Available online at www.sciencedirect.com
SCIENCE DIRECT
Mathematics and Computers in Simulation 67 (2004) 3–13
MATHEMATICS AND COMPUTERS IN SIMULATION
www.elsevier.com/locate/matcom

Automatic determination of envelopes and other derived curves within a graphic environment

Francisco Botana [a,*], José L. Valcarce [b]

[a] *Department of Applied Mathematics I, University of Vigo, 36005 Pontevedra, Spain*
[b] *IES Pontepedriña, 15701 Santiago de Compostela, Spain*

Available online 4 July 2004

Abstract

Dynamic geometry programs provide environments where accurate construction of geometric configurations can be done. Nevertheless, intrinsic limitations in their standard development technology mostly produce objects that are equationally unknown and so can not be further used in constructions. In this paper, we pursue the development of a geometric system that uses in the background the symbolic capabilities of two computer algebra systems, CoCoA and Mathematica. The cooperation between the geometric and symbolic modules of the software is illustrated by the computation of plane envelopes and other derived curves. These curves are described both graphically and analytically. Since the equations of these curves are known, the system allows the construction of new elements depending on them.

Keywords: Dynamic geometry; Symbolic computing; Symbolic–numeric interface; Groebner bases; Envelopes; Caustics; Pedals

1. Introduction

Considerable attention and efforts are being given to an emerging field which can be termed *Symbolic–Dynamic Geometry Environments* (SDGE), a synthesis of Dynamic Geometry Software (DGS) and Computer Algebra Systems (CAS). We will not give a definition of CAS, but a few words must be said about the dynamic geometry paradigm. DGS offers virtual environments where accurate construction of geometric configurations can be carried out. The key characteristic of these systems 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 [13,5].

Several authors have postulated a deeper cooperation between both types of systems in order to enhance the abilities of dynamic geometry programs with sophisticated algebraic algorithms. Recio and Vélez

* Corresponding author. Fax: +34 986 801 907.
E-mail addresses: fbotana@uvigo.es (F. Botana), jvalcarce@edu.xunta.es (J.L. Valcarce).

0378-4754/$30.00 © 2004 IMACS. Published by Elsevier B.V. All rights reserved.
doi:10.1016/j.matcom.2004.05.004

[16,17] have extended the automatic theorem-proving method proposed by Kapur [11] to deal with automatic discovery in elementary geometry. They conclude that their method can "be regarded as the core of a future program [· · ·] that allows, when linked simultaneously with a tool for displaying geometric constructions and a symbolic computation package, the interactive exploration of geometric properties". We have developed such a program, Discovery [1], where a dynamic geometry environment written in Prolog cooperates with Mathematica in order to automatize geometric discovery. Another piece of software using the above method is Lugares [2]. Lugares specializes in computing plane loci, and it uses CoCoA [3] in order to override current limitations of Mathematica when dealing with Groebner bases computations. Roanes [20] has also claimed boosting the cooperation betweeen DGS and CAS, having coauthored ParamGeo, a package that links Maple and Derive with The Geometer's Sketchpad (GSP) [10], currently at a beta-level version. Kortenkamp reports in [14] some experiments about integrating its dynamic geometry package Cinderella [19] with other software. Cinderella is used to visualize a simulation driven by a Mathematica notebook, or talks to JavaView [12] using its automatic loci generation to display a surface.

This paper describes our last findings in the development of a symbolic–dynamic geometry environment. While the loci computed by Lugares are special objects in the graphic environment (in the sense that they do not respond to dragging and do not support the construction of new points on them), the actual version of the program deals with such loci as full dynamic objects: the loci are recomputed with no appreciable delay whenever some part of a construction is moved. Furthermore, symbolic algorithms for computing envelopes, pedals and caustics have been added. Both enhancements conclude the development of our SDGE, now called GDI, a Spanish acronym for Intelligent Dynamic Geometry.

The structure of this paper is as follows. Section 2 compares four well-known dynamic geometry environments and GDI when computing a single envelope, the deltoid of Steiner. We recall the basic mathematical machinery necessary for the computation of plane envelopes and we illustrate the developed algorithms for some curve families in Section 3. Section 4 describes how caustics and pedals can also be computed in the system.

2. The deltoid of Steiner as an envelope of the Simson–Wallace lines

In an imprecise but descriptive language, the envelope of a curve family $F(x, y, \alpha) = 0$ consists of those points which belong to each pair of infinitely near curves in the family. Most DGS packages support graphical computation of envelopes. GSP and Geometry Expert (GEX) [6] trace a given curve of the family, thus suggesting in some cases the envelope. Cabri [15] and Cinderella employ a more sophisticated approach since they usually return the envelope as a single curve, but this line is merely a graphic object on the screen. That is, no analytic knowledge about the envelope is available, and it is a final object in the sense that it cannot be used to construct further objects.

Fig. 1 shows the behavior of these four systems when computing a single envelope, the deltoid of Steiner. Recall that the Simson–Wallace theorem states that given a triangle *ABC* and a point *X* on its circumcircle, the feet of the projections of *X* on the sides of triangle *ABC* are collinear. The envelope of this line family, when *X* moves along the circumcircle, is the deltoid of Steiner.

At first sight, the computation of this envelope in GDI shows a similar curve to those of Cabri or Cinderella. Nevertheless, there is a significant difference: a user can get the implicit equation of the curve (Fig. 2), thus is able to place points on it, which constitutes a prerequisite for any construction using the deltoid.

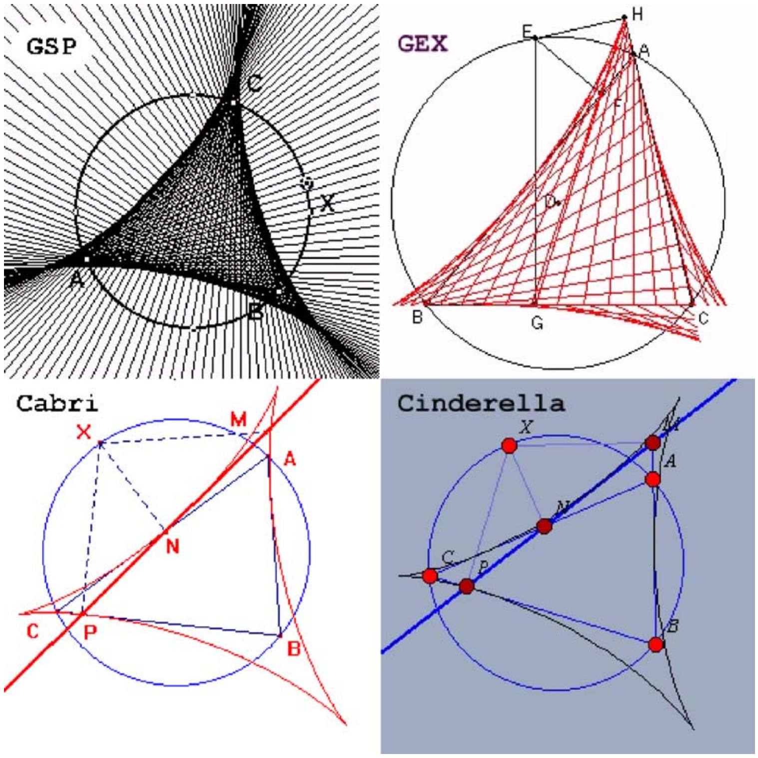

Fig. 1. The Steiner's deltoid in four DGE.

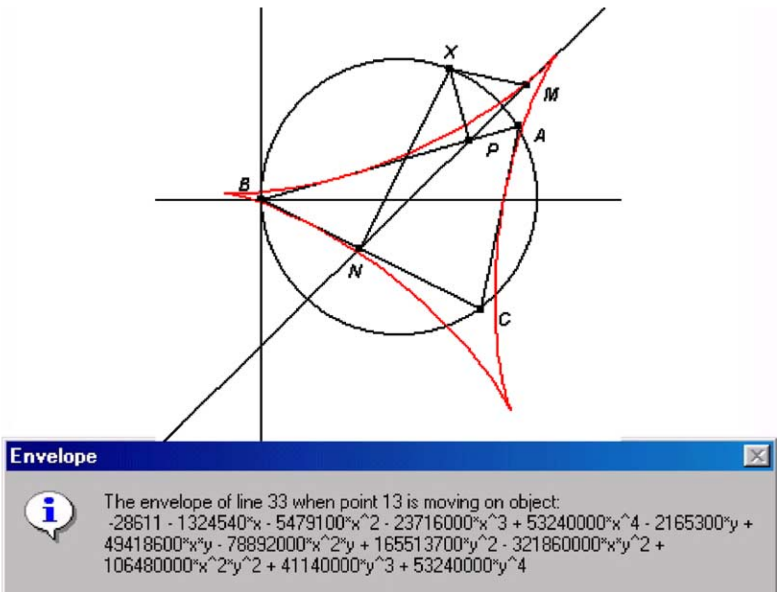

Fig. 2. The equation and plot of a Steiner's deltoid in GDI.

3. The computation of envelopes in GDI

3.1. The deltoid of Steiner

The construction in Fig. 2 is straightforward. Three points $A(.7, .2)$, $B(0, 0)$ and $C(0.6, -0.3)$ define a triangle and the midpoints $A_7(x_1, x_2)$, $A_9(x_3, x_4)$ of the sides BC and AC are also defined. The intersection of the perpendicular bisectors passing through A_7 and A_9 is the circumcenter $O(x_5, x_6)$, and a point $X(x_7, x_8)$ on the circumcircle is also constructed. Finally, the projections $M(x_9, x_{10})$, $N(x_{11}, x_{12})$, $P(x_{13}, x_{14})$ of X on the sides of the triangle are also defined. Note that some elements in the construction are hidden for clarity. Thus, this part of the construction can be linguistically specified as follows:

A_7 Midpoint(B,C),
A_9 Midpoint(A,C),
l_1 Perpendicular(A_7,BC),
l_2 Perpendicular(A_9,AC),
O Intersection(l_1,l_2),
c_1 CircleCenterPoint(O,B),
X PointOnCircle(c_1),
l_3 Perpendicular(X,AC),
M Intersection(l_3,AC),
l_4 Perpendicular(X,BC),
N Intersection(l_4,BC),
l_5 Perpendicular(X,AB),
P Intersection(l_5,AB),

and the coordinates of the constructed points are determined by

$$A_7 : -\left(\frac{3}{10}\right) + x_1 = 0, \quad \frac{3}{20} + x_2 = 0,$$

$$A_9 : -\left(\frac{13}{20}\right) + x_3 = 0, \quad \frac{1}{20} + x_4 = 0,$$

$$O : \frac{3\,(-x_1 + x_5)}{5} - \frac{3\,(-x_2 + x_6)}{10} = 0, \quad \frac{x_3 - x_5}{10} + \frac{x_4 - x_6}{2} = 0,$$

$$X : -x_5^2 - x_6^2 + (x_5 - x_7)^2 + (x_6 - x_8)^2 = 0,$$

$$M : \frac{7/10 - x_9}{2} + \frac{-(1/5) + x_{10}}{10} = 0, \quad \frac{x_7 - x_9}{10} + \frac{x_8 - x_{10}}{2} = 0,$$

$$N : \frac{-3\,x_{11}}{10} - \frac{3\,x_{12}}{5} = 0, \quad \frac{3\,(-x_7 + x_{11})}{5} - \frac{3\,(-x_8 + x_{12})}{10} = 0,$$

$$P : \frac{7/10 - x_{13}}{5} - \frac{7\,(1/5 - x_{14})}{10} = 0, \quad \frac{-7\,(-x_7 + x_{13})}{10} + \frac{x_8 - x_{14}}{5} = 0.$$

If the user defines a Simson–Wallace line passsing through M and P, the equation of which is

$$(-y+x_{10})\,(-x_9+x_{13})+(x-x_9)\,(-x_{10}+x_{14})=0,$$

and the user selects this line, which is a member of the curve family, and the point X in order to describe the whole family, the algorithm to compute the envelope is launched. Note that when X and A are the same point, so are M and P, and the line collapses to a point. This fact must be avoided by the user by declaring that points X and A can not coincide through the program option `Not Equal`.

Recall that when a given α is replaced by $\alpha+\Delta\alpha$ in the equation of the curve family $F(x, y, \alpha)=0$, we obtain a new curve implicitly defined by $F(x, y, \alpha+\Delta\alpha)=0$. The set of points that belong to both curves satisfies

$$F(x, y, \alpha+\Delta\alpha)-F(x, y, \alpha)=0.$$

Suppose that F is differentiable with respect to α; if this variable α can be eliminated from the system

$$F(x, y, \alpha)=0$$
$$\frac{\partial F(x, y, \alpha)}{\partial\alpha}=0,$$

the resulting curve implicitly defined by $f(x, y)=0$ is called the envelope of the curve family $F(x, y, \alpha)=0$.

If the curve family is biparametrically defined by $F(x, y, \alpha, \beta)=0$, where the parameters are related by $g(\alpha, \beta)=0$, the elimination of α and β takes place in the system

$$F(x, y, \alpha, \beta)=0$$
$$g(\alpha, \beta)=0$$
$$\frac{\partial F(x, y, \alpha, \beta)}{\partial\alpha}\frac{\partial g(\alpha, \beta)}{\partial\beta}-\frac{\partial F(x, y, \alpha, \beta)}{\partial\beta}\frac{\partial g(\alpha, \beta)}{\partial\alpha}=0.$$

where F and g are assumed to be differentiable.

In the case we are dealing with, the equations defining the curve family and involving the parameters

$$(-y+x_{10})(-x_9+x_{13})+(x-x_9)(-x_{10}+x_{14})=0,$$
$$-x_5^2-x_6^2+(x_5-x_7)^2+(x_6-x_8)^2=0,$$

depend on extra variables, $x_5, x_6, x_9, x_{10}, x_{13}, x_{14}$. Since these variables can be expressed in terms of x_7, x_8, the Mathematica command `Solve`, or in case of no success, `Reduce`, return a list of rules

$$x_5\to\frac{83}{220},$$
$$x_6\to\frac{1}{220},$$
$$x_9\to\frac{33+2\,x_7+10\,x_8}{52},$$

$$x_{10} \to \frac{-33 + 50\,x_7 + 250\,x_8}{260},$$

$$x_{13} \to \frac{7\;(7\,x_7 + 2\,x_8)}{53},$$

$$x_{14} \to \frac{2\;(7\,x_7 + 2\,x_8)}{53},$$

which are used, also renaming x_7, x_8 to α, β, to rewrite the system as

$$-119\,\alpha + 70\,\alpha^2 - 34\,\beta + 370\,\alpha\,\beta + 100\,\beta^2 + 53\,x + 30\,\alpha\,x - 370\,\beta\,x + 265\,y - 370\,\alpha\,y - 30\,\beta\,y = 0,$$

$$-66754\,\alpha + 85550\,\alpha^2 + 6217\,\beta + 85550\,\beta^2 = 0.$$

Both equations, together with the corresponding differential equation for biparametric curve families, are passed to CoCoA, where variables α, β are eliminated via Groebner bases. The reason for using CoCoA is its superior performance for polynomial elimination when compared with Mathematica. The responses of CoCoA were at least ten times faster than those of Mathematica in some empiric tests that we have conducted. Furthermore, there are cases where Mathematica could not eliminate the parameters in a reasonable amount of time (1 h), while CoCoA succeeded in seconds. So our strategy consists of using CoCoA whenever possible, and giving Mathematica a chance if there are non-polynomial equations in the system.

The equation resulting from the elimination is passed to Mathematica, where a graphic object is generated via the package `Graphics‘ImplicitPlot‘`. Since some equations can be extremely complicated and hard to graph in the interval of time indicated by the user, two methods are used. First, the `ContourPlot` method treats the equation as a function in three--dimensional space, and generates a contour of the equation cutting through the plane where z equals zero. This method handles a great variety of cases, but may generate rough graphs, especially around singularities or intersections of the curve. If there is time left, the second method uses `Solve` to find solutions to the equation at each point in the x range and generates a smoother graph. This last graphic object, if it can be computed, or the first one, is returned to GDI as a list of points which are scaled to the current window size. The deltoid of Steiner obtained by the `ContourPlot` method is shown in Fig. 3, while the other method returns the plot in Fig. 2.

3.2. The envelope of Giering–Guzmán lines

our previous paper on this topic [2], we showed how a recent generalization of Simson–Steiner theorems due to de Guzmán [9] could be easily computed with the old version of our SDGE. This extension states that given a triangle *ABC* and three directions, not all three equal nor parallel to the triangle sides, the locus of points X such that their projections M, N, P along the three directions determine a triangle of oriented area k, is a conic. Besides this theorem, de Guzmán has studied the deltoid of Steiner [8], giving a simpler method than the original one of Steiner [21] to prove that it is a tricuspidal hypocycloid. Surprisingly, as far as we know de Guzmán did not consider connecting both works to study the envelope of lines passing through M, N, P (that is, when $k = 0$).

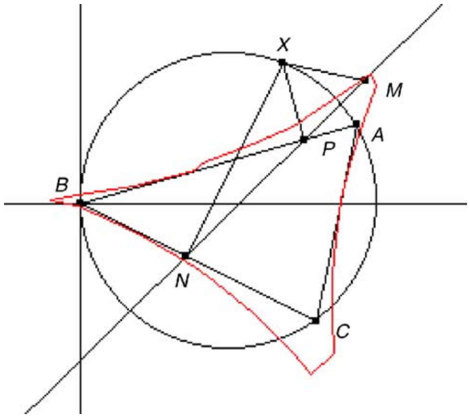

Fig. 3. A rough plot of the deltoid.

A major difference between GDI and other DGEs consists of the ability of the former to compute the locus of a point moving along a path whose parametric equations are unknown [18]. This case is perfectly illustrated with the above mentioned extension of Simson–Steiner theorems. In an analogous manner to that of the preceding subsection, once the conic of de Guzmán is obtained for $k = 0$, a point X on the conic is constructed and also the feet of two projections from X to the triangle sides, M and N for instance. The envelope of the family of lines MN when X moves along the conic is shown in Figs. 4 and 5 for an ellipse ($18128x^2 + 22290xy + 46077y^2 - 18128x - 47706y = 0$) and a hyperbola ($33232x^2 + 71460xy - 9367y^2 - 33232x - 33805y = 0$).

Since GDI can work without a rectangular coordinate system, we tried to find the equation of this envelope in the general case. Given an arbitrary triangle ABC, with vertices $A(0, 0)$, $B(1, 0)$, $C(u_1, u_2)$ (there is no lost of generality in this assumption), the locus is a conic described by 154 terms. Not having

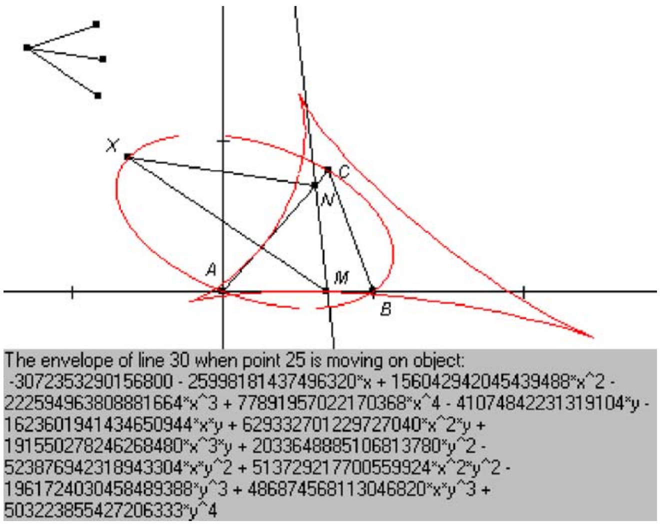

Fig. 4. An ellipse and the envelope of their Giering–Guzmán lines.

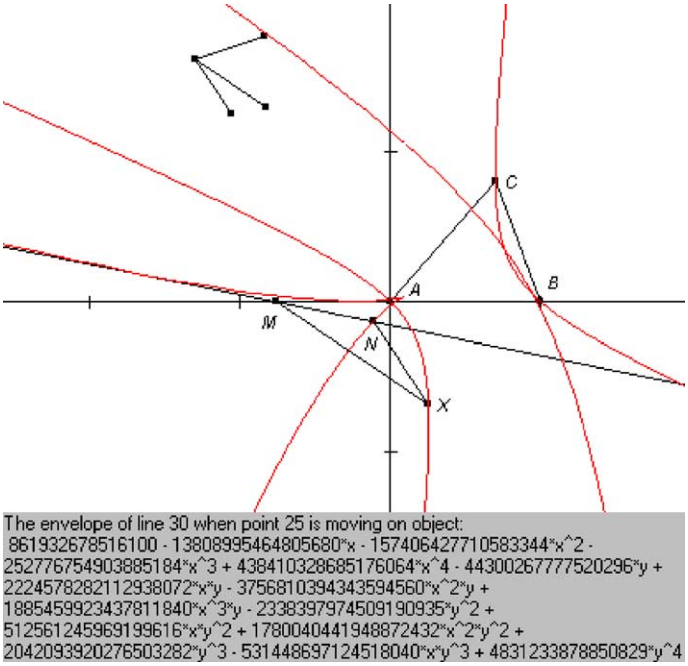

Fig. 5. A hyperbola and the envelope of their Giering–Guzmán lines.

the graph of the locus, we could not continue using GDI. A point on this conic and the feet of two projections were considered in order to get the polynomial system for elimination. Nevertheless, CoCoA could not eliminate variables after a couple of days. Another attempt to solve the problem with resultants was also unfruitful [4]. Finally, we found that the problem had been solved some years ago. Using affine and projective geometries, Giering [7] proved that the envelope is a Steiner's hypocycloid, a rational curve of fourth order and third class, which is tangent to the sides of the triangle ABC.

3.3. The offset curve of a parabola

Let us consider a circle centered at a point moving along a parabola. The envelope of this family of circles is known as an offset curve of the parabola (Fig. 6). Graphing this envelope is a difficult task in most DGEs or CAD systems. Furthermore, its equation remains unknown in these systems, while it is easily obtained by GDI.

4. Computation of caustics and pedals

4.1. Caustics by reflection

Given a light ray from a point X to a point P moving along a simple curve C, the envelope of the reflected rays is the catacaustic of C relative to X. The symmetrical point Y of X with respect to the

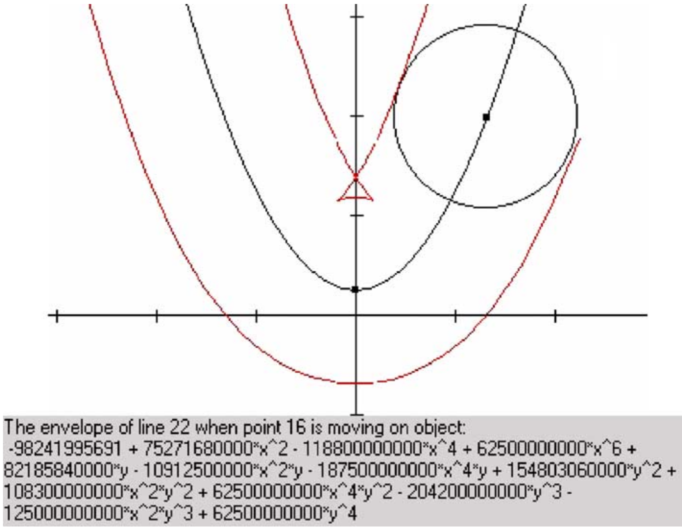

Fig. 6. The offset curve of a parabola.

normal line of the curve C at P can be used to define the family of lines PY. The envelope of these lines is the catacaustic.

It is usually difficult to trace the tangent or normal lines at a point of a general simple curve in a purely geometric context. Thus, standard DGE approaches to caustic generation rely heavily on this prerequisite knowledge. If the curve is as simple as a circle, the reflected rays are easily obtained due to the coincidence between the radius and the normal line. Nevertheless, tracing the normal line at a point on a general conic with ruler and compass is a hard task for a non-expert user.

Since the equation of any constructible curve in GDI is known, the operation of computing normal lines is an elementary one calling Mathematica. GDI computes caustics by selecting a simple curve C and a point X. The normal line of C at an internally constructed point P is used to define the symmetrical point Y of X with respect to this line, and finally the algorithm for finding the envelope of the line family PY is launched. Fig. 7 shows the caustics found by GDI given the ellipse $3x^2 + 4y^2 = 12$ (obtained as the locus of the points whose distances to $(1, 0)$ and $(-1, 0)$ sum to 2) for three positions of the light source. Their equations are a quartic if the source is on the conic, and two sextics in the other cases.

4.2. Pedals

Recall that a pedal curve is the locus of the feet of perpendiculars that fall from a fixed point upon the tangent lines to a given curve. Furthermore, the envelope of perpendiculars to lines joining the fixed point and a point moving along the curve and passing the point on the curve is called the negative pedal of the curve with respect to the fixed point.

Combining the call to the CAS for computing normal lines with the capabilities of the system for obtaining loci of points and envelopes of curve families, both positive and negative pedals can be obtained. The pedal of the parabola $2y = x^2$ with respect to its vertex is a cissoid of Diocles, and the negative pedal of this cissoid with respect to the same point is the original parabola (Fig. 8).

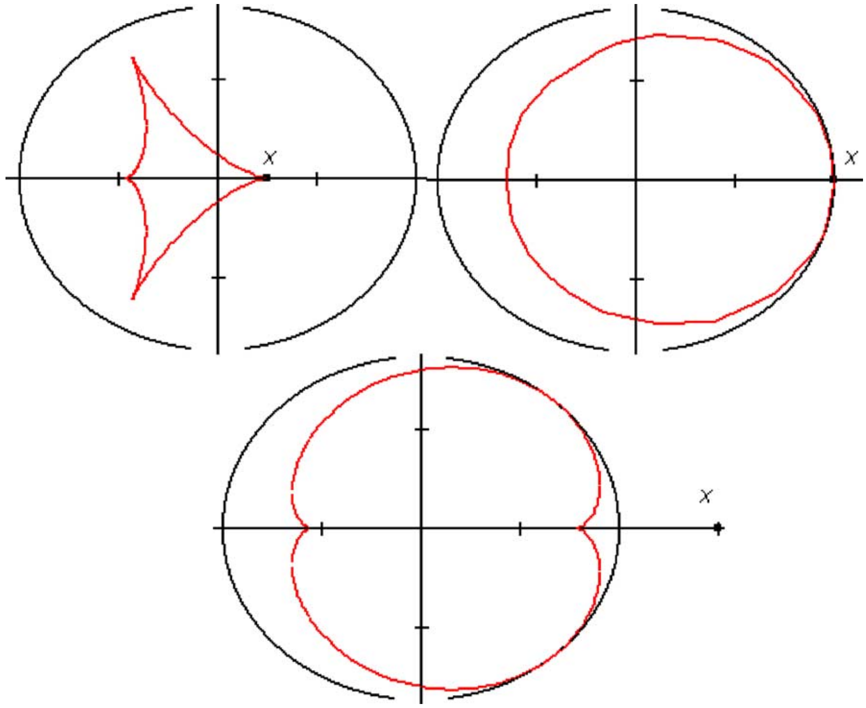

Fig. 7. Caustics of the ellipse $3x^2 + 4y^2 = 12$ relative to $X(.5, 0)$, $X(2, 0)$ and $X(3, 0)$.

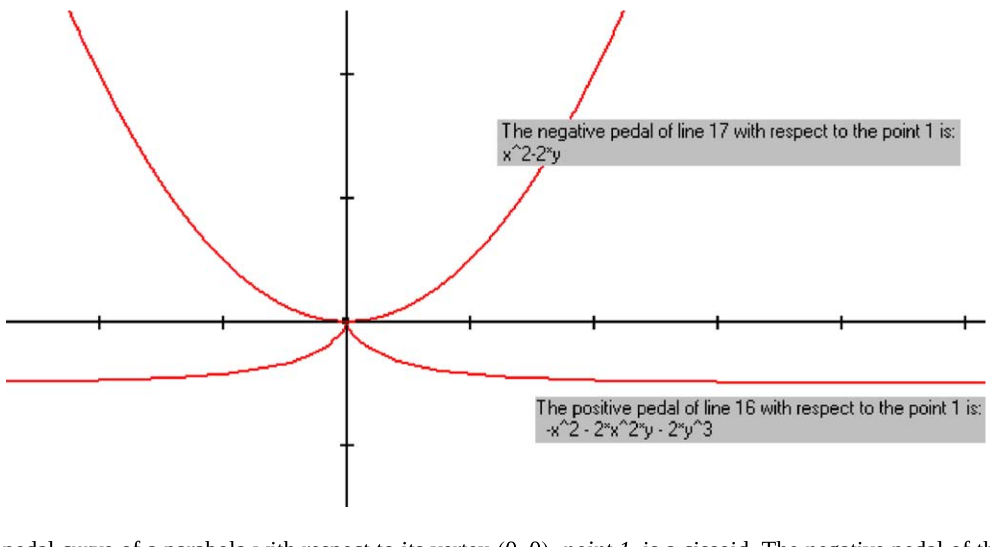

Fig. 8. The pedal curve of a parabola with respect to its vertex $(0, 0)$, *point 1*, is a cissoid. The negative pedal of the cissoid is the original parabola.

5. Summary

We described how envelopes of family curves and other derived curves can be automatically computed within a graphical interface. We found that the implementation of the algorithm of Buchberger in CoCoA is efficient for common elimination tasks in the domain of plane geometry with numeric coordinates. Nevertheless, the complexity of some problems involving more than a few variables exceeds the capabilities of current elimination algorithms.

In this paper, we complete the description of GDI, a dynamic geometry environment which uses the symbolic abilities of two computer algebra systems, CoCoA and Mathematica. The system may be downloaded (for academic purposes) from José L. Valcarce by writing to the given e-mail address.

References

[1] F. Botana, J.L. Valcarce, A dynamic-symbolic interface for geometric theorem discovery, Comput. Educ. 38 (1/3) (2002) 21–35.
[2] F. Botana, J.L. Valcarce, A software tool for the investigation of plane loci, Math. Comput. Simul. 61 (2) (2003) 141–154.
[3] A. Capani, G. Niesi, L. Robbiano, CoCoA, a system for doing computations in commutative algebra. Available via anonymous ftp from: `cocoa.dima.unige.it`.
[4] A. Chtcherba, Personal communication, 2001.
[5] X.S. Gao, Automated geometry diagram construction and engineering geometry, in: Proceedings of ADG'98, Lecture Notes in Artificial Intelligence, vol. 1669, Springer, Berlin, 1998, pp. 232–257.
[6] X.S. Gao, J.Z. Zhang, S.C. Chou, Geometry Expert, Nine Chapters Publ., Taiwan, 1998.
[7] O. Giering, Affine and projective generalization of Wallace lines, J. Geom. Graph. 1 (2) (1997) 119–133.
[8] M. de Guzmán, The envelope of the Wallace–Simson lines of a triangle, http://www.mat.ucm.es/deptos/am/guzman/deltoide121298/00delten.htm.
[9] M. de Guzmán, An extension of the Wallace–Simson theorem: projecting in arbitrary directions, Am. Math. Mon. 106 (6) (1999) 574–580.
[10] N. Jackiw, The Geometer's Sketchpad, Key Curriculum Press, Berkeley, 1997.
[11] D. Kapur, A refutational approach to geometry theorem proving, Artif. Intell. 37 (1988) 61–94.
[12] JavaView, http://www.javaview.de.
[13] J. King, D. Schattschneider, Geometry Turned On, Mathematical Association of America, Washington, DC, 1997.
[14] U. Kortenkamp, Integrating CAS with Cinderella, Dagstuhl–Seminar No. 01421, Integration of Algebra and Geometry Software Systems, 2001.
[15] J.M. Laborde, F. Bellemain, Cabri Geometry II, Texas Instruments, Dallas, 1998.
[16] T. Recio, Cálculo simbólico y geométrico, Síntesis, Madrid, 1998 (in Spanish).
[17] T. Recio, M.P. Vélez, Automatic discovery of theorems in elementary geometry, J. Autom. Reasoning 23 (1999) 63–82.
[18] T. Recio, Personal communication, 2002.
[19] J. Richter-Gebert, U. Kortenkamp, The Interactive Geometry Software Cinderella, Springer, Berlin, 1999.
[20] E. Roanes–Lozano, Boosting the geometrical possibilities of dynamic geometry systems and computer algebra systems through cooperation, in: M. Borovcnik, H. Kautschitsch (Eds.), Technology in Mathematics Teaching, öbv & hpt, Vienna, 2002, pp. 335–348.
[21] J. Steiner, Gesammelte Werke, Band II (Herausgegeben von K. Weierstrass), G. Reimer, Berlin, 1882, pp. 639–647.