A symbolic-numeric dynamic geometry environment for the computation of equidistant curves

Miguel Á. Abánades*[1] and Francisco Botana†[2]

[1]CES Felipe II, Universidad Complutense de Madrid, 28300 Aranjuez, Spain
[2]Dpto. Matemática Aplicada I, Universidad de Vigo, Campus A Xunqueira, 36005 Pontevedra, Spain

Abstract

A web-based system that determines point/curve and curve/curve bisectors in a dynamic geometry system in a completely automatic way is presented. The system consists of an interactive drawing canvas where the bisector is displayed together with the initial point/curve elements. Algebraic methods are used to provide the equation of an algebraic variety containing the bisector. A numeric approach is followed to provide the graph of the semi-algebraic subset corresponding to the true bisector. It is based on the free dynamic geometry system GeoGebra and the open source computer algebra system Sage.

Introduction

A Dynamic Geometry System (DGS) is a computer application that allows the on-screen drawing of (generally) planar geometric diagrams and the manipulation of these diagrams by mouse dragging. The first standard systems to appear (in the late 80's) were *Cabri* in France [10] and *The Geometer's Sketchpad* in USA [9]. Nowadays special mention deserves *GeoGebra* [6], whose free software model and effective community development has resulted in a spectacular world wide distribution.

From the beginning, DGS have been the paradigm of new technologies applied to Math education. However, most DGS rely on numeric computations and approximate graphs, which make them prone to inaccuracies. Moreover, their lack of symbolic tools prevent DGS from realizing a thorough algebraic treatment of geometry.

In this work we develop a symbolic treatment of bisectors in the plane (locus set of points equidistant to two geometrical elements).

In [1], symbolic algorithms to determine algebraic descriptions of point/curve and curve/curve bisectors are described. In this note we present the implementation of these algorithms in an open web-based system in which symbolic capabilities are added to the DGS GeoGebra by connecting it (remotely) to the Computer Algebra System (CAS) Sage [13].

The presented prototype is based on the remote use of Sage rather than GeoGebraCAS [8], the CAS available within GeoGebra, due to higher versatility of Sage, that includes multiple specialized software packages. The prototype wants to illustrate a general philosophy of DGS-CAS connection that we find more appropriate to implement with the most general systems possible. For this same reason, direct use of Python from GeoGebra has not been considered.

Together with these symbolic web applications, numeric graphic alternative tools are provided to visualize a bisector when the exact symbolic computation of its algebraic description is not computationally possible.

1 Bisectors

Given two geometrical elements (points, curves, surfaces, etc.), their bisector is the locus set of points equidistant to them. We consider bisectors of two geometric objects O_1 and O_2 in the Euclidean 2-space E^2, where each object is a point or an algebraic curve (see for instance Figure 1). Bisectors play an important role when constructing Voronoi diagrams, medial axis transformations and in a variety of algorithms related to shape decomposition (see [12]). A systematic study of plane bisectors can be traced back to [4], where the curves are parametrically described, and [7], where a set containing the bisector is obtained by solving a system of nonlinear equations.

Standard DG systems do not consider bisectors. Besides the usual computation of the parabola via its focus and directrix, there are not other primitives for such computations. Nevertheless, the bisector of a point and a linear object in a DG environment can be partially determined through an elementary locus operation. Since a bisector point is the center of a circle

*Email: abanades@ajz.ucm.es. Research partially supported by the project MTM2011-25816-C02-(01,02) funded by the Spanish *Ministerio de Economía y Competitividad*.

†Email: botana@uvigo.es. Research partially supported by the project MTM2011-25816-C02-(01,02) funded by the Spanish *Ministerio de Economía y Competitividad*.

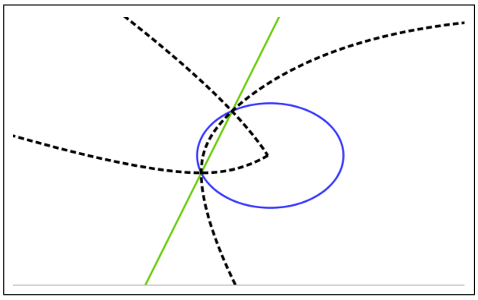

Figure 1: Bisector (dotted) of a line and an ellipse.

tangent to the linear object and passing through the point, the locus tool can suggest a graphical path containing the bisector. Similarly, following the symbolic locus approach in [2], an algebraic variety containing the bisector can be found by solving the corresponding nonlinear polynomial system. We refer to this set as the *algebraic* or *untrimmed* bisector.

In [1], a mixed algebraic-numeric approach for the study of bisectors of points and lines within a DGS is sketched. More precisely, elimination techniques are used for obtaining the detailed description of a bisector. Since the complete bisector description falls out of the algebraic setting, a numerical approach to trim the algebraic bisectors is shown to provide an easy generation of bisectors.

More concretely, the following is the algorithm proposed to compute point/curve bisectors:

Input: curve $c : f(x, y) = 0$, point A
Step 1: Compute (symbolically) the untrimmed (algebraic) bisector of A and c
Step 2: Define this object as an implicit curve d
Step 3: Construct a point B on d
Step 4: Construct the point D on c closest to B
Step 5: If D and A are at the same distance from B, then construct the point E, $E = B$
Return:
i) the locus graphic object *truelocus* of E when B moves along d
ii) the equation(s) of the untrimmed bisector

Moreover, to compute the untrimmed bisector in step 1, a solution based on the remote use of a CAS is indicated.

The system presented here results mainly from the implementation of these algorithms in a web system with a GeoGebra applet through an automatic connection with a remote Sage server.

Determining the algebraic description of some bisectors is computationally out of reach. To obtain the graph of these bisectors we provide an alternative numerical method based on the dynamic color property of GeoGebra whose details can be found in Section 2.2.

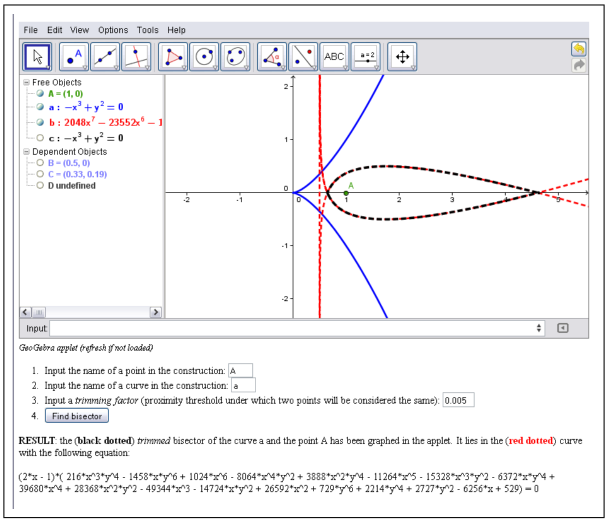

Figure 2: Algebraic bisector (red dotted) and true trimmed bisector (black dotted) of point $A(1, 0)$ and curve $y^2 = x^3$.

2 System Description

The system consists of two main web applications corresponding to the symbolic treatment of point/curve and curve/curve bisectors. Moreover, two auxiliary web applications showing a graphic illustration of a bisector are also provided. They all have been included in a simple web page together with examples and instructions freely available at [15].

All four applications consist of a drawing canvas where the bisectors are displayed together with the initial elements. They all are based on the DGS GeoGebra and the CAS Sage.

GeoGebra is a free DGS with multiple representations of objects in different windows: graphics, algebra, and spreadsheet. Its remarkable world wide use makes GeoGebra a *de facto* standard in the field. Sage is an open source CAS that integrates more than 100 open-source packages (including Singular).

2.1 Symbolic web applications

After the user has input the point and the curve in the applet, he/she just has to press the *Find bisector* button. The aleph.sagemath.org sagecell server [14] is then used to remotely obtain an algebraic variety containing the bisector whose graph is input in the applet. To determine the true (trimmed) bisector, a numeric comparison of distances is carried out. Figure 2 shows how the answer provides both the algebraic description of the bisector together with the true (trimmed) bisector.

The algebraic treatment of the geometric data obtained from the applet is done in Sage with some ad-hoc Python code composed of several hundred lines of

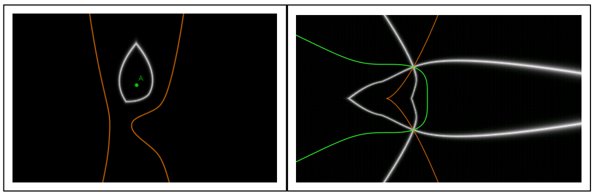

Figure 3: Bisector (white) of point $(0, 2)$ and curve $y^4 + y^2 - y - x^{10} - x^3 - x = 0$ (left) and bisector (white) of curves $y^2 - x^3 = 0$ and $x^3 + y^4 = 9$ (right).

code. More precisely, once the data are sent to Sage, the appropriate variables are initialized and the ideal corresponding to the task is generated. Singular (a CAS included in Sage with special emphasis on commutative algebra) is then called to basically compute a Groebner basis for this ideal.

It has to be noted that the integration of GeoGebra and Sage has been implemented without loosing interactivity. The DGS-CAS communication is synchronous. That is, changing an element in the GeoGebra construction does automatically trigger the corresponding update on the Sage side.

The structure of the symbolic web application for the computation of simple curve/curve bisectors is similar. It implements in a GeoGebra applet the analytic method sketched in [1] (algorithm 1, Section 3).

2.2 Graphic web applications

In the case of bisectors, even point/curve bisectors in simple situations can be very involved. For instance, the bisector of the point $(2, 2)$ and the curve $y = x^7 + 2$ is a polynomial of degree 20 with more than 150 terms.

A curve/curve bisector is symbolically computed as the intersection of two envelopes (see [3]), each one previously obtained with an elimination procedure using Groebner bases. This makes the process too complex for the symbolic computation of bisectors of curves other than lines and circles with current algorithms and standard hardware.

When obtaining the algebraic description of a point/curve or curve/curve bisector is not possible, two web applications providing a graphic illustration of the bisector have been implemented, one for each type of bisector

The idea of these graphic applications is to scan the different 1-pixel points in the applet to change their RGB color code. The color of a point P is changed according to a formula related to the distances to, for instance, curves a and b, in such a way that the closer the value $distance(P, a)$ is to $distance(P, b)$ the *whiter* the point P becomes. Figure 3 shows a point/curve and a curve/curve bisector as graphed by the applications.

This idea, based on the dynamic color property of GeoGebra, was first used by R. Losada [11]. Nevertheless, this approach should be taken with care. Given the numerical character of the application, misleading answers can be returned.

3 Point/Curve Bisectors

In Figure 2 above we have already seen how the answer given by the application provides a complete description of the bisector of a point and a curve, both algebraically and graphically. Here we give a rough description of the method.

If the curve is non-singular, obtaining the algebraic bisector is a direct application of the elementary method for computing bisector points. Each bisector point must lie on the intersection of the normal line to the curve by a generic point on it, and the perpendicular bisector of this point and the given one. Computing the locus of these points we get the algebraic bisector, which will be trimmed in a subsequent step. Note that if the initial point lies on the curve itself, the bisector is contained in the normal line to the curve, as noted in [5]. However, if the curve is singular, the normal line will remain undefined when the generic point is a singular one, thus including a spurious factor (the perpendicular bisector of the singular point and the initial point) in the elimination result. Nevertheless, after the trimming process, all but a finite number of points in this perpendicular bisector will be excluded.

If the initial point is a singular point on the curve, the normal line will be undefined and the perpendicular bisector will be the whole plane, so the process returns the ideal $\langle 0 \rangle$ after elimination. In this case, the trimming procedure does not have a proper variety to trim, since the algebraic bisector is the whole plane. Following [7], we exclude the singular points from our locus finding algorithm. In this way, the application returns some partial information about the bisector. For instance, we have a bi-dimensional bisector for the curve $y^2 = x^3$ and the point $(0, 0)$ as shown by the point/curve graphic application in [15] (Figure 4, left). In this case, the symbolic web application provides the algebraic description of the boundary of this bi-dimensional bisector (Figure 4, right).

As a conclusion, we note that the proposed applications deal efficiently with regular curves, while for singular ones the results, although sometimes clever, must be taken with caution. The final decision about bisectors in such cases should be guided by an ad-hoc and specific study. The automatic determination of the bisectors in these cases is work in progress.

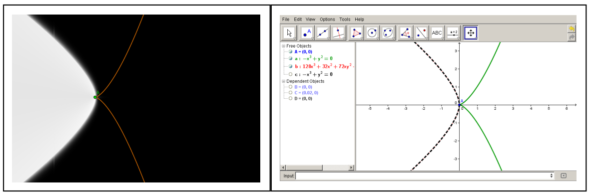

Figure 4: 2D bisector of curve $y^2 = x^3$ and its cusp $(0, 0)$ (left) and its boundary (right).

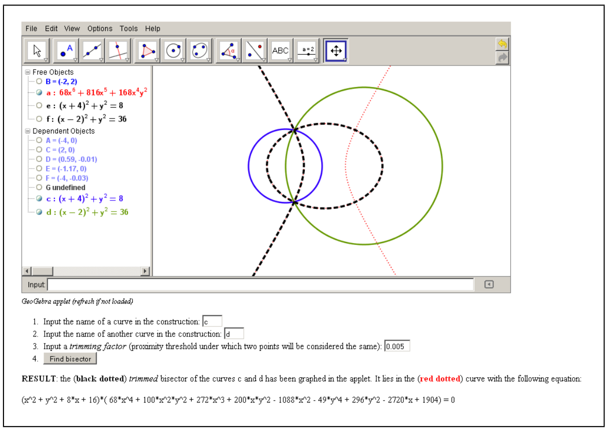

Figure 5: Algebraic bisector (dotted red) and true trimmed bisector (dotted black) of circles $(x + 4)^2 + y^2 = 8$ and $(x - 2)^2 + y^2 = 36$.

4 Curve/Curve Bisectors

As mentioned above, for algebraic curves of low degree the symbolic application for the computation of curve/curve bisectors in [15] provides an algebraic variety where the bisector lies. This algebraic bisector is then trimmed numerically within GeoGebra to display the true bisector.

Figure 5 shows the algebraic bisector and true bisector of the intersecting circles $(x+4)^2+y^2 = 8$ and $(x - 2)^2 + y^2 = 36$ as provided by the prototype.

For curves of higher degree, the computation of the algebraic bisector is (currently) computationally out of reach. In these cases we have already seen in 2.2 how the graphic web application allows the display of curve/curve bisectors for curves of any degree (see Figure 3).

5 Conclusion

The prototype presented provides tools for the study of bisectors in the plane (point/curve and curve/curve). Complete graphic information is provided together with an exact algebraic description when computationally possible.

The system, web-based and interactive, shows the power of the remote automatic connection of CAS and DGS. Moreover, the exclusive use of free software, shows that it can be done without resorting to expensive commercial systems.

References

[1] F. Botana. Computing bisectors in a dynamic geometry environment. *International Journal of Mathematical Education in Science and Technology*, 44(2):299–310, 2012.

[2] F. Botana and J. L. Valcarce. A software tool for the investigation of plane loci. *Mathematics and Computers in Simulation*, 61(2):139–152, 2003.

[3] R. Farouki and J. Johnstone. Computing point/curve and curve/curve bisectors. In R. B. Fisher, editor, *Design and Application of Curves and Surfaces*, number V in The Mathematics of Surfaces, pages 327–354. Oxford University Press, London, 1994.

[4] R. T. Farouki and J. K. Johnstone. The bisector of a point and a plane parametric curve. *Computer Aided Geometric Design*, 11:117–151, 1994.

[5] R. T. Farouki and R. Ramamurthy. Degenerate point/curve and curve/curve bisectors arising in medial axis computations for planar domains with curved boundaries. *Computer Aided Geometric Design*, 15:615–635, 1998.

[6] GeoGebra. `http://www.geogebra.org`, Last accessed May 2013.

[7] C. M. Hoffmann and P. J. Vermeer. Eliminating extraneous solutions in curve and surface operations. *International Journal of Computational Geometry and Applications*, 1:47–66, 1991.

[8] http://dev.geogebra.org/trac/wiki/GeoGebraCAS, Last accessed May 2013.

[9] N. Jackiw. *The Geometer's Sketchpad v 4.0.* Key Curriculum Press, 2002.

[10] J. M. Laborde and F. Bellemain. *Cabri Geometry II.* Texas Instruments, Dallas, 1998.

[11] R. Losada. Propiedad de color dinámico en GeoGebra, `http://geogebra.es/color_dinamico/color_dinamico.html`, Last accessed May 2013.

[12] M. Peternell. Geometric properties of bisector surfaces. *Graphical Models and Image Processing*, 62:202–236, 2000.

[13] Sage. `http://www.sagemath.org`, Last accessed May 2013.

[14] `https://github.com/sagemath/sagecell`, Last accessed May 2013.

[15] `http://webs.uvigo.es/fbotana/bisectors`, Last accessed May 2013.