

Extending envelope computations in Computer Algebra/Dynamic Geometry environments

Agustín Carrillo de Albornoz
e-mail: agustincarrillo@fespm.es
Instituto GeoGebra de Andalucía
Andújar, Jaén, Spain

Tomas Recio
e-mail: trecio@nebrija.es
Universidad Antonio de Nebrija
Madrid, Spain

Abstract

In this note we present and exemplify a new protocol for envelope computation, specifically conceived for its implementation in programs, such as GeoGebra, that combine Computer Algebra and Dynamic Geometry features. It extends, in different ways, the usual envelope concept, allowing, in particular, the computation of several kind of envelopes in 3D. The motivation, antecedents and advantages of this approach in an educational context are highlighted.

1. Introduction

Envelope computation is a traditional topic, present in the curriculum of different scientific careers for its ubiquitous emergence as a tool to model many situations (eg. robot motion planning, geometric optics, Computer Aided Design, Economics, etc.), and, of course, as a basic concept in Differential Geometry with connections to other mathematics fields, such as singularity theory, algebraic geometry, matrix theory, hyperbolic geometry, etc. We remit, for example, to the classical survey by Pottmann-Peternell [18] or to the quite recent one, on applications, by Bicker-Gorkin-Tran [2], and to the references therein. As a consequence, envelopes are particularly well suited for a STEAM (Science, Technology, Engineering, Art, Mathematics)-focused education.

The concept of envelope merges several characteristics that make it especially attractive for an ample spectrum of researchers in mathematics education, in computational mathematics, etc. For example, envelope's definition involves just elementary notions, yet it is a manifold concept, and it is still an open task to establish precise relations among them. Thus [7] –probably, see [8]—the most reputed authority on the subject, describes in detail the four classical definitions: the *geometric envelope*, as tangent to every curve of the given family $F(t, x, y)=0$; the *limiting-position envelope*, determined by the intersection of close curves in the family; the *discriminant envelope*, obtained by eliminating the parameter t in the system $\{F(t, x, y)=0, F_t(t, x, y)=0\}$; and the *boundary envelope* (the boundary of the two-dimensional set of the plane described by all the curves in the family). But, while different works (eg. [8], [4], [2]) mention several results describing relations among these definitions and stating conditions for their coincidence, they also refer to the still unsolved difficulties for establishing a general theorem.

Thus, we should remark that, in what follows, we will focus, as in [7]¹, in the definition of envelope as a *discriminant*.

¹ That seems to be mostly the inspiration for [https://en.wikipedia.org/wiki/Envelope_\(mathematics\)](https://en.wikipedia.org/wiki/Envelope_(mathematics)) and other popular references for this subject.

A second reason supporting the presence of envelopes in the current scientific literature is the opportunity their study brings to foster the cooperation of algebra, geometry and technological tools in the classroom, via CAS (computer algebra systems) and DGS (dynamic geometry systems) programs, now quite widely and freely available, for example, through the dynamic mathematics program GeoGebra (www.geogebra.org), [10]. Certainly, DGS provide a relevant opportunity to visualize envelopes, while CAS make possible to consider a rigorous approach for computing them.

Let us recall that envelopes are a particular case of locus computation; and locus computation is recognized [1] as one of the five basic characteristics in the DGS paradigm, see also [3]. Yet, in [4], [5] a survey was detailed on the performance of different DGS (Cabri, Cinderella, The Geometer's Sketchpad, GeoGebra) for envelope computation, showing that all these programs presented notable limitations, in a double sense. One, regarding the kind of curves (mostly lines, circles, ...) that could be involved, for each value of t , in the input family $F(t, x, y)=0$, or that could be analytically recognized as the computed envelope. Two –and more relevant from a mathematical perspective—concerning the implemented methods, as they are mainly graphic. For example, computing envelopes by merely tracing a large sequence of curves of the given family, and visually displaying the intersection of nearby curves, etc. Obviously, this graphic approach is not well suited to deal with the most relevant concept of envelope as a discriminant, as it requires algebraic computations.

Thus, improving computation of envelopes in DGS contexts is a quite nuclear issue, that is currently attracting the attention of many researchers. See, for example, some previous work by one of the authors [4], [5], [6] or the large collection of contributions by prof. Dana-Picard and colleagues, both concerning the theoretical framework, as in [13], [14], [15], or, developing some concrete computations in particular instances [9], [11], [12].

Our approach collects some of our previous, disperse, work; focuses on the case of algebraic, implicit families in 2D and 3D, described with several parameters, both when they are given independently or constrained by some equations. The merging of all these characteristics in a single paper, plus the exemplification and analysis through GeoGebra of the different proposals, profiting from the interaction CAS/DGS that this program allows, is, perhaps, the most valuable contribution of this note.

2. Extending envelope computation in GeoGebra

To start with, working with envelopes in DGS requires first, in our opinion, to reconsider the classical approach and to extend the concept of envelope so that it includes some cases that arise very naturally in the DGS constructions. For example, the admissible input for envelope computations should go beyond the traditional restriction to uni-parametric families $F(t, x, y)=0$ and accept families of curves $F(a, b, x, y)=0$, depending on (a, b) , the coordinates of a *mover point*, ie. a point moving on another curve $G(a, b)=0$.

This is already possible with GeoGebra, after version 5, via the *Envelope*(*<Path>*, *<Point>*) command, where *<Point>* refers to the mover point, and *<Path>* points out to the curve built after performing different steps involving the mover. See Figure 1, adapted from the tutorial [16], displaying the envelope *eq2:Envelope(f, C)*, an ellipse, of the family of lines f (the output *Path*) perpendicular to $g=segment(CD)$, for a fixed point D and a point C moving on the circle c centered at A and passing through B .

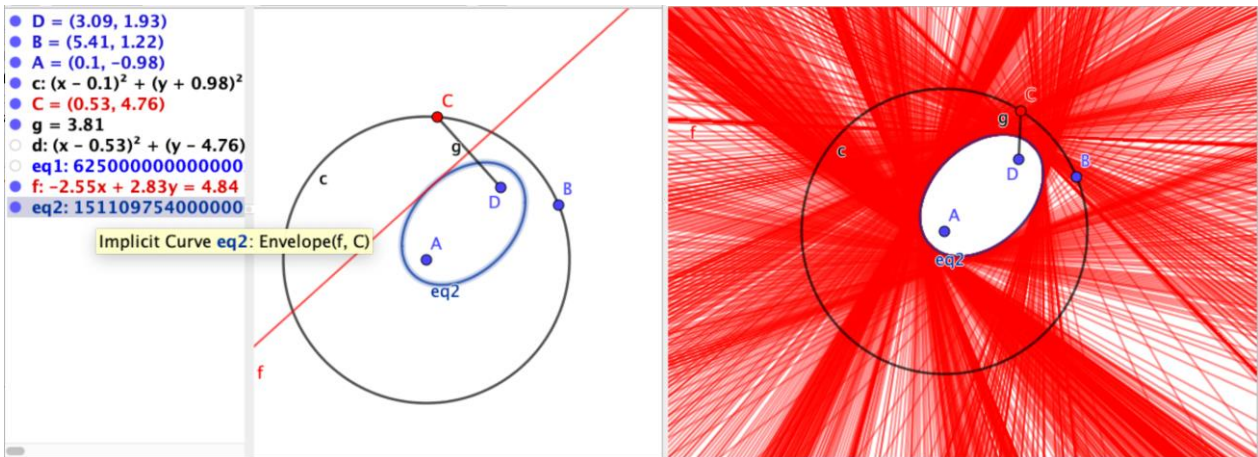


Figure 1: GeoGebra computation of the envelope of a family of lines f perpendicular to CD , when C moves on a circle c centered at A and going through B , and D is a fixed point.

As proposed in [4], the internal algorithm that GeoGebra applies for computing such envelopes proceeds by eliminating the variables (a, b) in the system $\{F(a, b, x, y)=0, \text{Jacobian}(\text{matrix}(\{\partial F/\partial a, \partial F/\partial b\}, \{\partial G/\partial a, \partial G/\partial b\})) = \partial F/\partial a * \partial G/\partial b - \partial F/\partial b * \partial G/\partial a = 0\}$. For example, assume the user takes a point $C(a, b)$ in the circle $G(a, b) = a^2 + b^2 - 4 = 0$, a fixed point $D(3/2, 0)$, and finally, considers the family of perpendicular lines to the segment CD passing through point C , $F := (3/2 - a)(x - a) + (0 - b)(y - b) = 0$. Then, GeoGebra computes the Jacobian of the matrix $\{\{\partial F/\partial a, \partial F/\partial b\}, \{\partial G/\partial a, \partial G/\partial b\}\}$, yielding the determinant $\text{Jacobian} = 2ay - 2bx - 3b$ and performs the elimination of all variables except $\{x, y\}$ in the ideal $\langle F, G, \text{Jacobian} \rangle$, obtaining as envelope the ellipse $7x^2 + 16y^2 - 28 = 0$.

As a second example, replicating manually these internal computations, Figure 2 shows, to the left, the CAS window of GeoGebra, introducing, in row 1, the equation of a circle of radius 1 centered at a point $A = (a, b)$ on the parabola $b = a^2$. Then the third row computes the Jacobian matrix, with determinant $4ab - 4ay + 2a - 2x$. Finally, row 5 does the elimination of the ideal of polynomials appearing in rows 1, 2, and 4; and row 6 shows the output, the envelope, a curve of degree 6 (in blue in the Figure).

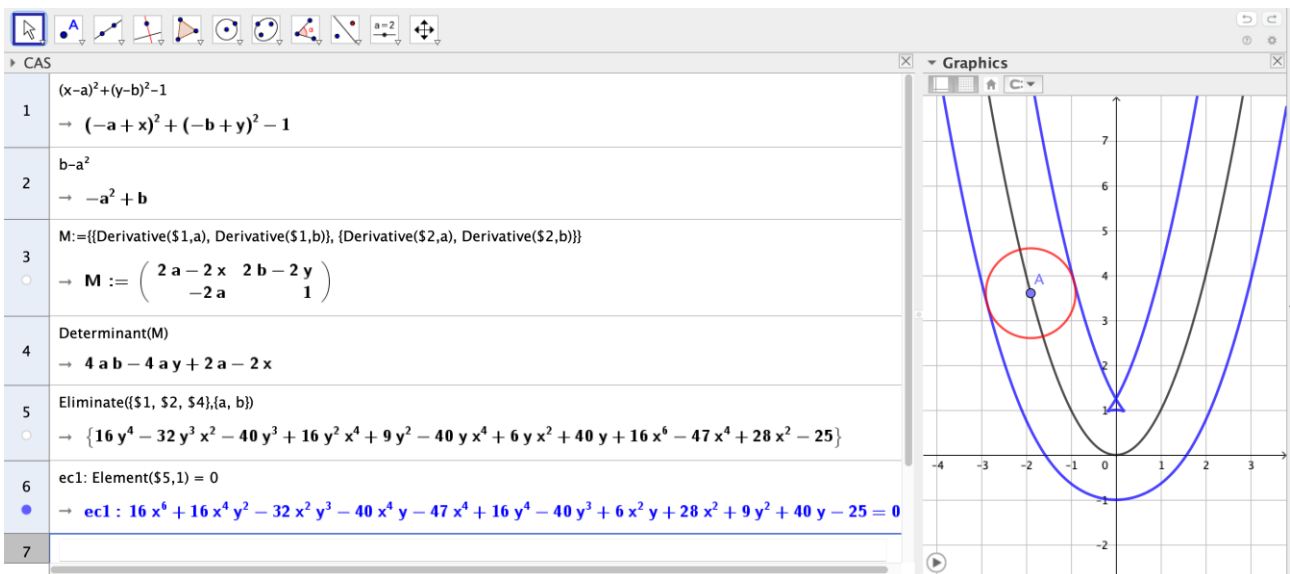


Figure 2: The envelope of a family of circles of radius 1 centered at a point $A = (a, b)$ moving on a parabola.

Two remarks are needed at this point. One is to recall that the *Eliminate* command in GeoGebra, and the associated mathematical concept, deals only with polynomials. Of course it can be simulated by using some alternative approaches, such as solving $\{F=0, G=0\}$ for a and b in terms of x, y and substituting the obtained solutions in the values of a and b in the Jacobian, yielding an equation in $\{x,y\}$ only. But, again, there is not a computational sound definition of “solving” except for the case of polynomial equations, and it is related to the concept of elimination!²

The second remark we would like to point out is about this particular way of finding the envelope, as detailed in Figure 2, cannot be always replaced by using the standard *Envelope* command of GeoGebra. For instance, in this case this is so since the parabola f and the circle c have not been introduced through GeoGebra constructions, but as implicitly given curves. See Figure 3 that shows, to the left, the envelope computed using GeoGebra’s CAS in the previous Figure 2. It cannot be obtained using the *Envelope(c, A)* command, that returns *undefined*. To the right, the same envelope but this time obtained by GeoGebra after introducing circle d and the parabola c through some construction commands (eg. the parabola with focus A and directrix f , and circle centered at D with radius $g=\text{distance}(BC)$).

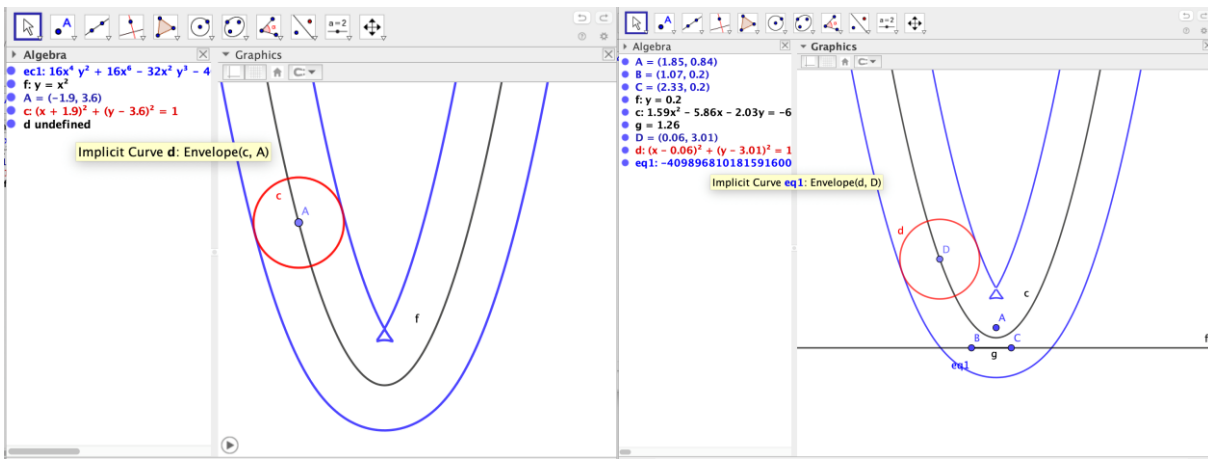


Figure 3: To the left, the envelope *Envelope(c, A)* command returns *undefined*. To the right, the same envelope obtained by GeoGebra after introducing the involved elements through some construction commands.

As initially remarked in [4], and developed in [5] and [6], the computational approach we have described can be easily extended to consider families of curves depending on many parameters, $F(a_1, a_2, \dots, a_n, x, y)=0$, as long as the parameters are constrained by $n-1$ independent conditions $\{G_1(a_1, a_2, \dots, a_n)=0, \dots, G_{n-1}(a_1, a_2, \dots, a_n)=0\}$, computing the Jacobian $Jacobian(matrix(\{\partial F/\partial a_1, \dots, \partial F/\partial a_n\}, \{\partial G_1/\partial a_1, \dots, \partial G_1/\partial a_n\}, \dots, \{\partial G_{n-1}/\partial a_1, \dots, \partial G_{n-1}/\partial a_n\}))$ and eliminating all variables except $\{x,y\}$ in the ideal generated by all these polynomials F, G 's and the *Jacobian*, or, geometrically and roughly speaking, projecting the associated 1-dimensional algebraic set $\{F=0, G_1=0, \dots, G_{n-1}=0, Jacobian=0\}$ over the $\{x, y\}$ plane.

This opens a series of delicate issues about the best algorithm to achieve more accurately such projection, such as using Gröbner basis, the current standard option in GeoGebra; or Gröbner Cover [17], also available over GeoGebra 5, but outsourcing the Gröbner Cover computation to the CAS program Singular (<https://www.singular.uni-kl.de>). Indeed, in our opinion, at this moment the more general and sophisticated approach to envelope computation --that carefully analyzes and

² One of the authors of the paper did ask some responsible person of a DGS team about how rigorous could be considered their currently implemented *Solve* and *Solutions* commands, and the answer was that, except for the polynomial case, the commands were a collection of (very efficient, but) “hacks”....

classifies the different kind of fibers projecting over each point in the zero set of the elimination ideal, as originated in [1]-- is developed in [17] with a whole chapter devoted to this issue, but involving some complex mathematical issues whose description is beyond the scope of our current note. It must be remarked that this advanced approach is a kind of prototype proposal, not yet implemented in any DGS.

3. Envelopes 3D in GeoGebra

As our computational approach to envelopes is quite general, the 3D case can be handled in a quite similar way to the one described in the previous section. Let us start with the following example. We consider the family $F(a, b, c, x, y, z) = a(x-a) + b(y-b) + c(z-c) = 0$ of planes tangent to the circle intersection of a sphere $G_1 = a^2 + b^2 + c^2 - 4 = 0$ and the plane (in green in Figure 4) $G_2 = a + b + c - 1 = 0$. Now, computing the envelope of this family requires just the elimination of the parameters in the ideal generated by the equation of the family, the equations of the sphere and the plane, and the Jacobian $-2ay + 2az + 2bx - 2bz - 2cx + 2cy$ of the 3×3 matrix, where the rows include the partial derivatives of these three equations with respect to a, b , and c . The elimination yields the cone (in red in Figure 4) $7x^2 - 8xy - 8xz + 7y^2 - 8yz + 7z^2 + 8x + 8y + 8z - 48 = 0$. See Figure 4, with GeoGebra 3D view to the right and the CAS view to the left, showing the performance of the mentioned computations.

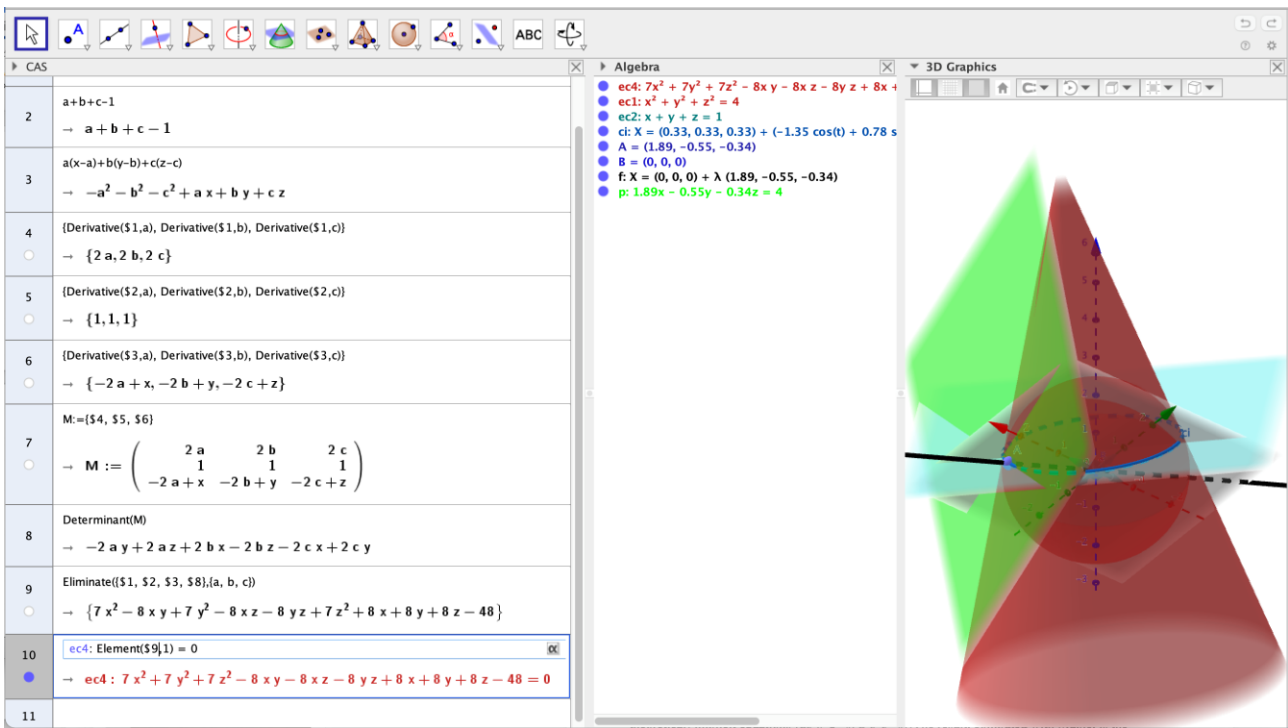


Figure 4: A cone as the envelope of a family of tangent planes to a circle in the unit sphere.

Of course, Elimination can be quite complex as the number of variables increase and, if this is the case, GeoGebra is programmed to abort the computation. See Figure 5, attempting to compute the envelope of a family of unit spheres centered at the points in the twisted cubic (t^3, t^2, t) with (set-theoretical) implicit equations $\{ac - b^2 = 0, a - c^3 = 0\}$. The result, computed with Maple, is the product of the equation of the unit sphere to the fourth power (a degenerate case), times the product of two similar equations, the one displayed in the Appendix to this Note and a similar one obtained replacing y by $-y$ in this equation. It is, thus, a degree 48 surface! See Figure 6, where, copying the

Maple output and pasting it in the Surfer³ input line, we have drawn the twisted tube surface given by the equation in the Appendix.

```

1 (x-a)^2+(y-b)^2+(z-c)^2-1
  → (-a+x)^2+(-b+y)^2+(-c+z)^2-1
2 ac-b^2
  → -b^2+ac
3 a-c^3
  → -c^3+a
4 M:={{Derivative($1,a), Derivative($1,b), Derivative($1,c)}, {Derivative($2,a), Derivative($2,b), Derivative($2,c)}, {Derivative($3,a), Derivative($3,b), Derivative($3,c)}}
  → M := ( 2 a - 2 x  2 b - 2 y  2 c - 2 z
            0      -2 b      0
            1      0      -3 c^2 )
5 Determinant(M)
  → 12 a b c^2 - 12 x b c^2 + 4 b c - 4 b z
6 Eliminate({$1, $2, $3, $5},{a, b, c})
  → Calculation took too long and was aborted
    
```

Figure 5: failing to compute the envelope of a family of spheres centered at points in the twisted cubic.

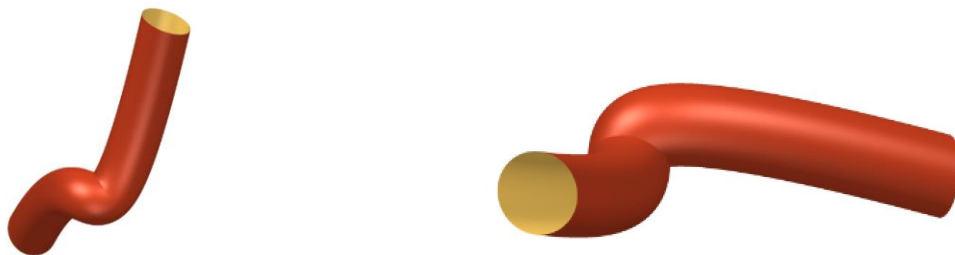


Figure 6: two perspectives of a twisted tube, drawn with Surfer

7. Conclusions

We have presented the advantages of a definition of envelope that allows a) its rigorous implementation in cooperative Dynamic Geometry/Computer Algebra programs such as GeoGebra, b) solving some deficiencies of these programs, when implicit objects are involved as input, instead of geometric constructions, c) easily extending envelope computation to the 3D case.

These issues have been illustrated through some examples and figures, hoping that they will facilitate the introduction, with the help of GeoGebra, of this relevant issue at different educational levels.

Acknowledgements. The authors want to thank the referees for their useful suggestions and several corrections of this paper. Second author partially supported by FEDER/ Spanish Ministerio de Ciencia, Innovación y Universidades – Agencia Estatal de Investigación/MTM2017-88796-P (Symbolic Computation: new challenges in Algebra and Geometry and their applications).

³ An excellent program for drawing surfaces given by implicit equations, see <https://imaginary.org/es/program/surfer>

8. Supplementary Electronic Materials

- [S1] GeoGebra file for Figure 1
- [S2] GeoGebra file for Figure 2
- [S3] GeoGebra file for Figure 3, left
- [S4] GeoGebra file for Figure 3, right
- [S5] GeoGebra file for Figure 4
- [S6] GeoGebra file for Figure 5
- [S7] Surfer file for Figure 6

9. References

- [1] Abanades, M., Botana, F., Montes, A. and Recio, T., *An algebraic taxonomy for locus computation in dynamic geometry*, Computer-Aided Design 56, pp. 22-33, 2014
- [2] Bickel, K., Gorkin, P. and Tran, T., *Applications of envelopes*, Complex Anal. Synerg. **6**, 2(2020). Available from: https://www.researchgate.net/publication/338458122_Applications_of_envelopes [accessed Jan. 30, 2021].
- [3] Blažek, J., Pech, P., *Locus Computation in Dynamic Geometry Environment*. Math. Comput. Sci. 13, pp.31–40, 2019.
- [4] Botana, F. and Recio, T., *Some issues on the automatic computation of plane envelopes in interactive environments*, Mathematics and Computers in Simulation **Volume 125**, pp. 115-125, 2016.
- [5] Botana, F. and Recio, T., *Computing envelopes in dynamic geometry environments*, AMAI (Annals of Mathematics and Artificial Intelligence), Volume 80, Issue 1, pp 3–20, 2017.
- [6] Botana, F. and Recio, T., *A proposal for the automatic computation of envelopes of families of plane curves*, Journal of Systems Science and Complexity 32(1), pp. 150- 157, 2019.
- [7] Bruce, J.W. and Giblin, P.J., *Curves and Singularities*. Cambridge University Press, Cambridge, 2nd edition, 1992.
- [8] Cox, D.A., Little, J. and O’Shea, D., *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra*, Undergraduate Texts in Mathematics, 4th ed., Springer Cham, Heidelberg, New York, Dordrecht, London, 2015.
- [9] Dana-Picard, Th., *The isoptics of an astroid*, Journal of Symbolic Computation 97, pp. 56-68, 2020.
- [10] Dana-Picard, Th. and Kovács, Z., *Networking of technologies: a dialog between CAS and DGS*, The Electronic Journal of Mathematics and Technology, Volume 14, Number 1, 2021.
- [11] Dana-Picard, Th., Naiman, A., Mozgawa, W. and Ciéslak, V., *Exploring the isoptics of Fermat curves in the affine plane using DGS and CAS*, Mathematics and Computer Science 14, pp.45-67, 2020.
- [12] Dana-Picard, Th. and Mozgawa, W., *Automated exploration of inner isoptics of an ellipse*, Journal of Geometry 111(34), 2020.
- [13] Dana-Picard Th. and Zehavi, N., *Revival of a classical topic in Differential geometry: the exploration of envelopes in a computerized environment*, International Journal of Mathematical Education in Science and Technology 47(6), pp. 938-959, 2016.
- [14] Dana-Picard, T. and Zehavi, N., *Automated Study of Envelopes transition from 1-parameter to 2-parameter families of surfaces*, The Electronic Journal of Mathematics and Technology, Volume 11, Number 3, pp. 147-160, 2017.
- [15] Dana-Picard, T. and Zehavi, N., *Automated study of envelopes: The transition from 2D to 3D*, The Electronic Journal of Mathematics and Technology, Volume 13, Number 2, pp. 121-135, 2019.
- [16] Kovács, Z., Recio, T., Vélez, M.P., *GeoGebra Automated Reasoning Tools*. A Tutorial, March 21, 2018. Available at <https://github.com/kovzol/gg-art-doc/blob/master/pdf/english.pdf>

[17] Montes, A., *The Gröbner Cover*, Algorithms and Computations in Mathematics 27, Springer, 2018.

[18] Pottmann, H. and Peternell, M., *Envelopes-computational theory and applications*, Proceedings of Spring Conference in Computer Graphics, pp. 3–23. Budmerice, Slovakia (2000).

Appendix

Equation of the twisted tube (Figure 6):

$$\begin{aligned} &729*x^4*y^6+2187*x^4*y^4*z^2+2187*x^4*y^2*z^4+729*x^4*z^6+1458*x^2*y^8+5832*x^2*y^6*z^2+8748*x^2*y^4*z^4+5832*x^2*y^2*z^6+1458*x^2*z^8+729*y^10+3645*y^8*z^2+7290*y^6*z^4+7290*y^4*z^6+3645*y^2*z^8+729*z^10+1458*x^6*y^3- \\ &4374*x^6*y*z^2+8748*x^5*y^3*z+8748*x^5*y*z^3+2916*x^4*y^5-7776*x^4*y^3*z^2- \\ &10692*x^4*y*z^4+18468*x^3*y^5*z+36936*x^3*y^3*z^3+18468*x^3*y*z^5+1242*x^2*y^7- \\ &5778*x^2*y^5*z^2-15282*x^2*y^3*z^4- \\ &8262*x^2*y*z^6+9720*x*y^7*z+29160*x*y^5*z^3+29160*x*y^3*z^5+9720*x*y*z^7-216*y^9- \\ &2592*y^7*z^2-6480*y^5*z^4-6048*y^3*z^6-1944*y*z^8+729*x^8- \\ &4374*x^7*z+8019*x^6*y^2+8991*x^6*z^2-9234*x^5*y^2*z- \\ &13122*x^5*z^3+13176*x^4*y^4+28971*x^4*y^2*z^2+15039*x^4*z^4-8964*x^3*y^4*z- \\ &15174*x^3*y^2*z^3- \\ &13122*x^3*z^5+3877*x^2*y^6+2259*x^2*y^4*z^2+3807*x^2*y^2*z^4+4401*x^2*z^6- \\ &4104*x*y^6*z-6102*x*y^4*z^3-6804*x*y^2*z^5-4806*x*z^7-2009*y^8-18947*y^6*z^2- \\ &33219*y^4*z^4-18657*y^2*z^6-2376*z^8+7560*x^6*y- \\ &9936*x^5*y*z+13478*x^4*y^3+32112*x^4*y*z^2-49116*x^3*y^3*z- \\ &45180*x^3*y*z^3+9134*x^2*y^5+35628*x^2*y^3*z^2+31068*x^2*y*z^4-40152*x*y^5*z- \\ &74428*x*y^3*z^3- \\ &37476*x*y*z^5+3432*y^7+8690*y^5*z^2+16894*y^3*z^4+11124*y*z^6+151*x^6+7134*x^5*z- \\ &13094*x^4*y^2-10107*x^4*z^2-5180*x^3*y^2*z+10160*x^3*z^3-10066*x^2*y^4- \\ &8008*x^2*y^2*z^2-6687*x^2*z^4-8696*x*y^4*z- \\ &8092*x*y^2*z^3+7026*x*z^5+3784*y^6+31270*y^4*z^2+33178*y^2*z^4+4439*z^6- \\ &19808*x^4*y+15472*x^3*y*z-22664*x^2*y^3- \\ &30496*x^2*y*z^2+35968*x*y^3*z+35168*x*y*z^3-6256*y^5-5648*y^3*z^2-8696*y*z^4- \\ &4408*x^4-1640*x^3*z+4180*x^2*y^2-2092*x^2*z^2+13920*x*y^2*z+2928*x*z^3-1480*y^4- \\ &15884*y^2*z^2-7500*z^4+17176*x^2*y-5536*x*y*z+7968*y^3+2792*y*z^2+5464*x^2- \\ &1120*x*z+912*y^2+3208*z^2-4928*y-1936=0 \end{aligned}$$