

Computing Locus Equations for Standard Dynamic Geometry Environments

Francisco Botana, Miguel A. Abánades, and Jesús Escribano

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

`fbotana@uvigo.es`

Ingeniería Técnica en Informática de Sistemas, CES Felipe II (UCM), 28300
Aranjuez, Spain

`mabanades@cesfelipesecondo.com`

Departamento de Sistemas Informáticos y Computación, Universidad Complutense de
Madrid, 28040 Madrid, Spain

`escribano@sip.ucm.es`

Abstract. GLI (*Geometric Locus Identifier*), an open web-based tool to determine equations of geometric loci specified using *Cabri Geometry* and *The Geometer's Sketchpad*, is described. A geometric construction of a locus is uploaded to a Java Servlet server, where two computer algebra systems, CoCoA and Mathematica, following the Groebner basis method, compute the locus equation and its graph. Moreover, an OpenMath description of the geometric construction is given. GLI can be efficiently used in mathematics education, as a supplement of the locus functions of the standard dynamic geometry systems. The system is located at <http://nash.sip.ucm.es/GLI/GLI.html>.

Keywords: Interactive geometry, Automated deduction, Locus, OpenMath.

1 Introduction

Dynamic geometry programs are probably the most used computers tools in mathematics education today, from elementary to pregraduate level. At the same time, computer algebra systems are widely employed in the learning of scientific disciplines, generally starting from high school. Nevertheless, the top ranked programs in both fields have evolved separately: although almost all of the computer algebra systems offer some specialized library for the study of geometry (cf. Mathematica, Maple, Derive, ...), none of them use the dynamic paradigm, that is, in their geometric constructions free elements cannot be graphically dragged making dependent elements behave accordingly. On the other side, standard dynamic geometry environments, such as Cabri Geometry [11] and The Geometer's Sketchpad [10], are self-contained: whenever, albeit rarely, they need some symbolic computation, they use their own resources.

Some attempts connecting both types of systems have been reported, mainly coming from academia. Besides the above mentioned geometric libraries, specialized packages for geometry using the symbolic capabilities of computer algebra

systems exist (see, for instance, [1]). Nevertheless, they lack the dynamic approach, being more an environment for exact drawing than a tool for interactive experiments. We can also cite [12] as an interesting work, where a dynamic geometric system is developed inside Mathematica through an exhaustive use of MathLink.

The approach of using computer algebra algorithms in dynamic geometry has been more fruitful. Two ways have been devised for dealing with this cooperation. Some systems incorporate their own code for coping with algebraic techniques in geometry ([14,8,9],...), while other systems emphasize on reusing software ([2,15,17],...). Both strategies have been partially successful on solving some of the three main points in dynamic geometry, that is, the continuity–determinism dilemma, the proof and discovery abilities, and the complete determination of loci (see [6] for an extensive study of the subject).

This paper describes a web–based resource allowing the remote discovery of equations of algebraic loci specified through the well–known Cabri and The Geometer’s Sketchpad environments. Moreover, OpenMath has been chosen as the communicating language allowing other OpenMath compliant systems to make use of the tool.

2 Numerical vs. Symbolic Loci

An astonishing feature of dynamic geometry systems is their ability to draw the path of an object dependent on another one while this last element is dragged along a predetermined path. The trajectory of the first object is called its *locus*. Since their earliest versions, both Cabri and The Geometer’s Sketchpad offered graphical loci generation through their trace mode. Roughly speaking, the strategy consists of sampling the path of the dragged object and constructing, for each sample, the position of the point generating the locus. An interpolation of these support points returns the locus as a graphical continuous object in the screen. The heuristics used for the interpolation produce anomalous loci in some border cases, and this strategy is not well suited to produce the equation of the obtained loci. Reacting against this drawback, the newest version of Cabri incorporates a tool for computing approximate algebraic expressions of loci. Although this new feature is not documented, as usual in the market considerations of the Cabrilog company, the algorithm is sketched in [16] as follows: 1) random selection of about one hundred locus supporting points, and 2) calculation of the $(n + 1)(n + 2)/2$ coefficients of the bivariate locus equation of degree n , from $n = 1$ onwards and using a system of equations derived from the support points, until they approximately satisfy the locus equation.

There is no doubt about the interest of functions returning equations of loci, as Cabri does although in an approximate way. Nevertheless, no comment is made on the exactness of the equation (hence inducing an unexpert user to take it as an accurate one). Furthermore, some loci are described by equations of different degree if the user exploits the dynamic character of the environment. We have shown in [3] the superior performance of symbolic approaches for dealing with

loci: our proposal also obtains the equations of algebraic loci and bans out some anomalous results in standard systems. The equations are sound (since the used Groebner bases method is), and the method is fast enough to be integrated in a dynamic environment, as tested with GDI (see [4] where the detailed algorithm is described).

3 User Interface and Architecture

Although conceived with the spirit of a plug-in for Cabri and The Geometer’s Sketchpad, simplicity and convenience of use were the guiding lines in the design of GLI. Consequently, its interface (<http://nash.sip.ucm.es/GLI/GLI.html>) has been designed to look like a simple web page (see Figure 1). The user makes use of an applet to create a text file that is uploaded to the server. The equation and graph of the locus are then displayed in a new browser window.



Fig. 1. User interface

The Cabri or The Geometer’s Sketchpad file with the original specification of the locus undergoes a double translating process. First the original file is processed as a text file producing an XML codification of an OpenMath element. OpenMath is an extensible standard for representing the semantics of mathematical objects (see [13]). This OpenMath description is then translated into webDiscovery code, which is the description that the final application is designed

to interpret. webDiscovery is a web application developed by Botana (see [5]) capable of performing a wide variety of discovery tasks whose kernel has been appropriately modified to be integrated in GLI as the final computational tool. Unlike the files generated by Cabri, the files directly generated by The Geometer's Sketchpad are coded. A JavaSketchpad .htm file has to be used instead.

The decision of making the whole translating process available to the user was thought, on one hand, as a testimonial statement to the computational community where lack of transparency stands in the way of attempts to use the available tools. On the other hand, the OpenMath description of the geometric locus is made available to other OpenMath compliant geometric systems.

GLI is based on webMathematica [18], a Java servlet technology allowing remote access to the symbolic capabilities of Mathematica. Once the user has created and uploaded the appropriate text file, a Mathematica Server Page is launched, reading the file and initializing variables. An initialization file for CoCoA [7] containing the ideal generated by the appropriate defining polynomials is also written out, and CoCoA, launched by Mathematica, computes a Groebner basis for this ideal. The returned factors are classified as points, lines, conics or general curves. Although Mathematica provides an implementation of the Groebner basis algorithm, CoCoA has been chosen mainly due to the experience of better performance in several appropriate examples. Additionally, the Mathematica graphing abilities are used to plot the locus.

4 Examples

4.1 An Ellipse

The old exercise of drawing an ellipse using a pencil, two pins, and a piece of string is frequently proposed as the first one when young students begin practising loci computation.

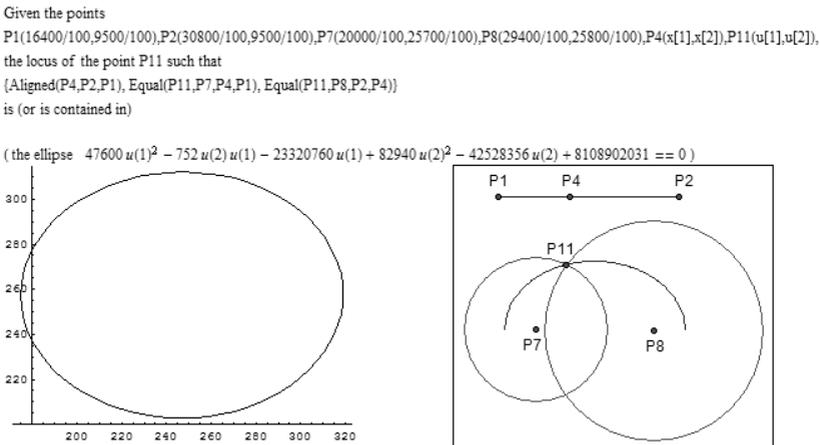


Fig. 2. The ellipse as a locus

It is well known that both environments, Cabri and The Geometer’s Sketchpad, return a surprising answer when simulating the construction, giving just half of the ellipse. Figure 2 shows the construction made in The Geometer’s Sketchpad (inside the square) and the answer returned by GLI. The plot range in the graph produced by GLI depends on ad-hoc computations of the graph size made by Mathematica. Despite being an independent process, the computations of an optimum plot range can be directed by a line of code in the webDiscovery description of the task. A default but modifiable value has been included in all webDiscovery descriptions.

4.2 Limaçon of Pascal

Given a fixed point P_3 and a variable one P_6 on a circle, the limaçon of Pascal is the locus of points P_{14} such that P_3, P_6 and P_{14} are collinear and the distance from P_6 to P_{14} is a constant, specified in Figure 4 by the segment P_8P_9 (see <http://www-groups.dcs.st-and.ac.uk/~history> for a historical reference).

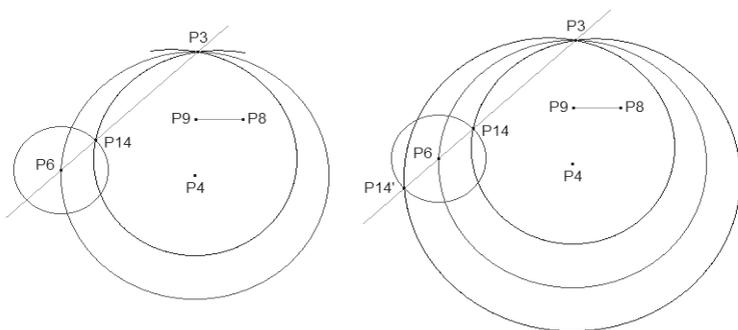


Fig. 3. The limaçon of Pascal in Cabri

As in the case of the preceding subsection, computing the locus of P_{14} gives just a part of it (Figure 3, left). It is necessary to compute the locus of the other intersection, P'_{14} , in order to get the whole locus (Figure 3, right). It seems that Cabri is making some extra assumptions, for instance, the point P_{14} is *inside* the circle, whereas the imposed constraints subsume both points in just one. Regarding the locus equation, Cabri returns two equations (instead of one!)

$$\begin{aligned}
 0.04x^2y + 0.04y^3 + 0.22x^2 + 0.19xy + 0.82y^2 + 0.87x + 5.07y + 10 &= 0 \\
 0.04x^2y + 0.04y^3 + 0.17x^2 + 0.20xy + 0.95y^2 + 0.77x + 5.62y + 10 &= 0.
 \end{aligned}$$

Plotting these equations with Mathematica we get the curves in Figure 4, left, while the curve returned by our system GLI is shown at the right. The equation of the limaçon, a quartic, is included as a factor of the solution (see Figure 5). The extraneous factor of the circle is due to a degenerated condition: note that,

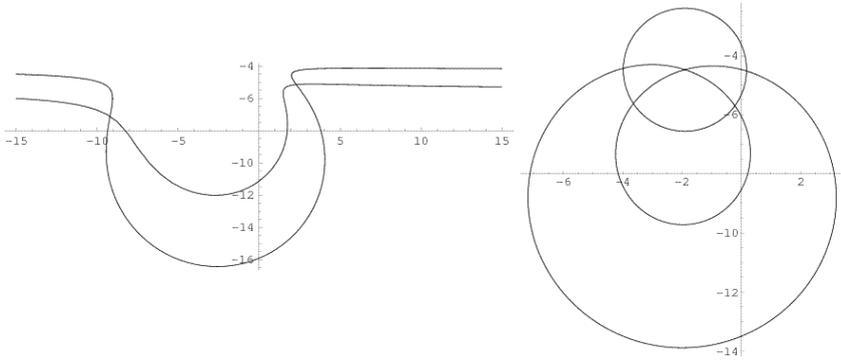


Fig. 4. Plots of the limaçon

since P_6 is a variable point on the circle, it can coincide with P_3 , so reducing the locus to a circle centered at P_3 and with radius P_8P_9 . The generation of this factor could be avoided in GLI by adding the condition `NotEqual(P3,P6)`.

$$\left(\begin{array}{l} \text{the circle} \quad 2000 u(1)^2 + 7600 u(1) + 2000 u(2)^2 + 17920 u(2) + 38703 == 0 \\ \text{the quartic} \quad 25000000 u(1)^4 + 198000000 u(1)^3 + 50000000 u(2)^2 u(1)^2 + 814000000 u(2) u(1)^2 + \end{array} \right.$$

Fig. 5. A fragment of the equation returned for the limaçon

4.3 A Simple Locus, Different Answers

We will use a very simple example to illustrate the different behavior of standard systems and the one proposed here. Let us consider a line P_1P_2 with a variable point on it, P_4 , taken as the center of a circle with radius P_5P_6 . Compute the locus of a point P_9 bound to the circle when P_4 moves along the line. The locus found by The Geometer’s Sketchpad is, as expected, a line parallel to P_1P_2 (Figure 6). Nevertheless, GLI answers that *The locus is (or is contained in) the whole plane*. This is due to the different treatment of point P_9 : in standard systems its definition is taken not just as a point on the circle, but other constraints not explicitly given are assumed. However, from an strictly algebraic point of view, P_9 is *any* point on the circle. So, when the circle moves, the locus is a non linear part of the plane, and our non semialgebraic approach detects this fact answering as stated.

Note that if Cabri is used and the first line is defined through a point and a graphically specified slope, the translation of the construction will fail since the current version of GLI does not support this Cabri primitive. When working with a non allowed locus file an error message will appear in the corresponding text area in the applet. The user is then instructed to inspect the JAVA console

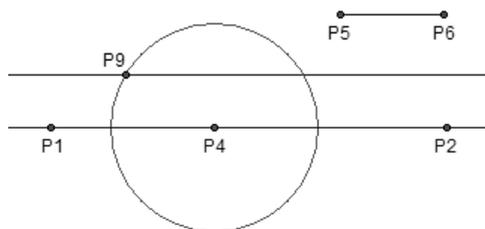


Fig. 6. A simple locus

to see a sequential description of the translation process, with which one can determine the primitive not admitted by the current version of GLI that has been used.

4.4 Extending the Scope of Loci Computations

Cabri and The Geometer's Sketchpad can only find loci of points that have been effectively constructed in their systems, that is, points who parametrically depend on another one. Hence, points which are *implicitly* defined by simultaneous multiple conditions cannot be used for generating loci. A classical result such as the theorem of Wallace–Steiner (stating that the locus of points X such that their orthogonal projections to a given triangle are collinear is the triangle's circumcircle) cannot be discovered unless the user previously knows the result! The symbolic kernel of GLI has been designed to be easily modified to support this type of implicit loci. Further versions of GLI are under development to support an extended class of computable loci.

5 Conclusion

The work presented here, although of a small magnitude, shows the possibilities of the interconnection between dynamic geometry and computer algebra systems. Moreover, we think that the generalization of broad band internet connections will make remote access to applications the main general trend, of which GLI is a perfect example. The decision of making OpenMath the communication language between the systems involved could be seen as secondary from a computational point of view. However, the use of standard semantic representation of mathematical objects is, as we see it, the main challenge in the computational community. GLI wants to be an example of that too. Moreover, the use of OpenMath as intercommunicating language opens the door to further connections with different geometry related systems. As future work, a two-folded ongoing research is being conducted to extend GLI's domain, to other Dynamic Geometry Systems on one hand, and to nonpolynomial equations and inequalities on the other. This latter extension of GLI will allow a considerable increase in the set of possible relations between the different geometric elements and hence of its applications.

Acknowledgments. This work has been partially supported by UCM research group ACEIA and research grants MTM2004-03175 (Botana) and MTM2005-02865 (Abánades, Escribano) from the Spanish MEC.

References

1. Autin, B.: Pure and applied geometry with Geometrica. *Proc. 8th Int. Conf. on Applications of Computer Algebra* (ACA 2002), 109–110 (2002)
2. Botana, F., Valcarce, J.L.: A dynamic-symbolic interface for geometric theorem discovery. *Computers and Education*, 38(1–3), 21–35 (2002)
3. Botana, F.: Interactive versus symbolic approaches to plane loci generation in dynamic geometry environments. *Proc. I Int. Workshop on Computer Graphics and Geometric Modeling* (CGGM 2002), LNCS, 2330, 211–218 (2002)
4. Botana, F., Valcarce, J.L.: A software tool for the investigation of plane loci. *Mathematics and Computers in Simulation*, 61(2), 141–154 (2003)
5. Botana, F.: A Web-based intelligent system for geometric discovery. *Proc. I Int. Workshop on Computer Algebra Systems and Applications* (CASA 2003), LNCS, 2657, 801–810 (2003)
6. Botana, F., Recio, T.: Towards solving the dynamic geometry bottleneck via a symbolic approach. *Proc. V Int. Workshop on Automated Deduction in Geometry* (ADG 2004), LNAI, 3763, 92–110 (2006)
7. Capani, A., Niesi, G., Robbiano, L.: CoCoA, a system for doing Computations in Commutative Algebra. Available via anonymous ftp from: cocoa.dima.unige.it
8. Gao, X.S., Zhang, J.Z., Chou, S.C.: *Geometry Expert*. Nine Chapters, Taiwan (1998)
9. <http://www.geogebra.at>
10. Jackiw, N.: *The Geometer's Sketchpad v 4.0*. Key Curriculum Press, Berkeley (2002)
11. Laborde, J. M., Bellemain, F.: *Cabri Geometry II*. Texas Instruments, Dallas (1998)
12. Miyaji, C., Kimura, H.: Writing a graphical user interface for Mathematica using Mathematica and Mathlink. *Proc. 2nd Int. Mathematica Symposium* (IMS'97), 345–352 (1997)
13. <http://www.openmath.org/>
14. Richter-Gebert, J., Kortenkamp, U.: *The Interactive Geometry Software Cinderella*. Springer, Berlin (1999)
15. Roanes-Lozano, E., Roanes-Macías, E., Villar, M.: A bridge between dynamic geometry and computer algebra. *Mathematical and Computer Modelling*, 37(9–10), 1005–1028 (2003)
16. Schumann, H.: A dynamic approach to simple algebraic curves. *Zentralblatt für Didaktik der Mathematik*, 35(6), 301–316 (2003)
17. Wang, D.: GEOTHER: A geometry theorem prover. *Proc. 13th International Conference on Automated Deduction* (CADE 1996), LNCS, 1104, 166–170 (1996)
18. <http://www.wolfram.com/products/webmathematica/index.html>