

A common approach to present envelopes consists of mimicking the traditional dynamic geometry way to plot loci. A member of the family of curves is traced while the user moves the point which determines the elements of the family. A minor improvement in some DGS automates this procedure by sampling the path of the parametric point and plotting the corresponding curve for each sample. Both strategies just return a graphic object in the screen, without any knowledge about its algebraic description. This special character obstructs treating envelopes as ordinary objects in the interactive environment (placing points on them, computing intersections with other elements, drawing tangents, ...). Even when considered as mere graphic objects, users can experience difficulties to visualize them. Consider, for instance, a circle centered at $A(4, 2)$ and passing through $B(4, 4)$, a point C moving on it, and the line BC . Since the envelope of the family of perpendicular lines to lines BC passing through C sweeps the whole plane (Figure 1), a naive user will find difficult to state that the sought envelope reduces to just a point ($(4, 0)$ for the shown construction). Note that this envelope could be easily obtained if considering it as the locus of points which are

limit of intersections of nearby curves of the family (i.e., using the definition E_2). Although the objects computed through this definition determine only partially the true envelope (see Section 1), this non-automatic approach can be used with DGS lacking a specific command, as has been suggested³ by the The Geometer's Sketchpad developers. However, it should be noted that these points of intersection are not always the limits of real intersection points. This would be the case when computing a curve as the envelope of its osculating circles, since these circles have no real intersection for infinitely close curve points [9, p. 37]. Furthermore, selecting the intersection points is not always an easy task and can be out of scope for unexperienced users, as can be checked trying selected examples [2].

As said above, some DGS automate the computation of envelopes sampling the parameter path and plotting elements of the family. Although not reported, it seems that Cabri and Cinderella combine this strategy with tracing the locus of limit intersection points in order to return the envelope as a curve. Furthermore, Cabri claims that envelope equations can also be computed, numerically approaching a curve up to degree six to a hundred intersection points. This procedure is not robust for geometric loci [3], and also returns incorrect results even for simple envelopes, as can be checked if asking for the equation of the envelope in Figure 1, where Cabri returns a line instead of the point. Cinderella, more successfully, constrains its computations to families of straight lines, being able to plot the envelope curve in a correct way. Moreover, if the envelope is at most a conic enveloping lines, Cinderella also computes the equation, similarly to the process used for computing simple loci equations [10].

3. Envelope computation with Gröbner covers

As stated above, we assume that the algebraic construction for envelopes is correct, and, following the definition E_1 (see Section 1), we will compute envelopes as projections of varieties. Thus, computing the envelope of a family $F(x, y, \alpha) = 0$ can be seen as solving the parametric polynomial system $\{F = 0, \partial F / \partial \alpha = 0\}$. Note that if the construction is parametrized by more than one α , we assume extra conditions are also specified in such a way that no fake parameters exist. That is, if the family is

³http://www.dynamicgeometry.com/Technical_Support/FAQ/Constructions_and_Use/Envelope_Constructions.html

described by $F(x, y, \alpha_1, \dots, \alpha_n) = 0$, there are given $n - 1$ extra equations $g_1 = 0, \dots, g_{n-1} = 0$ such that the Jacobian of F, g_1, \dots, g_{n-1} with respect to $\alpha_1, \dots, \alpha_n$ can be computed.

Many methods have been developed to obtain such projections (Gröbner bases, characteristic sets, discriminant varieties, ...). For example, a procedure for automatic computing envelopes within an interactive graphic environment used elimination through Gröbner bases [5]. As it is well-known, such elimination returns not only the sought projection but its closure, thus perhaps introducing spurious elements. Here we propose the use of the recently developed GröbnerCover (GC) algorithm [12]. What follows is a summary of the main properties of the GC algorithm. Note that we refer to parameters α_i as variables, and to variables x, y as parameters, according to standard terminology in parametric polynomial systems solving.

Let $\mathcal{I} = \langle \{F, g_1, \dots, g_{n-1}, \left| \frac{\partial(F, g_1, \dots, g_{n-1})}{\partial(\alpha_1, \dots, \alpha_n)} \right| \} \rangle \subset \mathbb{Q}[x, y][\boldsymbol{\alpha}]$ be a polynomial ideal in the parameters x, y and the variables $\boldsymbol{\alpha} = \alpha_1, \dots, \alpha_n$, and consider $\mathbb{V}(\mathcal{I})$ the *solution set* of the system given by \mathcal{I} :

$$\mathbb{V}(\mathcal{I}) = \{(x, y, \boldsymbol{\alpha}) \in \mathbb{C}^{2+n} : \forall f \in \mathcal{I}, f(x, y, \boldsymbol{\alpha}) = 0\}$$

Given the ideal $\mathcal{I} \subset \mathbb{Q}[x, y][\boldsymbol{\alpha}]$ (and a monomial order in the variables), its Gröbner cover provides a set of pairs $\{(S_i, B_i) : 1 \leq i \leq s\}$ describing segments S_i of the parameter space \mathbb{C}^2 , that classify points (x, y) according to the different number of solutions of the specialized system $f(x, y, \boldsymbol{\alpha}) = 0$ in the $\boldsymbol{\alpha}$ variables.

1. The segments S_i are disjoint.
2. The segments S_i are locally closed subsets of the parameter space \mathbb{C}^2 , expressed in the canonical P-representation

$$\{(\mathbf{p}_{ij}, (\mathbf{p}_{ijk} : 1 \leq k \leq s_{ij})) : 1 \leq j \leq s_i\}.$$

A segment is

$$S_i = \bigcup_j \left(\mathbb{V}(\mathbf{p}_{ij}) \setminus \left(\bigcup_k \mathbb{V}(\mathbf{p}_{ijk}) \right) \right).$$

3. There is a basis B_i associated to each segment S_i . B_i specializes to the reduced Gröbner basis of \mathcal{I} for every point $(x, y) \in S_i$ of the segment.
4. The kind of solution in the variables is given by the set of *leading power products* (lpp's) of the bases B_i , that are fixed on each GC segment S_i

(and is also explicitly given by the algorithm). Thus, for all points in the segment, the ideal \mathcal{I} has the same type of solutions (finite, infinite or no solution).

5. Moreover, if the ideal \mathcal{I} is homogeneous, then the lpp's sets are different on each segment. (The lpp's of the homogenized ideal are also explicitly given by the algorithm for each segment of S_i as they characterize the segments).

Thus, the algorithm can be used to find the projection of the surface $F(x, y, \boldsymbol{\alpha}) = 0$ into the (x, y) -plane. Note that, following 3., the union of segments with basis different from $\{1\}$ is exactly the sought projection. Furthermore, this set, being the union of constructible sets, is also constructible.

Denoting by π_1 and π_2 the projections onto the parameters and variables space, respectively:

$$\begin{array}{ccc} \pi_1 : & \mathbb{C}^{2+n} & \longrightarrow & \mathbb{C}^2 \\ & (x, y, \boldsymbol{\alpha}) & \mapsto & (x, y) \end{array} \qquad \begin{array}{ccc} \pi_2 : & \mathbb{C}^{2+n} & \longrightarrow & \mathbb{C}^n \\ & (x, y, \boldsymbol{\alpha}) & \mapsto & \boldsymbol{\alpha} \end{array} ,$$

we define the generic complex envelope E associated to the system given by \mathcal{I} as the set $E = \pi_1(\mathbb{V}(\mathcal{I})) \subset \mathbb{C}^2$. A taxonomy for computing geometric loci has been recently proposed [1]. There, a locus problem is seen as the solution of a parametric polynomial system, and the complete solution is analysed in order to return just the faithful parts of the locus. Here, this taxonomy can be rephrased as follows. A complex envelope point $(x, y) \in \mathbb{C}^2$ is *normal* if

$$\dim(\pi_2(\mathbb{V}(\mathcal{I}) \cap \pi_1^{-1}(x, y))) = 0,$$

and it is *non-normal* if

$$\dim(\pi_2(\mathbb{V}(\mathcal{I}) \cap \pi_1^{-1}(x, y))) > 0.$$

The normal points of an envelope are thus located into the GC segments with a finite number of solutions in the variables. Nevertheless, it could happen that a segment having infinite points stems from a finite set of variable values. While the locus taxonomy drops the segment out, this one could be an admissible case for envelopes. Consider, for instance, the envelope of horizontal lines through a point on the unit circle, $y = \pm 1$ (see Section 1). There is exactly a segment with basis different from $\{1\}$, namely,

$$(\mathbb{V}(y + 1) \setminus \mathbb{V}(1)) \cup (\mathbb{V}(y - 1) \setminus \mathbb{V}(1)),$$

coming from $\alpha_1 = \pm 1, \alpha_2 = 0$, respectively. So, both lines $y = \pm 1$ consist of normal points, despite they are 1-dimensional. On the contrary, consider the unit circle, a point $A(\alpha_1, \alpha_2)$ on it, and the line l through A and $(0, 1)$ (see Figure 1 for an instance of this construction). While the envelope of the family of perpendicular lines to l through A reduces to the point $(0, -1)$, the GC segments are

| | |
|---|-----------------|
| 1. Segment with $\text{lpp} = \{1\}$ Segment: $\mathbb{C}^2 \setminus \mathbb{V}(x)$ Basis: $B_1 = \{1\}$ (no solution over the segment) | Generic segment |
| 2. Segment with $\text{lpp} = \{\alpha_1, \alpha_2\}$ Segment: $\mathbb{V}(x) \setminus \mathbb{V}(y + 1, x)$ Basis: $B_2 = \{\alpha_1, \alpha_2 - 1\}$ | |
| 3. Segment with $\text{lpp} = \{\alpha_2^2\}$ Segment: $\mathbb{V}(y + 1, x) \setminus \mathbb{V}(1)$ Basis: $B_3 = \{\alpha_2^2 + \alpha_1^2 - 1\}$ | |

While segment 2, the vertical axis, consists of normal points, this line cannot be declared as part of the envelope since it derives from a degeneracy in the construction (when the point A is $(0, 1)$). The difference with the preceding case lies on this degeneracy. And since degeneracies can happen at most for a finite number of cases, testing the existence of the representative of the family of lines for such variable values helps to decide about the segment membership to the envelope. As an illustration, note that point $(0, -1)$, i.e. segment 3, is an envelope non-normal point.

4. Examples and limitations

We illustrate our proposal with the following examples. Some of them are also used to highlight its limitations towards an effective implementation in DGS: the field where solutions are found and time constraints. As said above, GC works on an algebraically closed field (\mathbb{C} in our case), so we will not try to discover semialgebraic knowledge about envelopes. Regarding the time complexity, it is well-known that Gröbner bases have the double exponential cost [11]. So, there will be some constructions where our approach will not

answer in a reasonable time lapse, where reasonable must be understood as few seconds. Although most envelope constructions in dynamic geometry will be efficiently solved, there are cases when interactivity will be lost. For such cases, we think that a smart combination of graphic methods (see Section 2) with simpler elimination techniques could make a sound framework for interactive geometry developers. While the above complex issue could be relaxed by using semialgebraic tools as Cylindrical Algebraic Decomposition and analogues, the current state of such tools is not enough mature to be incorporated as automatic helpers inside these environments.

We also want to remark that the only current implementation of the GC algorithm is in Singular, while other alternative packages exist under Maple, Mathematica, . . . Nevertheless, the open source character of GC and Singular and the deployment of a web service allowing remote access to Singular⁴ could mark the difference for including sophisticated proposals as the one here described in widespread geometric software. Let us cite, for instance, the **Envelope** command⁵, available in the forthcoming GeoGebra 5.0 beta version. This command incorporates a preliminary version of our proposal and it is, to the best of our knowledge, the first attempt to enhance DGS with efficient envelope abilities.

4.1. *The offset of a parabola*

Envelopes can be used, for instance, in geometrical optics to define caustics and in Computer-Aided Design to define offset curves. The family of circles with fixed radii d , centered at a point moving in a parabola, envelopes a sextic curve, the d -offset to the parabola. The equation of such offset can be easily obtained in DGS incorporating the GC algorithm. Figure 2 shows the 3-offset to the parabola $4y = x^2$ as computed by GeoGebra 5.0 beta. Note that no other current DGS is able to return both an accurate plot and the equation of the offset.

Since the radius of circle d is defined as the distance between E and F , one could ask for the 0-offset, i.e., the original parabola, just making both points coincident. GeoGebra would plot the offset exactly as the parabola, but the equation would be $x^4 + x^2y^2 - 6x^2y + x^2 - 4y^3 + 8y^2 - 4y = 0$, and not just $4y = x^2$. The reason behind such behavior contradicts our initial

⁴<http://code.google.com/p/singularws>

⁵http://wiki.geogebra.org/en/Envelope_Command

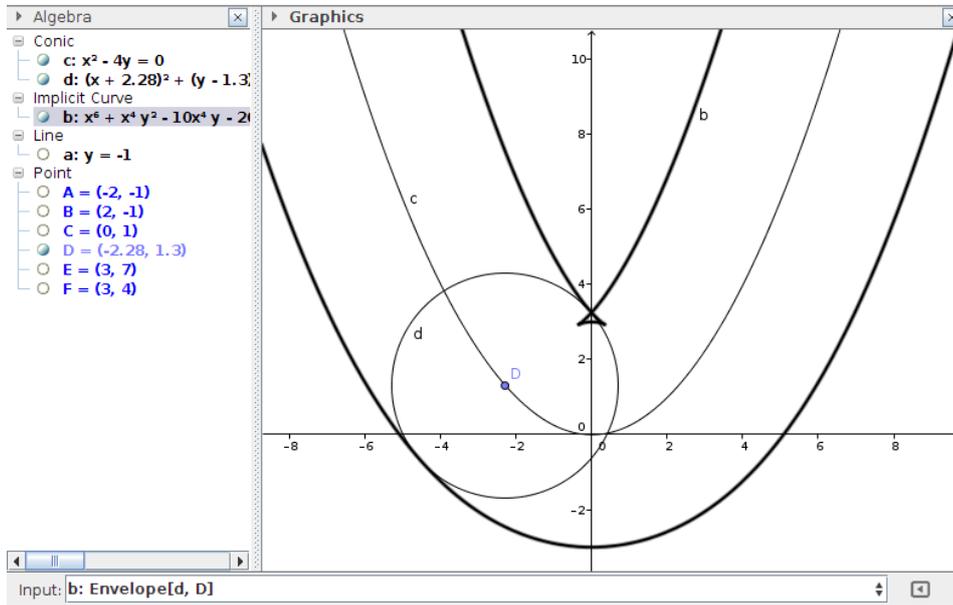


Figure 2: An offset to $4y = x^2$ computed in GeoGebra.

guess: the envelope is declared to be the parabola plus the extra point $(0, 1)$, as it can be checked inspecting the GC output.

1. Segment with $\text{lpp} = \{1\}$ Generic segment
 Segment: $\mathbb{C}^2 \setminus (\mathbb{V}(x^2 - 4y) \cup \mathbb{V}(x^2 + y^2 - 2y + 1))$
 Basis: $B_1 = \{1\}$ (no solution over the segment)
2. Segment with $\text{lpp} = \{\alpha_1, \alpha_2\}$
 Segment: $(\mathbb{V}(x^2 - 4y) \setminus \mathbb{V}(y + 1, x^2 + 4)) \cup (\mathbb{V}(x^2 + y^2 - 2y + 1) \setminus (\mathbb{V}(y - 1, x) \cup \mathbb{V}(y + 1, x^2 + 4)))$
 Basis: $B_2 = \{(y^2 - 1)\alpha_1 + (-x^3 - xy^2 + 4xy + x), (y + 1)\alpha_2 + (-x^2 - y^2 + 3y)\}$
3. Segment with $\text{lpp} = \{\alpha_2, \alpha_1^2\}$
 Segment: $\mathbb{V}(y - 1, x) \cup \mathbb{V}(y + 1, x^2 + 4)$
 Basis: $B_3 = \{(3y - 1)\alpha_2 + (2x)\alpha_1 + (-y + 3), (y)\alpha_1^2 + (2x)\alpha_1 + 4\}$

Despite the fact that GeoGebra does not plot the point $(0, 1)$ as part of

the envelope because isolated points are frequently beyond of its plotting capabilities, the computed 0-offset equation is not the true envelope. The third GC segment consists of a pair of complex points, $(\pm 2i, -1)$, discarded by their complex character, and the real point $(0, 1)$, which does not even belong to the envelope since it corresponds to complex values of α_1 and α_2 , namely $(\pm 2i, -1)$. Recall that we are following E_1 envelope definition [6, p. 102], where parameters x, y and variables α_i are constrained to \mathbb{R} . Thus, although our protocol returns sound computations in \mathbb{C} , spurious complex parts can be introduced in the obtained envelopes.

Concerning the required computing time for this offset, it should be noted that GeoGebra outsources GC computations to an external Singular web service. Using the canonical server⁶, a virtualized machine with 1 GB and 1 CPU 3 GHz, the equation of the offset shown in Figure 2 is found in 1 second. However, placing the basic points of the construction in arbitrary positions can sometimes prevent getting a result, due to the growth of the involved polynomials. In such cases the server kills the current computation after a predefined lapse (default 30 s), and the envelope is declared as *undefined*. Of course, this time could be optimized through a personal installation of the web service at the user machine.

4.2. The envelope of a family of ellipses

Consider the family of ellipses with a fixed focus $A(4, 0)$, major axis with length 5 and the other focus moving on the vertical axis, $B(0, \alpha)$. Thus, the uniparametric family $F(x, y, \alpha)$ is

$$4y^2\alpha^2 - 4y\alpha^3 - 36x^2 - 100y^2 + \alpha^4 - 32xy\alpha + 16x\alpha^2 + 164y\alpha - 82\alpha^2 + 144x + 81,$$

and simple elimination in the system $\{F = 0, \partial F/\partial \alpha = 0\}$ returns three factors for the envelope:

$$(2x + y^2 - 9)(18x - y^2 + 9)(x^2 - 8x + y^2 + 16).$$

At a first glance, both parabolas seem to be coherent with the graphic results if a user traces the family in a DGS (Figure 3). Although the parabolas envelope more curves than the ellipses of the family, it is easy to notice that the problem, as posed, is not algebraic, but semialgebraic. Since no constraint on α has been imposed, F is a hyperbola whenever $|\alpha| > 3$, and the found parabolas also envelope these hyperbolas.

⁶<http://singularws.idm.jku.at>

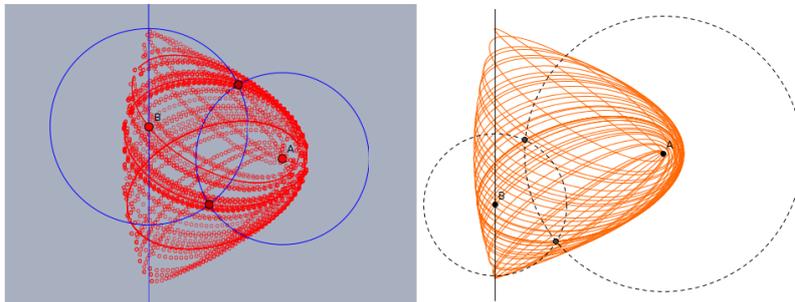


Figure 3: A trace of the family of ellipses in Cinderella (left) and GeoGebra (right).

The third factor of the envelope deserves special study. While in the preceding example we concluded that point $(0, 1)$ should be removed from the envelope, this factor, that is exactly the fixed focus, cannot be excluded as a real solution for complex values of α . The Gröbner cover of the parametric system $\{F = 0, \partial F/\partial \alpha = 0\}$, with parameters x, y and variable α , includes the segment

$$\mathbb{V}(x^2 - 8x + y^2 + 16) \setminus (\mathbb{V}(y^2 + 1, x - 5) \cup \mathbb{V}(y^2 + 81, x + 5)),$$

with basis $\{\alpha^2 + (-2y)\alpha + (8x - 41)\}$. Thus, the real point $(4, 0)$ corresponds to $x = 0, y = \pm 3$, and it cannot be discarded from the envelope. The envelope factor $x^2 - 8x + y^2 + 16$ is, if factored in \mathbb{C} , a pair of complex lines $y = \pm\sqrt{-x^2 + 8x - 16}$, whose projection on the real (x, y) -plane is exactly the point $(4, 0)$ [4]. Furthermore, note that this point can also be considered as the limit of intersection points between ellipses with α close to ± 3 and the double line representing the family for $\alpha = \pm 3$, respectively. This point belongs to E_2 envelope definition, and, thus, it is also included into E_1 definition.

5. Conclusions

We have described a protocol for the automatic computation of families of algebraic plane curves. It is based on the Gröbner Cover algorithm and the subsequent study and characterization of the returned segments. The proposal is inscribed in our general goal of improving dynamic geometry environments with new abilities related to automatic proof and discovery. The overall goal of this improvement is to support the problem solving process of users related to geometric proving and discovery.

A preliminary version of the protocol is currently supported by the beta version of GeoGebra 5.0. We discuss several examples illustrating our findings, highlighting new capabilities for envelope computations in interactive environments. Since the used framework works in the complex field, limitations of the method when dealing with real issues are also specified.

Acknowledgment

Both authors were partially supported by the Spanish “Ministerio de Economía y Competitividad” and by the European Regional Development Fund (ERDF), under the Project MTM2011–25816–C02–02.

References

- [1] M. Abánades, F. Botana, A. Montes, T. Recio, Software using the Gröbner Cover for geometrical loci computation and classification, in: *Proceedings of the 2014 International Conference on Mathematical Software*, Seoul, August 5–9, 2014.
- [2] A.R. Amir-Moez, Synthetic approach to the theory of the envelope, *Am. Math. Mon.* 64 (1957) 265–268.
- [3] F. Botana, M. Abánades, J. Escribano, Computing locus equations for standard dynamic geometry environments, *Lect. Notes Comput. Sci.* 4488 (2007) 227–234.
- [4] F. Botana, T. Recio, A propósito de la envolvente de una familia de elipses, *Bol. Soc. Puig Adam* 95 (2013) 15–30.
- [5] F. Botana, J.L. Valcarce, Automatic determination of envelopes and other derived curves within a graphic environment, *Math. Comput. Simulat.* 67 (1-2) (2004) 3–13.
- [6] J.W. Bruce, P.J. Giblin, *Curves and Singularities*, Cambridge University Press, Cambridge, 1984.
- [7] G. Dalzotto, T. Recio, On protocols for the automated discovery of theorems in elementary geometry, *J. Automat. Reason.* 43 (2009) 203–236.

- [8] X.S. Gao, Automated geometry diagram construction and engineering geometry, *Lect. Notes Artif. Intell.* 1669 (1999) 232–257.
- [9] G. Julia, *Éléments de Géométrie Infinitésimale*, GauthierVillars, Paris, 1936.
- [10] U. Kortenkamp, J. Richter-Gebert, Using automatic theorem proving to improve the usability of geometry software, in: *Proceedings of the 2004 Mathematical User-Interfaces Workshop*, Bialowieza (Poland), September 18, 2004.
- [11] E. Mayr, A. Meyer, The complexity of the word problem for commutative semigroups and polynomial ideals, *Adv. Math.* 46 (1982) 305–329.
- [12] A. Montes, M. Wibmer, Gröbner bases for polynomial systems with parameters, *J. Symb. Comput.* 45 (2010) 1391–1425.
- [13] T. Recio, F. Botana, Where the truth lies (in automatic theorem proving in elementary geometry), *Lect. Notes Comput. Sci.* 3044 (2004) 761–770.
- [14] T. Recio, M.P. Vélez, Automatic discovery of theorems in elementary geometry, *J. Automat. Reason.* 23 (1999) 63–82.