A dynamic symbolic geometry environment based on the GröbnerCover algorithm for the computation of geometric loci and envelopes

Miguel A. Abánades and Francisco Botana [⋆]

CES Felipe II, Universidad Complutense de Madrid
C/ Capitán 39, 28300 Aranjuez, Spain
abanades@ajz.ucm.es
Depto. de Matemática Aplicada I, Universidad de Vigo
Campus A Xunqueira, 36005 Pontevedra, Spain
fbotana@uvigo.es

Abstract. An enhancement of the dynamic geometry system GeoGebra for the automatic symbolic computation of algebraic loci and envelopes is presented. Given a GeoGebra construction, the prototype, after rewriting the construction as a polynomial system in terms of variables and parameters, uses an implementation of the recent GröbnerCover algorithm to obtain the algebraic description of the sought locus/envelope as a locally closed set. The prototype shows the applicability of these techniques in general purpose dynamic geometry systems.

Keywords: Dynamic Geometry, Locus, Envelope, GröbnerCover Algorithm, GeoGebra, Sage

1 Introduction

Most dynamic geometry systems (DGS) implement loci generation just from a graphic point of view, returning a locus as a set of points in the screen with no algebraic information. A simple algorithm based on elimination theory to obtain the equation of an algebraic plane curve from its description as a locus set was described in [1]. This new information expands the algebraic knowledge of the system, allowing further transformations of the construction elements, such as constructing a point on a locus, intersecting the locus with other elements, etc. The same consideration can be made with respect to the envelope of a family of curves. This algebraic approach is a significant improvement over the numeric-graphic method mentioned above. An implementation of the algorithm in a system embedding GeoGebra in the Sage notebook was described at CICM 2011 [2]. In fact, the algorithm is already behind the *LocusEquation*

---

command in the beta version of the next version of the DGS GeoGebra [3] (see `http://wiki.geogebra.org/en/LocusEquation_Command`). It has also recently been implemented by the DGS JSXGraph [4] to determine the equation of a locus set using remote computations on a server [5], an idea previously developed by the authors in [6].

Unfortunately, this algorithm does not discriminate between regular and special components of a locus (following the definitions in [7])[1]. More concretely, the obtained algebraic set may contain extra components sometimes due to the fact that the method returns only Zariski closed sets (i.e. zero sets of polynomials) and sometimes due to degenerate positions in the construction (e.g. two vertices being coincidental for a triangle construction).

There is little that can be done to solve these problems with the simple elimination approach. Concerning degeneration, there is no alternative except explicitly requesting information from the user about the positions producing special components. However, the recent GröbnerCover algorithm [8] has opened new possibilities for the automated processing of these problems. From the canonical decomposition of a polynomial system with parameters returned by the algorithm, and following a remark by Tomás Recio concerning the dimensions of the spaces of variables and parameters, a protocol has been established to distinguish between regular and special components of a locus set. For example, a circle, a variety of dimension 1, is declared to be a special component of a locus by the protocol if it corresponds to a point, a variety of dimension 0. This heuristic in the protocol improves the automatic determination of loci but does not fully resolve it. It is not difficult to find examples where this general rule does not suit the user's interests. This is a delicate issue because, in some situations, these special components are the relevant parts of the sought set (the study of bisector curves is a good source for such examples).

As an illustration, let us consider the following problem included in [9] together with a remark about its difficult synthetic treatment: *Given a triangle ABC. Take a point M on BC. Consider the orthogonal projections N of M onto AC, and P onto AB respectively. The lines AM and PN meet at X. What is the locus set of points X when M moves along the line BC?*

When the vertices of the triangle ABC are the points $(2,3)$, $(1,0)$ and $(0,1)$, the locus set is a conic from which a point has to be removed. That is, the locus set is not an algebraic variety but a locally closed set. Figure 1 shows the plotting of the conic in GeoGebra together with its precise algebraic description as provided by the prototype.

If we consider this same construction with $A(0,0), B(1,0)$ and $C(0,1)$, a standard DGS will plot a straight line as locus, while ordinary elimination will give the true locus $2x + 2y = 1$ plus two other lines, namely, the coordinate axes $x = 0$ and $y = 0$. These extra lines correspond to two degenerate positions for the mover: $M = B$ and $M = C$. Applying the criterion sketched above, the system identifies these two lines as special components and hence removes them

---

[1] A special component of a locus is basically a one-dimensional subset of the locus corresponding to a single position of the moving point.
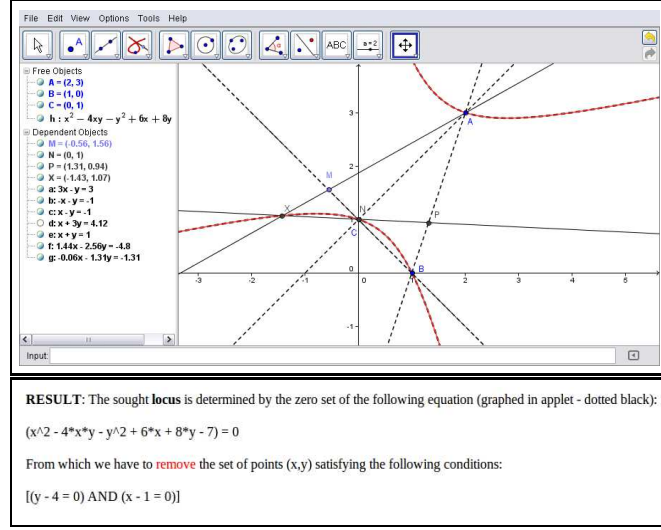
Fig. 1. Locus as a constructible set.

from the final description. In tables 1 and 2 we find the parametric systems and outputs from the GröbnerCover algorithm for the two considered instances respectively.

2   Prototype description

The system (accessible at [10]) consists of a web page with a GeoGebra applet where the user constructs a locus or a family of linear objects depending on a point. For any of these constructions (specified using a predetermined set of GeoGebra commands) the prototype provides the algebraic description of the locus/envelope by just pressing one button. Note that in its current state, the system does not provide the equation of the envelope, but the one of the discriminant line. A note stating this fact should be given if using the system for teaching purposes.

The process is roughly as follows. First, the XML description of the GeoGebra construction is sent to a Server where an installation of a Sage Cell Server ([11]) is maintained by the authors. There, the construction follows an algebraization process, as specified by a Sage library [12]. The communication Sage-GeoGebra is made possible by the JavaScript GeoGebra functions that allow the data transmission to and from the applet. In particular, the XML description of any GeoGebra diagram can be obtained. The processing of the XML description of the diagram is made by some ad-hoc code by the authors that use Sage through the Sage cell server, a general service by Sage. More concretely, Singular, included in Sage and with an implementation of the GröbnerCover algorithm, is used. The

Parametric System	$-x_1 - x_2 + 1, 2x_1 + 2x_2 - 2x_3 - 2x_4, -2x_3 + 2x_4 - 2, x_1 + 3x_2 - x_5 - 3x_6, -3x_5 + x_6 + 3, -(x_4 - y)(x_3 - x_5) + (x_4 - x_6)(x_3 - x), -(y - 3)(x_1 - 2) + (x - 2)(x_2 - 3)$
Basis segment 1	$\{1\}$
Segment 1	$\mathbb{V}(0) \setminus \mathbb{V}(x^2 - 4xy + 6x - y^2 + 8y - 7)$
Basis segment 2 *	$\{\{(5y - 20)x_6 + (-3x + 6y + 3), (x - 2y + 7)x_6 + (-3y)\}, \{(5y-20)x_5+(-x-3y+21), (x-7y+27)x_5+(4y-28)\}, x_4-1, x_3, \{(y-4)x_2+(-x+2y+1), (x-2y+7)x_2+(-5y)\}, \{(y-4)x_1+(x-3y+3), (x-y+3)x_1+(4y-4)\}\}$
Segment 2	$\mathbb{V}(x^2 - 4xy + 6x - y^2 + 8y - 7) \setminus \mathbb{V}(y - 4, x - 1)$
Basis segment 3	$\{1\}$
Segment 3	$\mathbb{V}(y - 4, x - 1) \setminus \mathbb{V}(1)$
Locus (after heuristic step)	$\mathbb{V}(x^2 - 4xy - y^2 + 6x + 8y - 7) \setminus \mathbb{V}(y - 4, x - 1)$

Table 1. Parametric system, GröbnerCover output and returned locus for instance with $A(2,3)$, $B(1,0)$ and $C(0,1)$.
* Includes regular functions (see [8]).

Parametric System	$-x_1 - x_2 + 1, x_3, -x_2 + x_4, -x_1 + x_5, -x_6, -x_1 y + x_2 x, -(x_4 - y)(x_3 - x_5) + (x_4 - x_6)(x_3 - x)$
Basis segment 1	$\{1\}$
Segment 1	$\mathbb{V}(0) \setminus (\mathbb{V}(2x + 2y - 1) \cup \mathbb{V}(x) \cup \mathbb{V}(y))$
Basis segment 2	$\{x_6, (x + y)x_5 - x, (x + y)x_4 - y, x_3, (x + y)x_2 - y, (x + y)x_1 - x\}$
Segment 2	$(\mathbb{V}(2x+2y-1) \setminus \mathbb{V}(1)) \cup (\mathbb{V}(x) \setminus \mathbb{V}(x,y)) \cup (\mathbb{V}(y) \setminus \mathbb{V}(x,y))$
Basis segment 3	$\{x_6, x_4 + x_5 - 1, x_3, x_2 + x_5 - 1, x_1 - x_5, x_5^2 - x_5\}$
Segment 3	$\mathbb{V}(x, y) \setminus \mathbb{V}(1)$
Locus (after heuristic step)	$\mathbb{V}(2x + 2y - 1)$

Table 2. Parametric system, GröbnerCover output and returned locus for instance with $A(0,0)$, $B(1,0)$ and $C(0,1)$.

Gröbner cover of the obtained parametric polynomial system is analyzed, and the accepted components of the locus/envelope are incorporated into the applet.

Note that the goal is not to provide a final tool but a proof-of-concept prototype showing the feasibility of using sophisticated algorithms like GröbnerCover to supplement the symbolic capabilities of existing dynamic geometry systems, as well as to show the advantage of connecting different systems by using web services.

References

1. Botana, F., Valcarce, J.L.: A software tool for the investigation of plane loci. Mathematics and Computers in Simulation **61(2)** (2003) 139–152
2. Botana, F.: A symbolic companion for interactive geometric systems. In Davenport, J.H., Farmer, W.M., Urban, J., Rabe, F., eds.: Calculemus/MKM. Volume 6824 of Lecture Notes in Computer Science., Springer Berlin Heidelberg (2011) 285–286
3. GeoGebra: `http://www.geogebra.org` (Last accessed January 2013)
4. JSXGraph: `http://jsxgraph.org/` (Last accessed January 2013)
5. Gerhäuser, M., Wassermann, A.: Automatic calculation of plane loci using Gröbner bases and integration into a dynamic geometry system. In Schreck, P., Narboux, J., Richter-Gebert, J., eds.: Automated Deduction in Geometry. Volume 6877 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2011) 68–77
6. Escribano, J., Botana, F., Abánades, M.A.: Adding remote computational capabilities to dynamic geometry systems. Mathematics and Computers in Simulation **80** (2010) 1177–1184
7. Sendra, J.R., Sendra, J.: Algebraic analysis of offsets to hypersurfaces. Matematische Zeitschrift **234** (2000) 697–719
8. Montes, A., Wibmer, M.: Gröbner bases for polynomial systems with parameters. Journal of Symbolic Computation **45** (2010) 1391–1425
9. Guzmán, M.: La experiencia de descubrir en Geometría. Nivola (2002)
10. Locus/Envelope Prototype: `http://193.146.36.205:8080/GgbSageDirect/LocusEnvelope/` (2012)
11. Simple sagecell server: `https://github.com/sagemath/sagecell` (Last accessed January 2013)
12. Botana, F.: On the parametric representation of dynamic geometry constructions. In Murgante, B., Gervasi, O., Iglesias, A., Taniar, D., Apduhan, B., eds.: Computational Science and Its Applications - ICCSA 2011. Volume 6785 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2011) 342–352