

Interactive versus Symbolic Approaches to Plane Loci Generation in Dynamic Geometry Environments

Francisco Botana

Departamento de Matemática Aplicada
Universidad de Vigo
Campus A Xunqueira, 36005 Pontevedra, Spain
fbotana@uvigo.es

Abstract. This paper reviews current approaches to plane loci generation within dynamic geometry environments. Such approaches are classified as interactive, when just a plot of the locus is shown, and symbolic, if, in addition to plotting the locus, its equation is also given. It is shown how symbolic approaches outperform the interactive ones when dealing with loci which are algebraic curves. Additionally, two experimental improvements are reported: i) an efficient computer algebra system allows symbolically generated loci to behave as dynamic objects, and ii) a general purpose computer algebra system is used to remove spurious parts of some loci.

1 Introduction

Dynamic geometry software refers to computer programs where accurate construction of geometric configurations can be done. The key characteristic of this software is that unconstrained parts of the construction can be moved and, as they do, all other elements automatically self-adjust, preserving all dependent relationships and constraints [13]. An immediate consequence of this behavior allows us to keep track on the path of an object that depends on another object while this one is dragged. If the dependent object is a point, its trace gives a locus, whereas if it has higher dimensionality, the path can be used to suggest related geometric elements, such as envelopes. Most dynamic geometry software implements loci generation just from a graphic point of view, returning them as states of the screen. The loci obtained in this way are restricted by the allowed transformations in the system, no algebraic information about them is known, and, sometimes, they become aberrant for particular positions of the construction. On the other side, the application of symbolic methods for loci generation, although restricted to algebraic curves, generalizes the class of obtainable loci, returns their algebraic expression, and behaves in a uniform way for all construction instances.

The structure of the paper is as follows. Section 2 describes current approaches to loci generation in classic dynamic geometry environments and in

two more recent programs. Symbolic methods for the problem are briefly recalled in Section 3, and a successful implementation of one of such methods into a dynamic geometry program is reported. Finally, some ways for further development in this field are proposed.

2 Interactive Approaches

The introduction, in the late eighties, of The Geometer's Sketchpad [9] and CABRI [16] marked the birth of dynamic geometry software. The Geometer's Sketchpad, GSP, was a project developed by Jackiw, whereas CABRI, called today Cabri Geometry, was designed by an interdisciplinary team led by Laborde. Both programs share the strategy for tracing loci: selecting an object, the *driver object*, with a predefined path, the locus of another object depending on the former is drawn by sampling the path and plotting the locus object for each sample. The driver object, in both GSP and Cabri, must be a point, whereas the path can be any linear object, arc, or even a locus, containing the driver object.

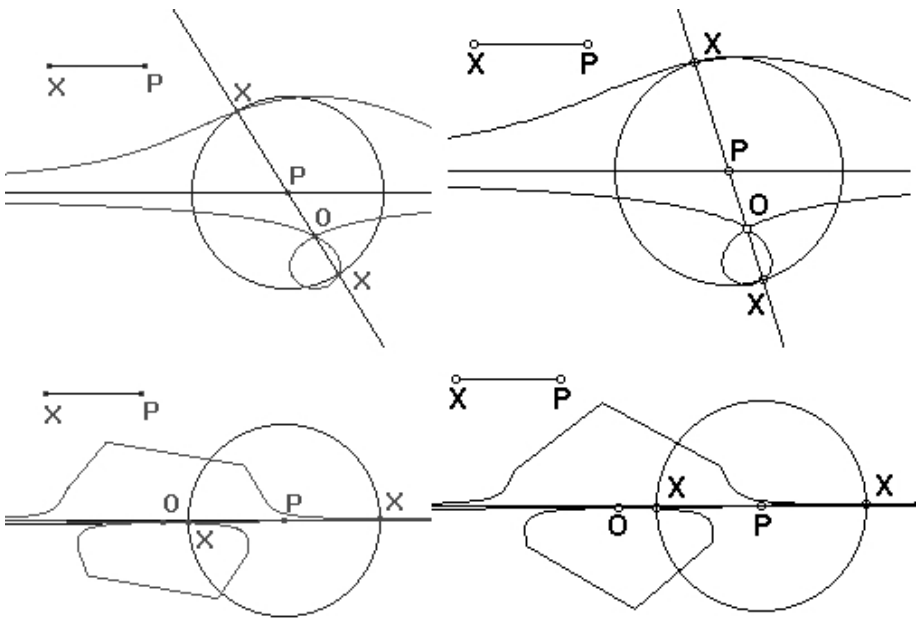


Fig. 1. Conchoid of Nicomedes in Cabri (left) and GSP (right)

If the locus object is a point (other options are linear objects and arcs), an option allows the user to fit a curve, by linear interpolation, to the locus points,

thus returning the locus as a usually continuous curve. Nevertheless, the uniform division of the path can produce anomalous loci. For the sake of illustration, let us consider the conchoid of Nicomedes as a locus. Given a point O and another point P lying on a line, the conchoid of Nicomedes is the locus of points X such that X, O and P are collinear, and the distance between X and P is constant. Both GSP and Cabri easily find the curve (Fig. 1). But note that if O is *close* to the line, the linear interpolation fails to return a correct locus, even setting the number of samples to its maximum value. Some contiguous positions of P on its path, the horizontal line, produce successive not-near positions of X .

Geometry Expert [7] tries to avoid this problem returning loci only as sequences of points and widening the range of sampling (Fig. 2). But this behavior cannot be corrected within this graphic approach, since small changes of some elements in a construction can sometimes produce sudden jumps of dependent objects.

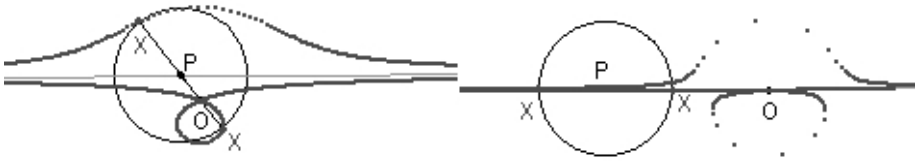


Fig. 2. Conchoid of Nicomedes in Geometry Expert

Using complex analysis, Cinderella [14, 18] has solved the problem of continuity, but, with regard to loci generation, it shares the problems of the interpolating approach of GSP and Cabri, as illustrated with the conchoid in Fig. 3. Furthermore, its strategy of returning loci as the positions only accessible by real continuous moves [14, p. 137] sometimes avoids their correct generation. Fig. 4 shows, in Cinderella, the locus of a point whose product of distances to F_1 and F_2 is constant (the similar triangles ABC and DBE in the left are used to multiply). The program finds just one of the pair of Cassinian ovals for this configuration.

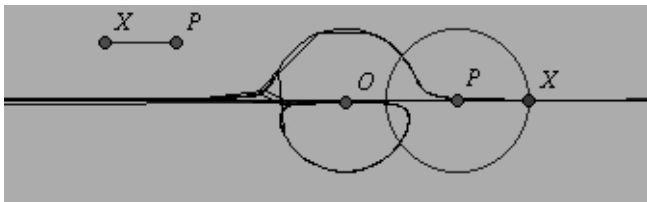


Fig. 3. Conchoid of Nicomedes in Cinderella

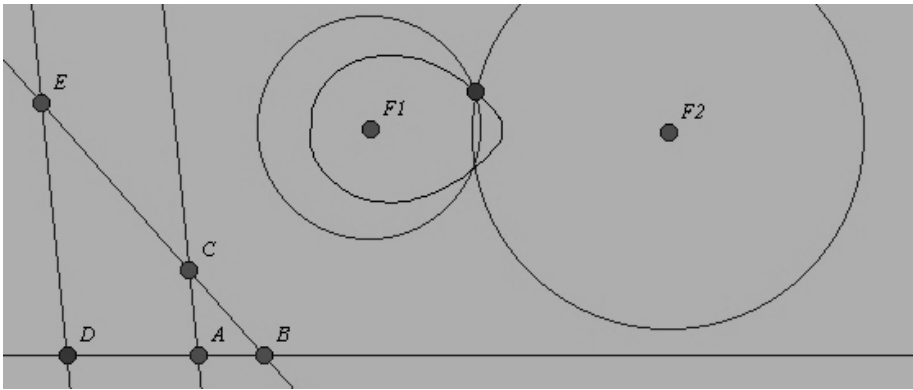


Fig. 4. Cinderella finds just a Cassinian oval

3 Symbolic Approaches

The main claim in this paper is that the above ways for generating loci are outperformed by a new class of methods from algebraic geometry. Although they apply only to algebraic curves, the amount and ubiquity of such curves would justify using these methods. Furthermore, there exists another reason to support their use. Up to now, no dynamic geometry software could obtain the locus of a non-dependent object, thus excluding a huge class of them. Symbolic approaches deal with locus points in a uniform way: it does not matter whether the point is dependent or there are other objects that depend on it.

Symbolic generation of loci can be seen as a subfield of geometric discovery, which is also related to automatic theorem proving. Constructive methods such as Groebner bases [4] or Wu's method [23] have been much more successful in automatic proving than earlier attempts based on logical approaches.

On the one hand, the method of Wu has been used for loci discovery by Chou [6], where Steiner theorem is rediscovered, and by Roanes and Roanes in the plane [19] and the space [20]. Both uses are purely algebraic in the sense that no graphical environment for diagram construction is provided, and they require human intervention, mainly when setting the order of algebraic operations. Geometry Expert, which incorporates the method of Wu among its impressive set of automatic provers, has not been designed for automatic generation of loci using its symbolic capabilities. The use of Wu's method for a true automatic generation of loci within a dynamic geometry environment remains unexplored.

On the other side, Groebner bases have been widely used for automatic theorem proving [10–12, 15]. A recent work due to Recio and Vélez [17] emphasizes using Groebner bases for automatic discovery, rather than for automatic proving. It proposes linking Cabri with the Groebner basis method for automatic discovery in an intelligent program for learning Euclidean geometry. Preliminary steps in this direction, with a new dynamic geometry environment, have been reported in [1, 2]. Narrowing the goal of automatic geometry discovery to

loci discovery, Groebner bases allow the easy obtaining of equations and plots of algebraic curves that are the locus of a point. Lugares [21] links a dynamic geometry software with Mathematica [22] and CoCoA [5]. Basically, it allows the user to draw a geometric construction and it returns the locus of a point on which some extra conditions are stated. In this way, the locus point can constrain some other elements of the construction. The construction and the locus point conditions are translated into polynomial equations, and an elimination process, using Groebner bases, is carried out on them. The geometric-dependent variables are eliminated, leaving a set of polynomials in the independent variables. This set, seeing the locus point variables as indeterminates and all of the rest as parameters, is the locus searched for (see [3] for a technical description). Let us consider, for example, the locus of a point such that its perpendicular projections on the sides of a triangle determine another triangle whose oriented area is k . There is no way to find this locus with the interactive approach taken in most dynamic geometry systems, whereas Lugares finds, in no appreciable time, that the locus is a circle (the circumcircle for $k = 0$ –Simson theorem–, and a concentric circle for $k \neq 0$ –Steiner theorem), as shown in Fig. 5 for Simson theorem.

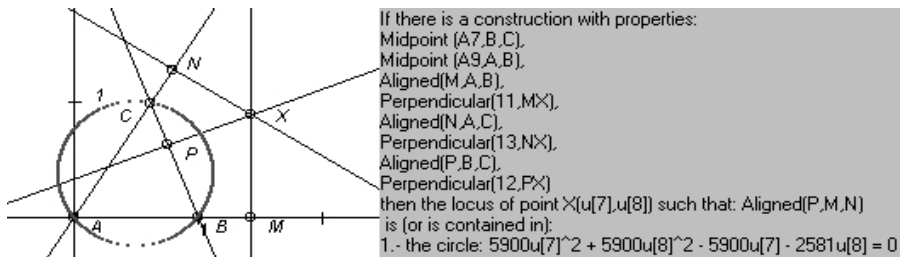


Fig. 5. The locus of X such that M, N, P are collinear is the circumcircle

Despite the doubly exponential computational cost of Buchberger's algorithm, most loci in elementary geometry are quickly computed by CoCoA, thus allowing their dynamic behavior when dragging any other element. Fig. 6 illustrates this assertion with a particular case of a recent extension of Simson theorem [8]: Given a triangle ABC and three directions, the locus of points X such that its projections on the sides are collinear is a conic. For a certain instance of the construction, the conic is an ellipse, while dragging C it becomes a hyperbola. Although the elapsed time between both drawings cannot be generally stated (it depends on the actual coordinates of the construction points), empirical findings show that the time between successive drawings moves between 0 and 2 seconds on a conventional PC.

In turn, Mathematica is considerably slower than CoCoA when dealing with Groebner bases, hence it is not used to show dynamic loci. Nevertheless, being

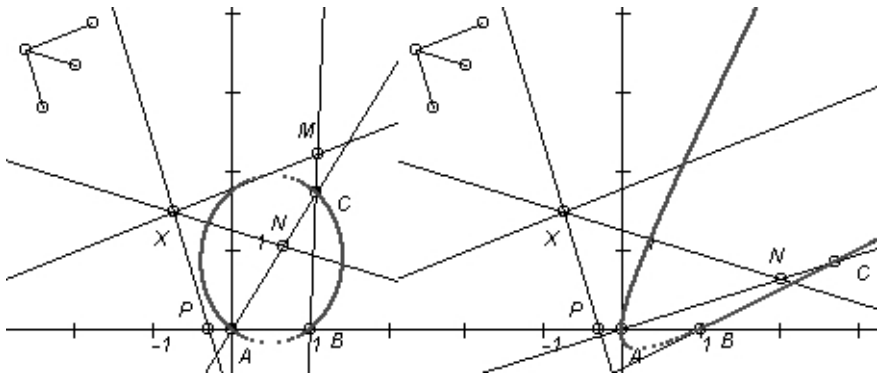


Fig. 6. Dynamic behavior of a locus

a general-purpose symbolic system, it allows to move a step forward in loci obtaining. As is well known, geometric relations such as *between* can not be expressed through polynomial equations. Let us take into account, for example, the task of separating triangles, that is, given a triangle ABC and a point P on segment AB , find all possible positions of C such that triangles APC and PCB are isosceles (Fig. 7).

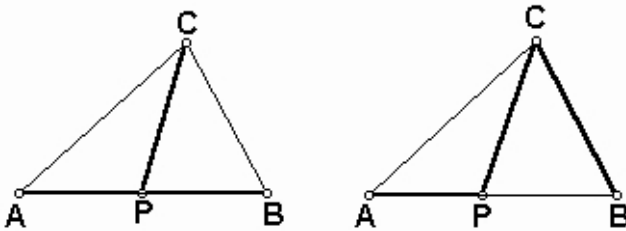


Fig. 7. Two configurations for separable triangles: $AP = PC = PB$, $AP = PC = CB$

While discovering that C lies on a circle centered in P with radius PA for the first configuration (where P is AB midpoint) is trivial, the answer to the second one is not so easy. Note that the locus we are searching for is a case where the locus point imposes conditions on other construction elements, so interactive approaches will generally fail when trying to looking for it. Furthermore, no polynomial-based symbolic approach can deal with the restriction on P lying on segment AB . Using CoCoA as the symbolic engine, Lugares returns an hyperbola and a circle as locus of C (Fig. 8, left). Nevertheless, it is clear that not all points on the hyperbola satisfy the locus requirements. No polynomial equation can express that P is on segment AB , but on line AB . Using the numerical

capabilities of Mathematica, the locus lines are sampled and a member of each sample is tested in order to see if it satisfies all construction properties: the points lying on the hyperbola and exterior to the circle derive from positions of P not in segment AB , hence these samples are rejected from the locus, which is returned as shown in the right of Figure 8.

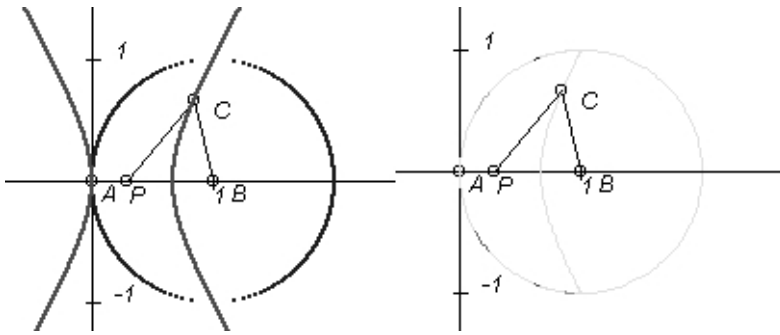


Fig. 8. The locus found with CoCoA (left) and with Mathematica (right)

4 Conclusion

In this paper, interactive and symbolic approaches to the problem of plane loci generation within dynamic geometry environments are compared. It is shown that Groebner bases method can be efficiently used to discover loci, allowing to automate this task in a graphical interface. The ability for finding the equations of loci balances the restriction of the method to algebraic curves. Furthermore, the symbolic approach is more general than the interactive one, in the sense that it allows to search for loci of points constraining other elements. It is also shown how efficient implementations of Groebner bases method compete with interactive approaches when simulating dynamic behavior of loci. First steps in generalizing the symbolic approach to deal with inequations are illustrated by means of a simple problem.

References

1. Botana, F., Valcarce, J. L.: A dynamic-symbolic interface for geometric theorem discovery. *Computers and Education*, to appear (2002)
2. Botana, F., Valcarce, J. L.: Cooperation between a dynamic geometry environment and a computer algebra system for geometric discovery. In V. G. Ganzha, E. W. Mayr, E. V. Vorozhtsov, *Computer Algebra and Scientific Computing CASC 2001*, Springer, Berlin, 63–74 (2001)
3. Botana, F., Valcarce, J. L.: A software tool for investigation of plane loci. Technical Report, University of Vigo, Pontevedra (2001)

4. Buchberger, B.: Groebner Bases: an Algorithmic Method in Polynomial Ideal Theory. In N.K. Bose, *Multidimensional systems theory*, Reidel, Dordrecht, 184-232 (1985)
5. Capani, A., Niesi, G., L. Robbiano: CoCoA, a system for doing Computations in Commutative Algebra. Available via anonymous ftp from: cocoa.dima.unige.it
6. Chou, S. C.: Proving Elementary Geometry Theorems Using Wu's Algorithm. *Automated Theorem Proving: After 25 years*, Contemporary Mathematics, AMS, 29, 243-286 (1984)
7. Gao, X. S., Zhang, J. Z., Chou, S. C.: *Geometry Expert*. Nine Chapters Publ. Taiwan (1998)
8. Guzmán, M.: An extension of the Wallace-Simson theorem: projecting in arbitrary directions. *American Mathematical Monthly*, 106(6), 574-580 (1999)
9. Jackiw, N.: *The Geometer's Sketchpad*. Key Curriculum Press, Berkeley (1997)
10. Kapur, D.: Geometry theorem proving using Hilbert's Nullstellensatz. In *Proc. SYMSAC'86*, Waterloo, 202-208, ACM Press (1986)
11. Kapur, D.: A refutational approach to geometry theorem proving. *Artificial Intelligence*, 37, 61-94 (1988)
12. Kapur, D.: Using Groebner bases to reason about geometry problems. *Journal of Symbolic Computation*, 2, 399-408 (1986)
13. King, J., Schattschneider, D.: *Geometry Turned On*. MAA, Washington (1997)
14. Kortenkamp, U.: Foundations of dynamic geometry. Ph. D. Thesis, ETH, Zurich (1999)
15. Kutzler, B., Stifter, S.: On the application of Buchberger's algorithm to automated geometry theorem proving. *Journal of Symbolic Computation*, 2, 389-397 (1986)
16. Laborde, J. M., Bellemain, F.: *Cabri Geometry II*. Texas Instruments, Dallas (1998)
17. Recio, T., Vélez, M. P.: Automatic discovery of theorems in elementary geometry. *Journal of Automated Reasoning*, 23, 63-82 (1999)
18. Richter-Gebert, J., Kortenkamp, U.: *The Interactive Geometry Software Cinderella*. Springer, Berlin (1999)
19. Roanes-Macías, E., Roanes-Lozano, E.: Búsqueda automática de lugares geométricos. *Proceedings Spanish Educ. Session IMACS-ACA'99, Boletín Sociedad Puig Adam de Prof. de Matemáticas*, 53, 67-77 (1999)
20. Roanes-Macías, E., Roanes-Lozano, E.: Automatic determination of geometric loci. 3D-extension of Simson-Steiner theorem. *Proceedings of Artificial Intelligence and Symbolic Computation 2000, Lecture Notes in Artificial Intelligence*, 1930, 157-173 (2000)
21. Valcarce, J. L., Botana, F.: Lugares. Manual de Referencia. Technical Report, University of Vigo, Pontevedra (2001) <http://rosalia.uvigo.es/sdge/lugares>
22. Wolfram, S.: *The Mathematica book*. Cambridge University Press, Cambridge (1996)
23. Wu, W. T.: *Mechanical Theorem Proving in Geometries*. Springer, Vienna (1994)