symmetry

Article

Computer-Aided Design: Development of a Software Tool for Solving Loci Problems

David Hernández-Díaz [1], Francisco Hernández-Abad [2], Vicente Hernández-Abad [2], Ricardo Villar-Ribera [3], Fernando Julián [4] and José Ignacio Rojas-Sola [5,*]

[1] TR5 Building, Terrassa Campus, Polytechnic University of Catalonia, 08222 Terrassa, Spain
[2] Department of Engineering Graphics and Design, TR5 Building, Terrassa Campus, Polytechnic University of Catalonia, 08222 Terrassa, Spain
[3] Department of Engineering Graphics and Design, Manresa Campus, Polytechnic University of Catalonia, 08242 Manresa, Spain
[4] Department of Organization, Business and Product Design, University of Girona, 17003 Girona, Spain
[5] Department of Engineering Graphics, Design and Projects, University of Jaen, 23071 Jaen, Spain
* Correspondence: jirojas@ujaen.es; Tel.: +34-953-212452

Abstract: In this work, we developed a software tool for the simplified, automated definition and viewing of loci in the plane, an essential task in the design, engineering and architectural fields. The process was started by identifying the most salient geometric loci in the engineering domain (i.e., tangents). A geometric approach was followed to deal with curves from the relative positions of two circles or a circle and a point. The algorithms used can be easily implemented in commercially available CAD software, which testifies to the usefulness and timeliness of the proposed tool. The tool successfully solved all target cases and was implemented as a function with the commercial software Adobe Authorware v.7.0.2.

Keywords: computer-aided design; computer-aided sketching; loci; geometric constraints; dimensional constraints

Citation: Hernández-Díaz, D.; Hernández-Abad, F.; Hernández-Abad, V.; Villar-Ribera, R.; Julián, F.; Rojas-Sola, J.I. Computer-Aided Design: Development of a Software Tool for Solving Loci Problems. *Symmetry* **2023**, *15*, 10. https://doi.org/10.3390/sym15010010

Academic Editors: Victor Eremeyev and Chin-Ling Chen

Received: 19 November 2022
Revised: 9 December 2022
Accepted: 17 December 2022
Published: 21 December 2022

1. Introduction

Elucidating the geometric properties of the elements of nature has been crucial for human development. Continuous evolution for millions of years has allowed many living beings to adopt their optimal geometric forms. This process has always been observed with interest by humans, who have carefully studied the forms around them, and their functions, aiming to find practical uses. Recently, for example, the geometry of dandelions was found to facilitate their propagation by inducing the formation of a vortex ring in response to the wind. This effect might be explored as a means of improving the flying efficiency of small aircraft [1]. This is but one example of the many illustrating the benefits that science can derive from geometric analysis (particularly in the design, engineering and architectural fields) [2].

Although computer-aided design (CAD) was conceived back in the early 1950s, it was not until 1963 that Ivan E. Sutherland developed the first proper CAD tool: SketchPad [3]. Since then, but especially after the inception of parametric design [4], commercial CAD software has dramatically evolved by introducing computer-assisted engineering (CAE) tools for optimal, more-economical design [5]. The best-known and most widely used CAE tools are those for finite element analysis (FEA) and computational fluid dynamics (CFD). Some CAE tools have been especially difficult to integrate into computer design software and success in this area can be viewed as a technological landmark because CAD and, for example, FEA, use rather different approaches [6]. In fact, each type of software has specific requirements for geometric processing that must be fulfilled by using reliable, flexible algorithms [7].

In the past two decades, CAD software development has relied on a functional branch called "computer-aided innovation" (CAI). CAI is intended to stimulate creativity among designers and engineers and to boost the development of innovative, longer-lived products [8,9]. At present, CAI tools play a major role in product design [10,11], which has been deemed the main route for the long-term development of organizations and a key factor for corporate survival [12]. Sketch-based modelling (SMB), which is probably the primary CAI tool, is a current focus of CAD research and development towards the simplified creation of 3D shapes from sketches [13,14]. Sketching is in fact a dynamic process with high relevance to product development [15–19]. However, digitizing sketches and automatically converting them into 3D shapes remains a complex, costly and not-always-efficient task [20]. In any case, CAD software has, unarguably, dramatically evolved since its inception, when it was conceived to simplify the representation and communication of complex technical elements such as parts, mechanisms, facilities or buildings. With time, however, CAD software has become an integral solution for a wide variety of design, engineering and architectural problems.

Automating geometric representation with CAD software has enormously facilitated engineering and architectural drawing tasks. To many experts, universities and scientific and technical schools, mastering CAD software has become even more important than having a sound knowledge of geometry and its representations [21,22]. Most daily objects are industrially designed in simple shapes for easier manufacturing and management. Other objects, however, are still functional but geometrically complex. Such is the case with ergonomic products, aircraft or large structures, for example, the construction of which often requires drawing complex classic geometric forms for which existing software falls short [23]. In some cases, to address specific problems, designers require special geometric knowledge they have not been taught. Digital representation has allowed the Euclidian drawing tools (ruler, square, bevel, compass) to be replaced with a much more efficient alternative: CAD software. However, may graphical representation processes, including sketching, require a solid knowledge and skill in classic geometry [24]. Also, conceiving and creating new products often requires mastering geometry, 3D viewing and some special skills [21,25]. It can therefore be highly useful to expand the functionality of CAD software for classic geometry in order to allow designers with a limited knowledge of geometry to address more-complex problems. Also, educational institutions should be aware of the significance of geometry to innovation [26], a key factor for prosperity and economic development, and make students especially proficient in this branch of mathematics.

The primary aim of this work was to develop a software tool to simplify and automate the identification and viewing of geometric loci in the plane. Loci problems are considered difficult in mathematical and graphical subjects all over the world [27], and their visual information can be useful in learning and professional contexts [28]. The tool, which is intended for educational used, can be implemented in any type of commercially available CAD software. This is because the application involves, as an input of their functions, simple entities that exist in current CAD software (points, line, circumferences, etc.) and mathematical equations to calculate the graphical solution. Although the range of potential problems involving planar loci is almost unlimited, the proposed tool was conceived to address only a few. Its potential benefits where previously discussed by the authors elsewhere [29]. No similar tools have to date been reported. Research started by identifying geometric locus cases of special engineering significance. Then, geometric mathematics was used to solve each individual case, and the procedure was implemented, as a function, to allow future users to select the most suitable among those provided according to the particular case to be addressed. The major advantage of the tool is to be able to get the solution to different loci problems in plane only through the selection of the graphical entities involved in the problem and also the selection of the type of loci case. In contrast with commercial available software, there is no need to make any graphical procedures. All functions of the tool were checked for correct performance.

2. Definitions and Foundations

A geometric locus (Figure 1) is an entity in which all points, whether in the plane or in space, fulfil a given geometric condition (e.g., distance, angle, another parameter). In other words, a locus is a set of points that share the same metric condition. The plural of locus is loci. Relying on this definition to solve graphical problems involving tangents proved highly useful in the past as it allowed a substantial number of graphical procedures to be replaced by a unified approach [30]. However, the range of potential cases is almost unlimited, and each individual instance must be examined in isolation for optimal resolution [29].

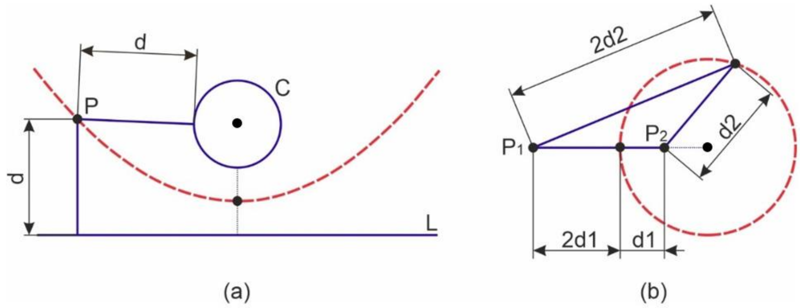


Figure 1. Examples of conical curves as loci: (**a**) equidistance between a straight line and a circle; (**b**) locus of points at given distance from a fixed point and twice that distance from another.

The number of geometric locus functions to be included in the proposed tool was restricted by compiling a series of exercises of tangents in the plane taught in the subject engineering graphics at the Aerospace Engineering School of the Polytechnical University of Catalonia. On the basis of occurrence frequency and complexity, the most interesting geometric loci for teaching purposes were found to fall into two categories:

1. Those involving a circle and an external point where the point curve (hyperbola) containing the centres of all possible circles, tangent to the given circle and passing through the point, was to be analytically defined.
2. Those involving any two mutually external circles where the point curve (hyperbola branch) containing the centres of a circle tangent to the two given circles was to be analytically defined.

A total of six cases (Table 1) were thus identified according to the geometric requirements of the tangent circle for implementation in the proposed tool. The first two sections of this paper examine and solve the six cases for the characteristics of the curves fulfilling the specific requirements (especially the hyperbola and which branch meets the conditions).

Table 1. Specific cases implemented in the proposed software tool.

Type	Geometric Restrictions on Tangency	Case Number
A	An external tangent to the circle passing through the point	1
	An internal tangent to the circle passing through the point	2
B	An internal tangent to both circles	3
	An external tangent to both circles	4
	An external tangent to the first circle and an internal tangent to the second	5
	An internal tangent to the first circle and an external tangent to the second	6

As shown below, the output curves were applicable to other relative positions of the circles (e.g., tangent, secant, inscribed), albeit with some restrictions. These special cases will be the subject of future work because some led to other types of conical curves or because a curve segment failed to share the characteristics of the tangent (e.g., with circles of an identical radius, the result is a straight line). The analysis of each case was followed by two steps:

1. The implementation of the routines allowing software to depict the correct function in situ and checking that the results were accurate by providing a graphical output as in professional CAD software.
2. Defining and solving an exercise to illustrate the process and envisage the potential for solving actual geometries in order to compare the proposed procedures with those used by existing graphical software.

In this sense, the procedure for solving the loci cases referred to in Table 1 is not automated in any available CAD software, to the authors' knowledge, making it necessary to carry out graphic procedures that can be complex and lengthy and where the solution is often not 100% accurate (because they are often approximations that cause errors in subsequent trim, expand or tangency operations). Automation is a relevant advantage because the user is required to select only the entities that participate in the problem and define the locus to be solved. The function returns to the user the calculated and precise solution for the indicated problem without the need for graphical procedures.

Case 1: With a circle and an external point (Figure 2), any circle tangent to the given circle and passing by the given point will fulfil the following conditions:

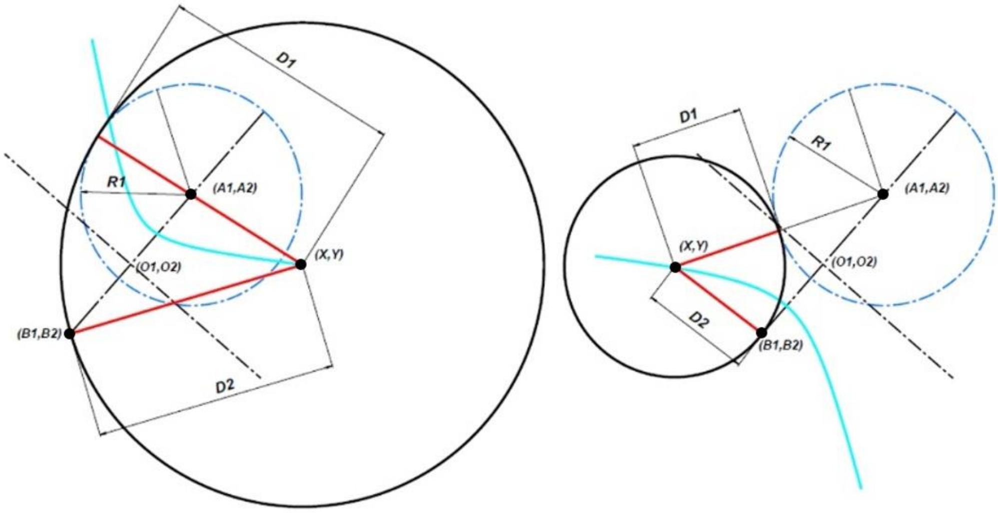


Figure 2. A circle and an external point, together with the curve containing all possible centres of the circles that are tangent interior (**left**) and exterior (**right**) to the given circle and pass by the given point (hyperbolic branches).

- Its centre will be a point in the geometric locus of the points lying at the smallest possible distance from the circle coinciding with its distance from the point. This will allow the centres of all circles passing by the point and being external tangents to the given circle to be identified.
- Its centre will be a point in the geometric locus of the points lying at the greatest possible distance from a circle coinciding with its distance from the point. This will allow the centres of all circles passing by the point and being internal tangents to the given circle to be identified.

The geometric locus of the points with variable coordinates (X1, Y1) (i.e., the curve where the centres of circles tangent to the given circle of radius R1 and centre A (A1, A2) that also pass through the point B (B1, B2)) is shown in cyan in Figure 2. Although the Figures are shown separately for easier understanding, the circle and the point in both are identical. Also, the geometric loci are the two branches of a hyperbola whose axis is at one-half the distance between points A (A1, A2) and B (B1, B2) and normal to the connecting segment. One of the circles for each case is highlighted in black in the Figure. D1 and D2 are the distance between the original entities' circles and point to the point (X, Y) of the loci curve, respectively. According to the definition of the locus, D1 and D2 are equal.

Although the foregoing is obvious, the discussion is simplified by taking this case to be a particular one of the next, where the point is assumed to be a circle of zero radius. It will thus suffice to identify and solve the cases of two circles to check the assumptions.

Case 2: In the case of circles tangent to two external ones of radius R1 and R2 (Figure 3), one of the following conditions must be fulfilled:

- The centre will be a point in the geometric locus of the points lying at the maximum distance from the two circles, which will allow the centres of all circles having the two given circles as internal tangents to be determined.
- The centre will be a point in the geometric locus of the points lying at the shortest possible distance from the two given circles, which will allow the centres of all externally tangent circles to be identified.
- The centre will be a point in the geometric locus of the points lying at the shortest possible distance from one given circle and greatest from the other, which will allow the centres of all circles being externally tangent to the first and internally tangent to the second to be identified.
- The centre will be a point in the geometric locus of the points lying at the greatest distance from one of the given circles and shortest from the other, which will allow the centres of all circles being internally tangent to the former and externally tangent to the latter to be identified.

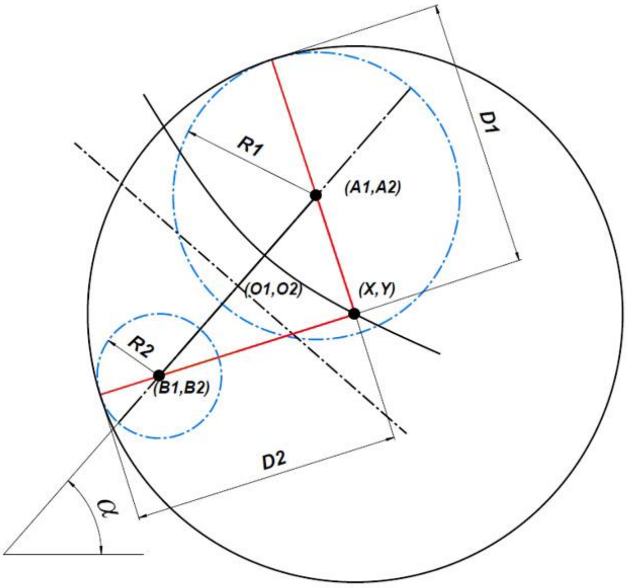


Figure 3. External circles and curve containing all possible centres of the circles that are internally tangent to them (hyperbolic branch).

As shown below, the solution to this case is a hyperbolic branch. With secant or inscribed circles, the solutions can differ, however. Therefore, only those cases where the solution is a whole hyperbola branch or a portion are considered.

The problem is depicted graphically in Figure 3, the curve fulfilling the requirements and the given (external) circles shown together with the solution (i.e., a circle being internally tangent to both given circles whose centre lies in the curve).

The given circles, in blue, represent any combination of external circles including external tangents. The maximum distances from the points in the curve (in red) are denoted by their generic coordinates (X, Y) and represent the whole curve—a condition that allows the equation making all distances identical to be established.

For the purpose of better characterizing the curve by considering relationships between tangents, the curve can be assumed to contain the centres of all circles having the given circles as internal tangents and hence fulfilling the following conditions for distances:

$$D1 = D2 \tag{1}$$

$$\sqrt{(\,(X-A1)^2+(Y-A2)^2)}+R1=\sqrt{((X-A1)^2+(Y-A2)^2)}+R2 \tag{2}$$

$$\sqrt{(X-A1)^2+(Y-A2)^2}-\sqrt{(X-A1)^2+(Y-A2)^2}=R2-R1 \tag{3}$$

As can be seen, the points in the curve (X, Y) must meet the condition that the difference between the distances at two points A (A1, A2) and B (B1, B2) should be constant and identical with the difference between radii, which constitutes the definition of a hyperbolic branch. Simply making R2 = 0 provides the equation for one of the branches of case A (a circle and an external point), which is depicted in Figure 2, left.

The hyperbola can be better characterized by considering the other branch and its symmetry axis. Obviously, for each coordinate X, there will be two points in the curve lying in the same vertical line and in the curve that is symmetric to the hyperbolic axis. This information allows one to determine the gradient α and its centre (O1, O2), which in turn will allow the axes to be transformed in order to place the centre at point (0, 0).

The centre of the hyperbola will fall at the midpoint of the segment connecting the centres of the two circles; also, its coordinates can be expressed in terms of the gradient of the centre-connecting segment, which is taken to be the angle α as measured counter-clockwise:

$$O1 = A1 + \overline{AB} * cos(\alpha)/2 \tag{4}$$

$$O2 = A2 + \overline{AB} * sin(\alpha)/2 \tag{5}$$

The coordinates of the midpoint of the connecting segment can also be expressed in terms of those of the two centres:

$$O1 = (A1 + B1)/2 \tag{6}$$

$$O2 = (A2 + B2)/2 \tag{7}$$

However, the curve can be better characterized by placing it at point (0, 0) with a vertical hyperbolic axis and then making the connecting axis horizontal by rotating the whole system by an angle –α:

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(-\alpha) & -\sin(-\alpha) & 0 \\ 0 & \sin(-\alpha) & \cos(-\alpha) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{8}$$

The last step involves translating the whole system as needed to have the centre of coordinates fall at (0, 0) in the new system of axes, where all parameters can be more easily identified.

Because neither rotating nor translating the system will affect coordinate Z, the combined transformation defining the two transformations can be expressed with the following matrix:

$$\begin{pmatrix} 1 & 0 & 0 & -O1 \\ 0 & \cos(-\alpha) & -\sin(-\alpha) & -O2 \\ 0 & \sin(-\alpha) & \cos(-\alpha) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{9}$$

The four intermediate values correspond to rotation about Z for horizontal positioning, and the first three values in the rightmost column define the vector for the translation of the coordinate centre in the X, Y, Z direction under constancy of Z (i.e., ΔZ = 0), under the provision that the signs of coordinates X and Y at the centre of the hyperbola have been reversed. As can be seen in Figure 4, the hyperbola is easier to recognize with its axes centred at point (0, 0).

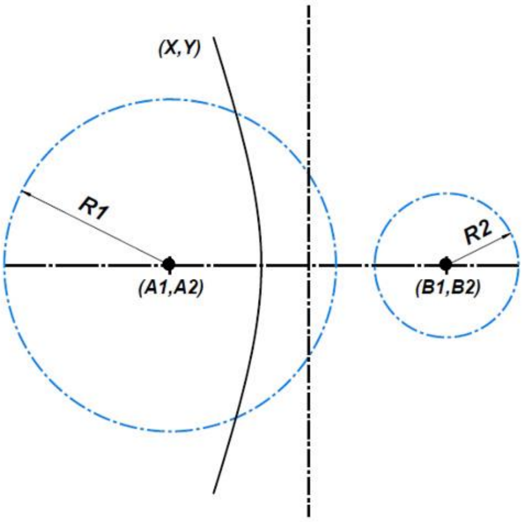


Figure 4. Canonical position of the hyperbola upon transformation.

The other branch will obey the following equation:

$$\sqrt{(X-A1)^2+(Y-A2)^2}-R1=\sqrt{(X-A1)^2+(Y-A2)^2}-R2 \quad (10)$$

$$\sqrt{(X-A1)^2+(Y-A2)^2}-\sqrt{(X-A1)^2+(Y-A2)^2}=R1-R2 \quad (11)$$

In this more easily recognized form, the hyperbola can be fully characterized. The process is made easier by using its canonical equation, defining the other branch and viewing the asymptotes (Figure 5).

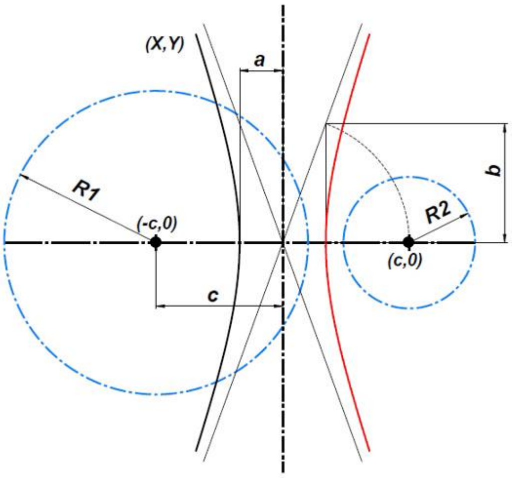


Figure 5. Whole hyperbola (canonical position) and definition of the parameters of the canonical equation of the hyperbola.

As can easily be seen by viewing the two branches in their original position, the other branch can be useful with externally tangent circles (Figure 6); also, the centre of the hyperbola lies at the midpoint of the segment connecting the centres of the circles—where point O (O1, O2) is to fall upon rotation and translation.

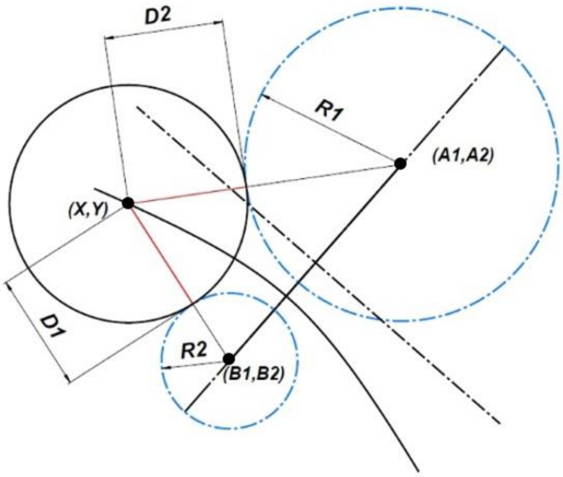


Figure 6. Hyperbolic branch corresponding to the geometric locus of all possible centres of circles that are external tangents to the given circles.

Once the centre of the hyperbola is at the centre of coordinates, the equation can be expressed in canonical form, which is useful to calculate y in terms of x,

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1 \tag{12}$$

all other parameters being known. Thus, c is one-half the distance between the centres of the circles,

$$c = \bar{A}B/2 \tag{13}$$

And a is the distance from the midpoint of the horizontal segment, which coincides with the maximum distance from the circles to point (0,0):

$$a = \frac{R1 - R2}{2} \tag{14}$$

On the basis of the relationships between the previous three parameters, b can be expressed as:

$$b = \sqrt{c^2 - a^2} \tag{15}$$

In our case, y can be obtained in terms of x as follows:

$$y^2 = b^2\,(1 - \frac{x^2}{a^2}) \tag{16}$$

$$y = \sqrt{(b^2\,(1 - \frac{x^2}{a^2}))} \tag{17}$$

With $-$a $\leq x \geq$ +a, this equation allows one to easily calculate the y values needed to draw the curve and represent the target geometric locus by using a suitable mathematical method.

Ensuring that the curve will be correctly placed and rotated only requires subjecting the points to the transformation that restores their original places by using the following matrix:

$$\begin{pmatrix} 1 & 0 & 0 & O1 \\ 0 & \cos(\alpha) & -\sin(\alpha) & O2 \\ 0 & \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{18}$$

As can be seen, angle α is now positive, and so are the coordinates of the centre of the hyperbola, which has become the entire geometric locus because its points are all at an identical maximum distance from the two circles. Therefore, all are possible centres of the circles that are internal tangents to the given circles (i.e., the new circle will contain the other two as internal tangents). Likewise, any of the points of the other hyperbolic branch (Figure 6) will be at an identical minimum distance from the two circles; with secant circles, however, not all points can be centres of externally tangent circles—rather, the segment joining the points of contact with the original circles will inevitably contain the potential centres for internally tangent circles as such centres will fall inside both circles.

We thought it especially important to identify the hyperbolic branch corresponding to the particular type of tangency. For this reason, we focused on algorithms allowing the target curve to be identified and individually obtained. Geometric design is made easier by CAD software capable of showing the target curve only.

The problem of a circle that is tangent to another two can be approached in a number of ways, depending on its specific properties and on the particular type of tangency. This paper focuses on the geometric loci that allow the hyperbolic curves containing the centres of circles exhibiting various types of tangency (external−external, internal−external, external−internal and internal−internal) to be defined.

We have so far examined two hyperbolic branches corresponding to circles that are externally or internally tangent to the given circles (Cases 1 and 2). There are, however, another two branches corresponding to mixed tangency types (i.e., to an externally tangent circle and an internally tangent one: Cases 3 to 6). These other equidistant curves obey the equations that equalize the maximum distance to a circle and the minimum distance to the other. Therefore, they will form another two hyperbolic branches.

The tangent circles must be externally tangent to the first given circle, centred at A (A1, A2), and internally tangent to the second, centred at B (B1, B2) (Figure 7).

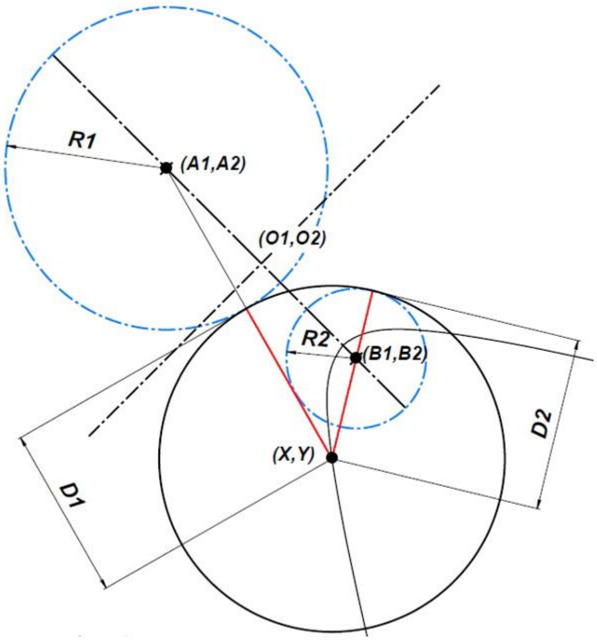


Figure 7. Curve (hyperbolic branch) corresponding to the geometric locus of the centres of circles that are external tangents to that of radius R1 and internal tangents to that of radius R2.

The equidistant curve will obey the following equation:

$$\sqrt{(X-A1)^2+(Y-A2)^2}-R1=\sqrt{(X-A1)^2+(Y-A2)^2}+R2 \quad (19)$$

$$\sqrt{(X-A1)^2+(Y-A2)^2}-\sqrt{(X-A1)^2+(Y-A2)^2}=R1+R2 \quad (20)$$

On the other hand, when the tangent circles are internally tangent to the first given circle and externally tangent to the second, the applicable equation is:

$$\sqrt{(X-A1)^2+(Y-A2)^2}+R1=\sqrt{(X-A1)^2+(Y-A2)^2}-R2 \tag{21}$$

$$\sqrt{(X-A1)^2+(Y-A2)^2}-\sqrt{(X-A1)^2+(Y-A2)^2}=-R1-R2 \tag{22}$$

These two cases can be approached very similarly to those of the other two hyperbolic branches. Therefore, the process will involve taking the curve to its canonical position and identifying the parameters that are altered by the transformation in order to include them in the routines—which should also provide for two branches resulting from the opposite relationships to the given circles.

Figure 8 depicts the two canonical hyperboles at the position of the starting circles. The branch on the right is the geometric locus containing the centres of all circles that are internal tangents to the circle of radius R2 and external tangents to that of radius R1.

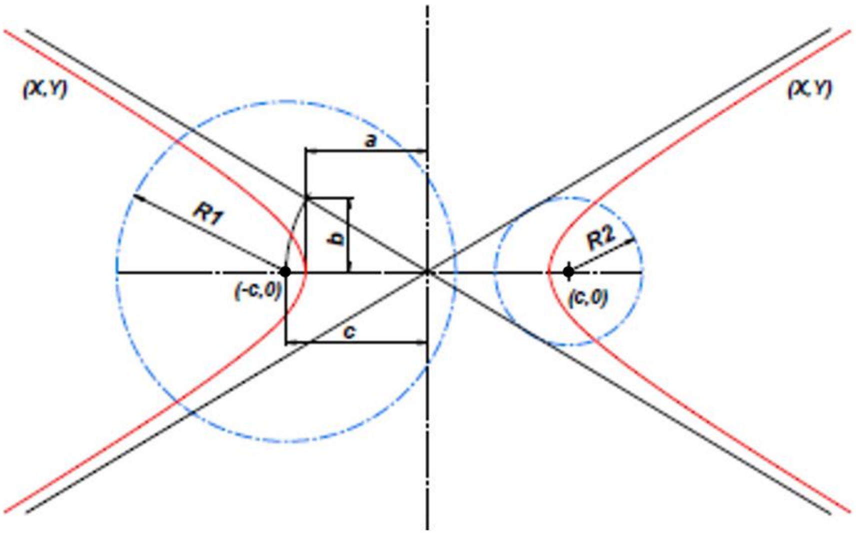


Figure 8. Curve (hyperbola) corresponding to the geometric locus of the centres of circles that are external tangents to the circle of radius R1 and internal to that of radius R2, and vice versa (two branches).

This allows one to use very similar algorithms for the four curves. In fact, at their canonical position, the two branches of each hyperbola will obey the same equation—specific parameters excepted—as a result of their having a common centre:

$$y^2=b^2\,(1-\frac{x^2}{a^2}) \tag{23}$$

$$y=\sqrt{(b^2\,(1-\frac{x^2}{a^2}))} \tag{24}$$

Also, because c will be identical, only a and b need be changed. Thus, a can be calculated as:

$$a=\frac{R1+R2}{2} \tag{25}$$

For obvious reasons, some of these curves (specifically, the first two examined here) can also be used with secant circles because the last two are required to be external. This is so because with secant or inscribed circles, the geometric loci containing the centres of the circles that are externally tangent to one and internally tangent to the other are not hyperbolic but conical curves of a different type and thus warrant separate treatment.

A tool for viewing and using the previous four types of geometric loci is not a purely theoretical development. In fact, such loci are useful to solve real-life problems, and implementing the tool in CAD software can enormously facilitate their geometric definition. Appendix A shows the feasibility of dealing with the different types of curves by implementing the required algorithms in CAD software.

3. Practical Example

For example, the practicality of the proposed tool is demonstrated here by solving the following problem: identifying the geometric locus of the points falling at the maximum possible distance from two given circles (Case 1). Such a locus will contain the centres of all circles internally tangent to the two given circles (Case 3). This problem has an infinite number of solutions that can be reduced to a unique one by imposing some constraints.

This can indeed be a real-life problem, such as finding where the centre of the red cogwheel in Figure 9 is to be placed in order to have its periphery externally connected to the outside of two straight-toothed cogwheels (green in the figure) of fixed size and position. All three cogwheels are assumed to have an identical modulus—otherwise, they would not engage properly.

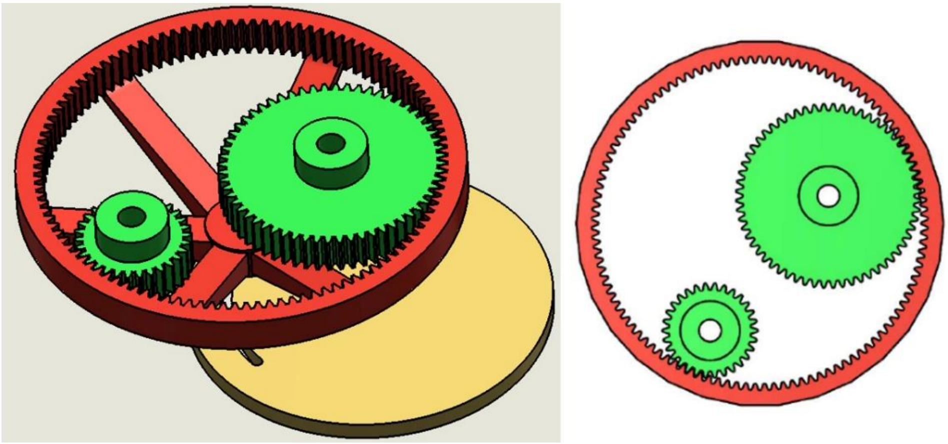


Figure 9. 3D and planar view of a cogwheel system illustrating the usefulness of geometric loci.

The original circles of the three cogwheels can be dealt with like the theoretical geometry problem of identifying circles having an internal tangent to the two fixed circles. Because there can be infinite solutions, this is an indeterminate problem; in all, however, the centre will lie in a curve at the same maximum distance from the two fixed circles.

Because properly defining the problem entails imposing some constraint, we used another condition involving geometric loci in the plane (e.g., the centre should lie at a distance of two-thirds from the centres of the fixed cogwheels). This condition is depicted in green in the next figure and clearly illustrates that nonequidistant geometric loci can provide a highly useful tool for implementation in CAD software.

Imposing a spatial constraint would lead to a much more complex problem that is not addressed in this paper, because no 3D geometric loci are dealt with here. As we all know, however, the actual geometry of objects includes an infinite number of geometric loci that very often require points in lines or surfaces in space to fulfil some metric conditions.

Figure 10 shows a circular locus that can be illustrated in practice with a slotted disc (Figure 11) along which the output cogwheel can move.

These characteristics are depicted in the design of Figure 12, where the target toothed cogwheel has been excluded for easier viewing of the slot.

As can be seen, implementing these geometric loci in commercial CAD software would allow two solutions to be obtained and the right one to be identified.

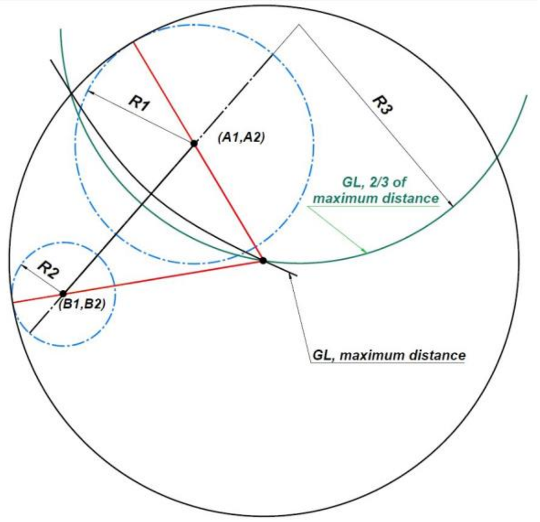


Figure 10. Graphical depiction of the problem involving the original circles of the two cogwheels in Figure 9.

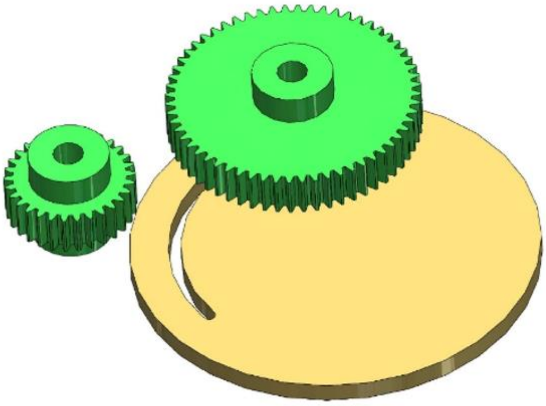


Figure 11. 3D view of the slotted disc.

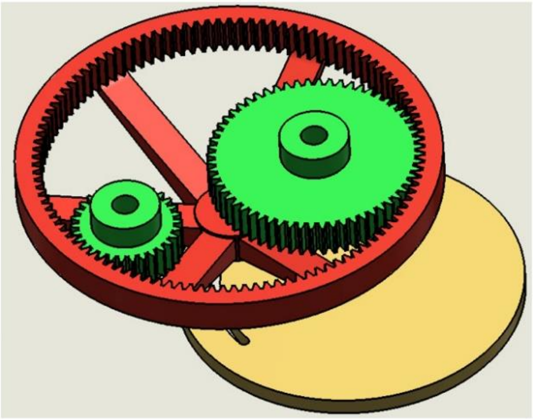


Figure 12. 3D view of the ensemble. The output cogwheel is shown in red.

4. Implementation of the Software Tool

The above-described process was illustrated and checked here by developing a tool containing a toolbar populated with buttons similar to those in top commercial CAD software to allow the most common types of equidistant geometric loci used in design processes to be used.

Drawing a hyperbola involves building a matrix of ordered points and representing it in the form of local spline curves (specifically, curves of the second order formed by three points each) that will obey the following equation:

$$y = ax^2 + bx + c \tag{26}$$

The initial curve starts at the precise point of the axis where the tangent is normal to the axis. Each three-point segment of the curve must meet the following conditions:

- The ends connecting the segment to the next should have the same tangent.
- The tangent at its midpoint should have the same direction as the segment connecting the previous and next point.
- The segment should pass through the three points.

These boundary conditions provide a hyperbolic branch defined by the size and position of the initial circles according to their specific relationship (external−external, internal−internal, external−internal or internal−external). Thanks to its smoothness, the resulting curve will be very similar to the actual curve, provided that it contains a large-enough number of points. Also, when the curve point to be used to draw the circle is selected, the proposed tool accurately determines the centre and radius of the circle.

As confirmed by using the tool, these curves are potential centres of tangent circles with the above-defined characteristics—in this case, the four hyperbolic branches specified for any pair of external circles. The tool could be expanded to address circles externally tangent to one of the given circles and externally tangent to the other when the two are secant or inscribed. Such cases involve other characteristics and will be examined in future work.

The proposed tool was developed with the aid of the software Authorware v. 7.0.2 from Adobe (San José, CA, USA), which uses instructions similar to those of in C++ programming language. This software was previously used by the authors to develop various interactive tools [31,32]. Figure 13 depicts a simplified flowchart for the process.

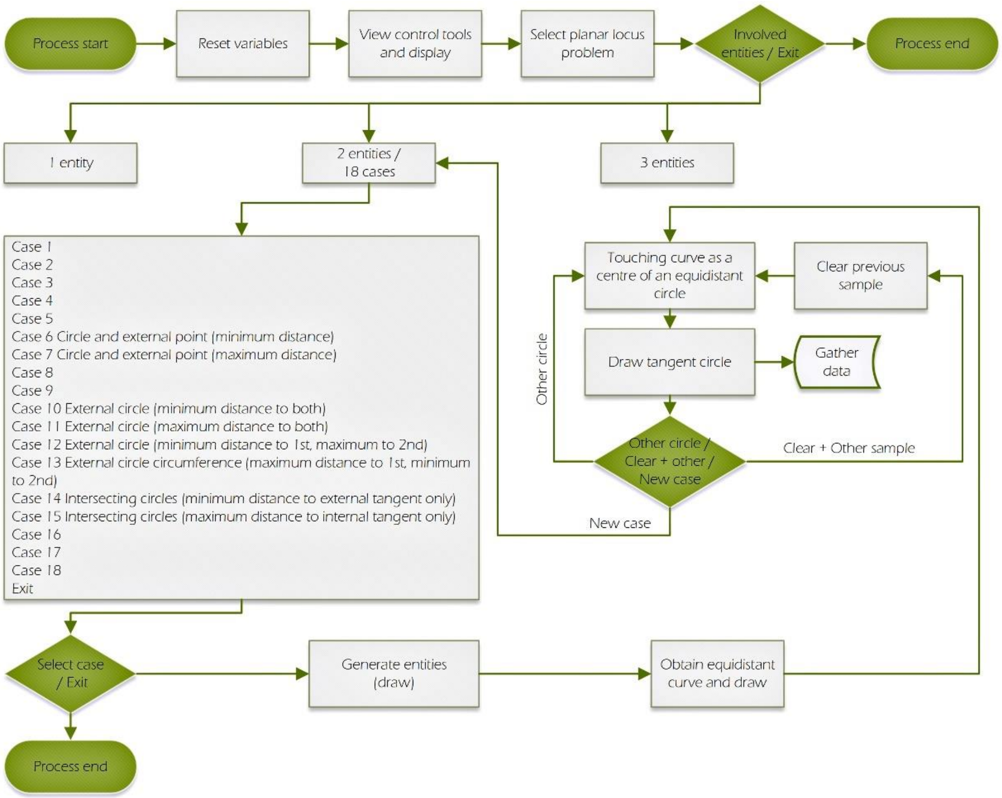

Process start
Reset variables
View control tools and display
Select planar locus problem
Involved entities / Exit
Process end
1 entity
2 entities / 18 cases
3 entities
Case 1
Case 2
Case 3
Case 4
Case 5
Case 6 Circle and external point (minimum distance)
Case 7 Circle and external point (maximum distance)
Case 8
Case 9
Case 10 External circle (minimum distance to both)
Case 11 External circle (maximum distance to both)
Case 12 External circle (minimum distance to 1st, maximum to 2nd)
Case 13 External circle circumference (maximum distance to 1st, minimum to 2nd)
Case 14 Intersecting circles (minimum distance to external tangent only)
Case 15 Intersecting circles (maximum distance to internal tangent only)
Case 16
Case 17
Case 18
Exit
Touching curve as a centre of an equidistant circle
Clear previous sample
Other circle
Draw tangent circle
Gather data
Other circle / Clear + other / New case
Clear + Other sample
New case
Select case / Exit
Generate entities (draw)
Obtain equidistant curve and draw
Process end

Figure 13. Simplified flowchart of the process.

Because Authorware's programming language allows graphical viewing as a scheme (Figure 14), we chose to show the four main routines (bottom right) used to obtain the six target hyperbolic branches. The first two allowed the hyperbolae with R2 = 0 (i.e., the case of a circle and a point), which were combined into a particular case of the first two (i.e., two circles), to be solved. Appendix A shows some of the programming code used.

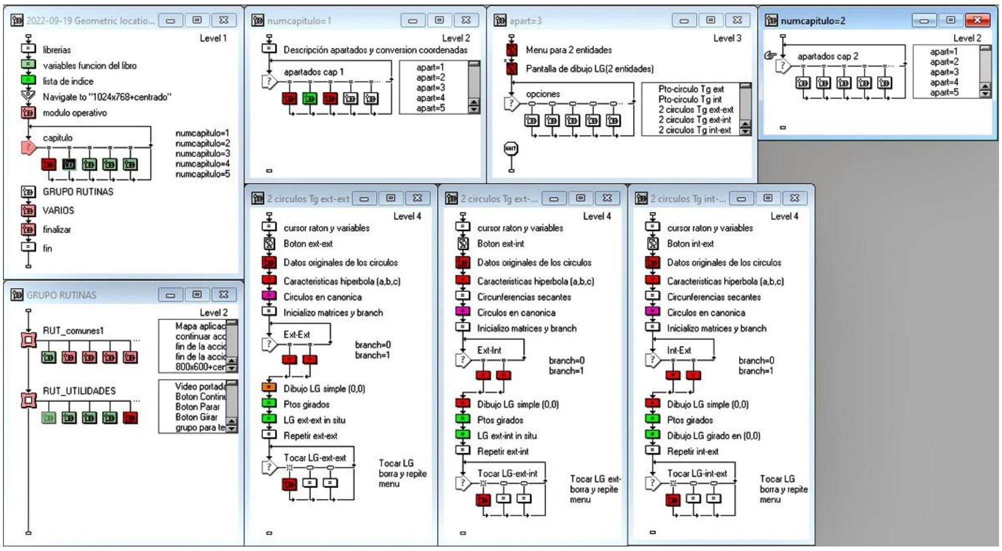


Figure 14. Implementation of the procedures used to represent the hyperbolic branches in graphical form.

Below are shown various sections of the tool's illustrating the use of specific routines to confirm that the sought auxiliary curves are indeed obtained and also that they fulfil every requirement to become highly relevant auxiliary elements for the qualitative improvement of existing professional CAD software.

Although the proposed tool was conceived to examine and incorporate various types of loci, this paper deals with only a few involving two entities (specifically, a point and a circle, or two circles external to each other), the minimum and maximum distance between which is considered. Each point is assumed to be a potential centre of the circle tangent to both entities—and hence of "zero" radius.

Figures 15 and 16 show the interface and index, respectively, of the tool.

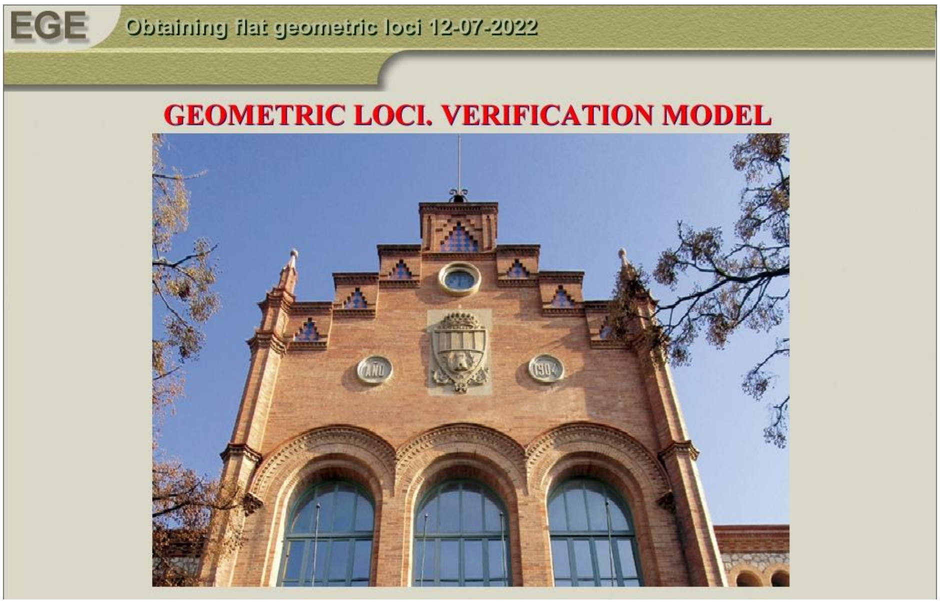
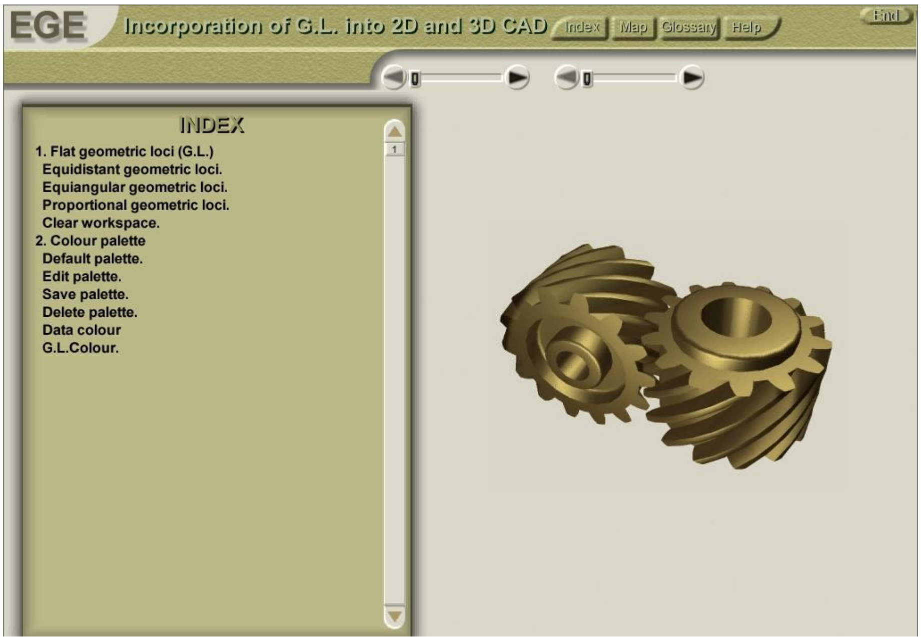

EGE
Obtaining flat geometric loci 12-07-2022
GEOMETRIC LOCI. VERIFICATION MODEL
ANO
1904

Figure 15. GUI of the proposed tool.

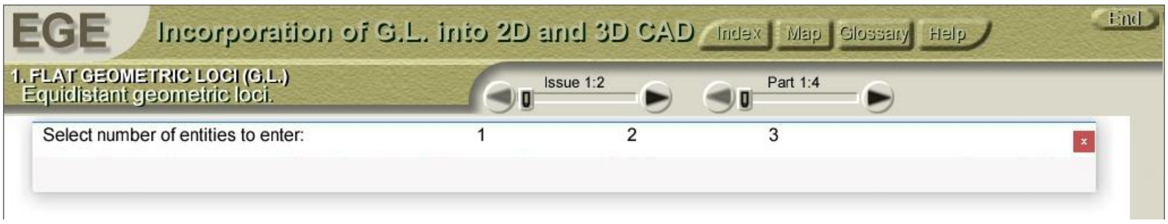


INDEX

1. Flat geometric loci (G.L.)
   Equidistant geometric loci.
   Equiangular geometric loci.
   Proportional geometric loci.
   Clear workspace.
2. Colour palette
   Default palette.
   Edit palette.
   Save palette.
   Delete palette.
   Data colour
   G.L.Colour.

Figure 16. Functions of the proposed tool.

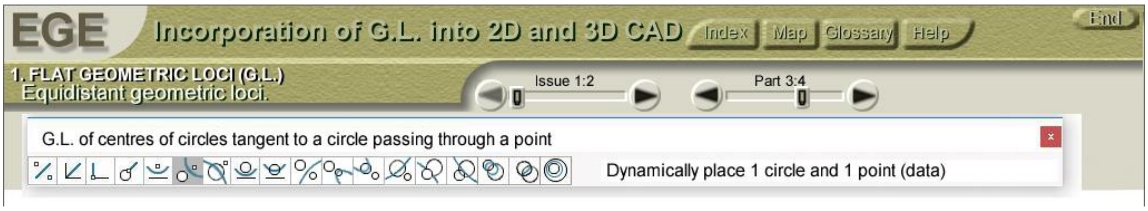
Because it is intended to encompass all possible cases and because cases might involve one, two or three entities, the initial screen allows one to choose the number of entities involved in the equidistance situation (Figure 17).


EGE
Incorporation of G.L. into 2D and 3D CAD
Index
Map
Glossary
Help
End
1. FLAT GEOMETRIC LOCI (G.L.)
Equidistant geometric loci.
Issue 1:2
Part 1:4

Select number of entities to enter: 1 2 3

Figure 17. Selection of the number of entities to be used as input.

Once the number of entities has been selected, the tool will show all possibilities for points, lines or circles (i.e., all possible pairs of combinations: point—point, point—line, point—circle, line—line, line—circle and circle—circle). The previous six cases can be split into a minimum of 18 to deal with maximum or minimum distances (e.g., with externally and internally tangent circles, respectively). For this reason, the interface includes 18 access buttons (Figure 18).


EGE
Incorporation of G.L. into 2D and 3D CAD
Index
Map
Glossary
Help
End
1. FLAT GEOMETRIC LOCI (G.L.)
Equidistant geometric loci.
Issue 1:2
Part 3:4

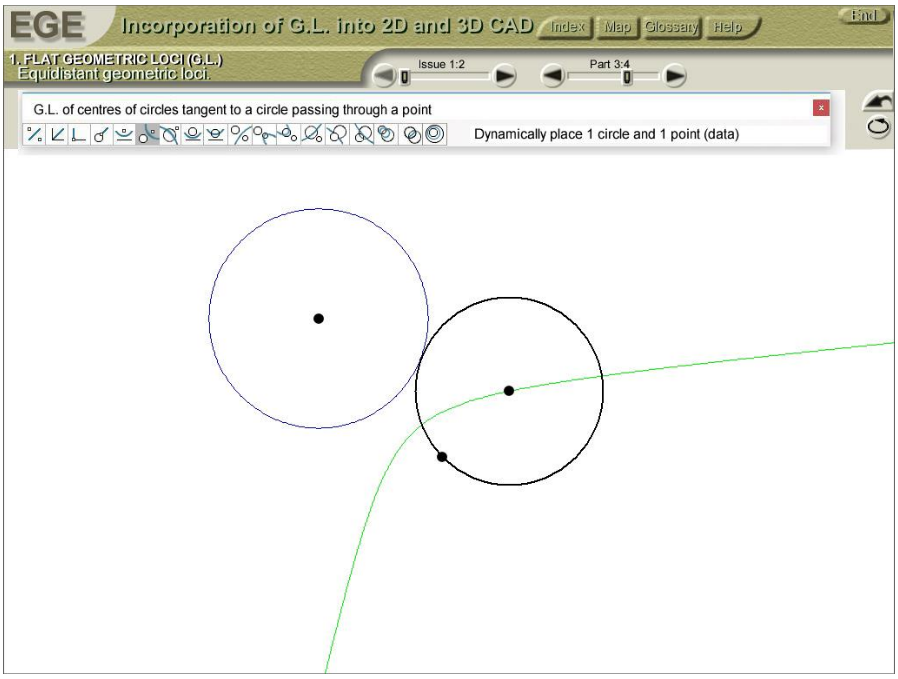
G.L. of centres of circles tangent to a circle passing through a point

Dynamically place 1 circle and 1 point (data)

Figure 18. Buttons accessing the different functions.

This paper focuses on eight buttons that allow one to deal with the following cases: point—circle (6 and 7), two external circles (10–13) and two secant circles (14 and 15). Button 14 depicts the zone containing solutions of external tangents to the two circles, which allows one to distinguish the zone encompassing centres of external tangents and the hyperbolic segment containing no solutions to be discarded.

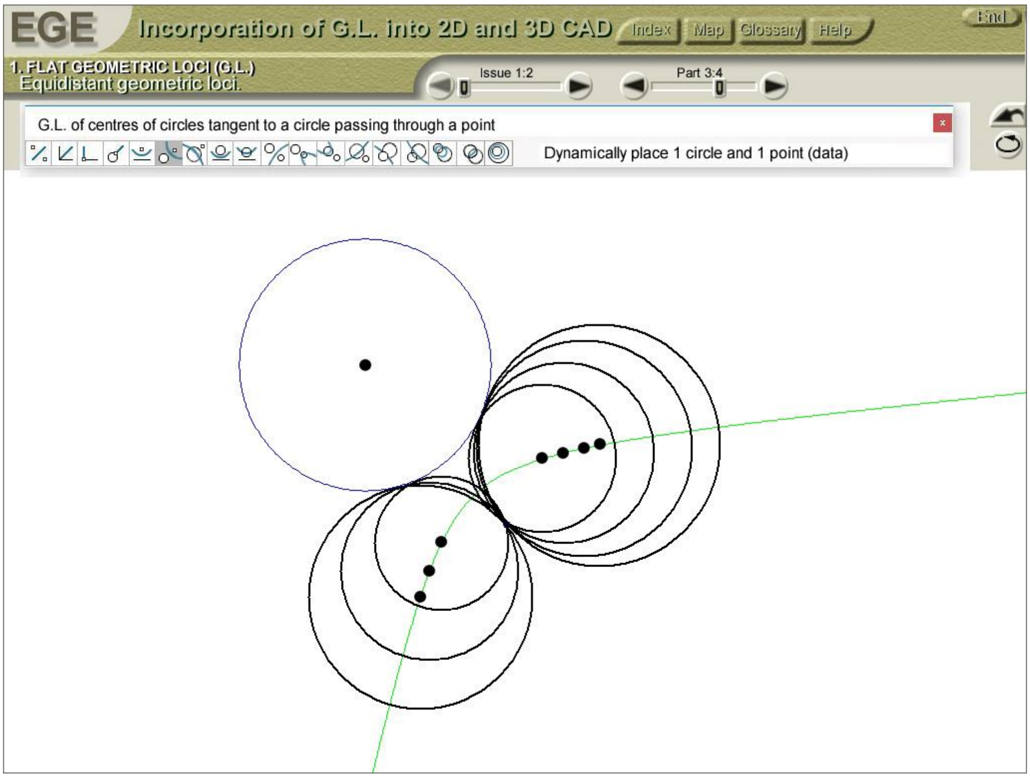
The first case to be displayed is that of the minimum, equal distance to a circle and a point (button 6, shaded in Figure 18), which allows the interactive introduction of the blue circle and the point (Figures 19 and 20).


EGE
Incorporation of G.L. into 2D and 3D CAD
Index
Map
Glossary
Help
End
1. FLAT GEOMETRIC LOCI (G.L.)
Equidistant geometric loci.
Issue 1:2
Part 3:4

G.L. of centres of circles tangent to a circle passing through a point

Dynamically place 1 circle and 1 point (data)

Figure 19. Output circle with its centre lying in the curve.

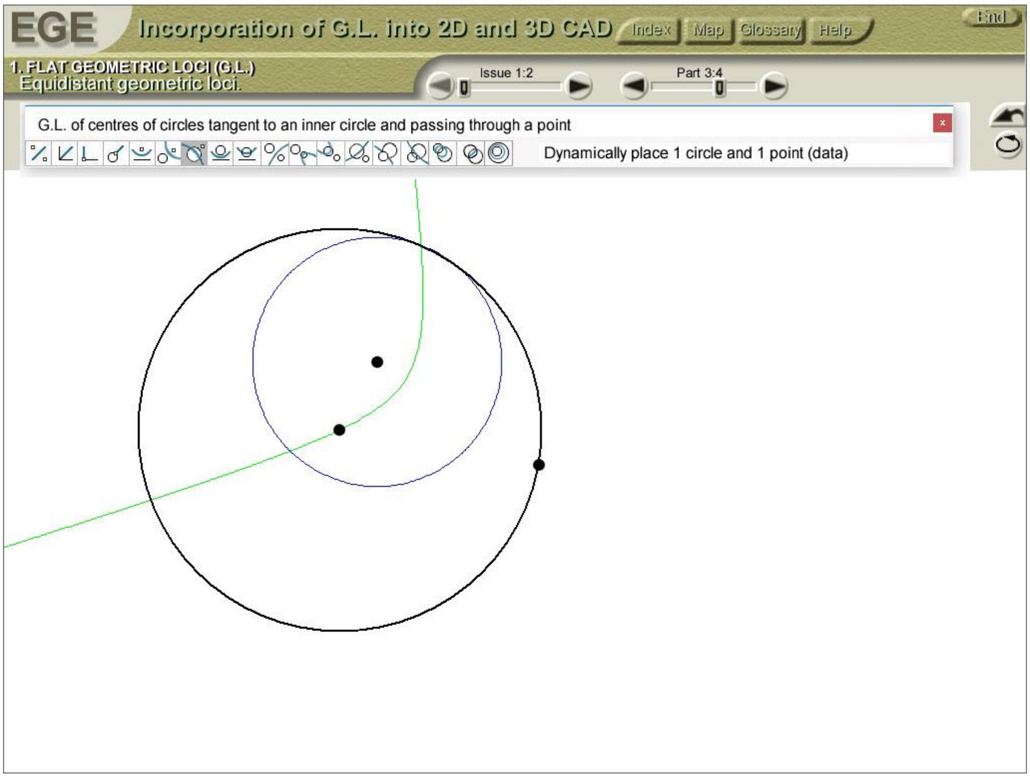

EGE
Incorporation of G.L. into 2D and 3D CAD
Index
Map
Glossary
Help
End
1. FLAT GEOMETRIC LOCI (G.L.)
Equidistant geometric loci.
Issue 1:2
Part 3:4

G.L. of centres of circles tangent to a circle passing through a point

Dynamically place 1 circle and 1 point (data)

Figure 20. Selected output circles with their centres lying in the curve.

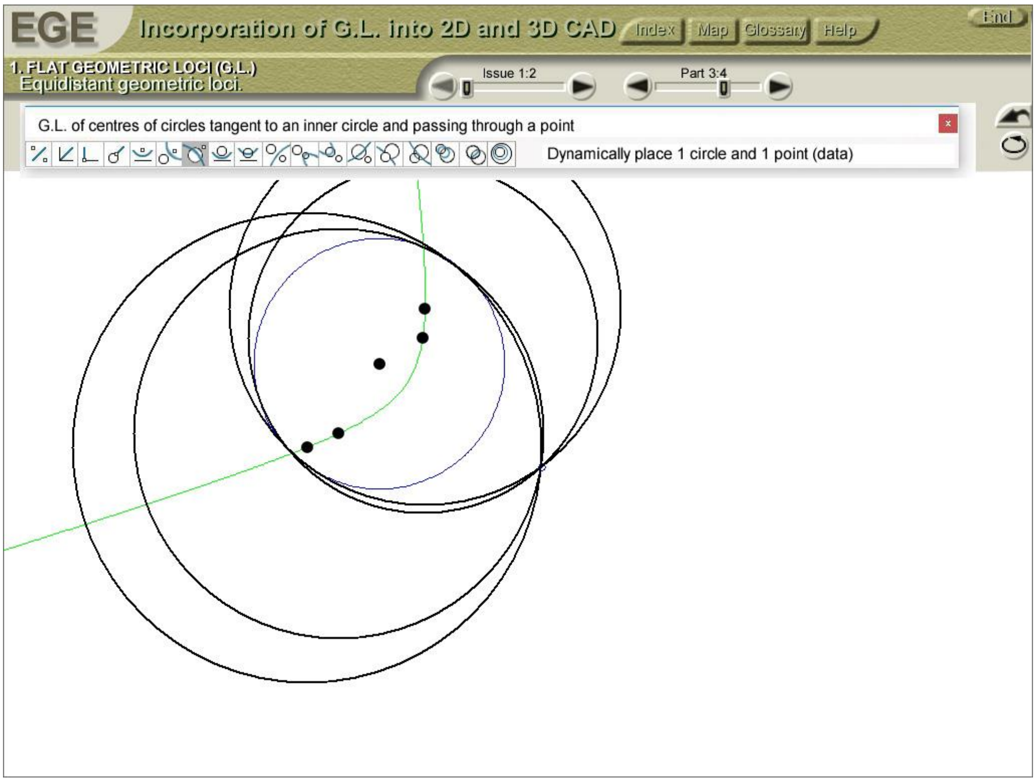
Figures 21 and 22 show the cases of a circle and a point with internal tangents to the given circle, and Figures 23 and 24 show those of two external circles.


EGE
Incorporation of G.L. into 2D and 3D CAD
Index
Map
Glossary
Help
End
1. FLAT GEOMETRIC LOCI (G.L.)
Equidistant geometric loci.
Issue 1:2
Part 3:4

G.L. of centres of circles tangent to an inner circle and passing through a point

Dynamically place 1 circle and 1 point (data)

Figure 21. Output circle with its centre lying in the curve.


EGE
Incorporation of G.L. into 2D and 3D CAD
Index
Map
Glossary
Help
End
1. FLAT GEOMETRIC LOCI (G.L.)
Equidistant geometric loci.
Issue 1:2
Part 3:4

G.L. of centres of circles tangent to an inner circle and passing through a point

Dynamically place 1 circle and 1 point (data)

Figure 22. Selected output circles with their centres lying in the curve.

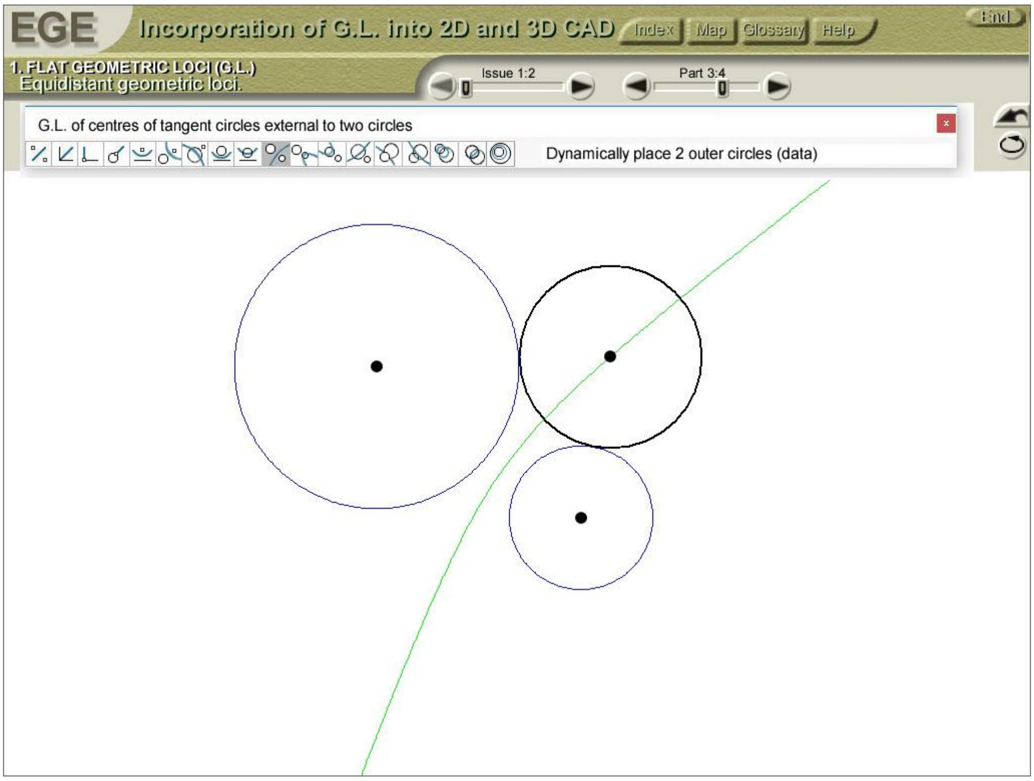

EGE
Incorporation of G.L. into 2D and 3D CAD
Index
Map
Glossary
Help
End
1. FLAT GEOMETRIC LOCI (G.L.)
Equidistant geometric loci.
Issue 1:2
Part 3:4
G.L. of centres of tangent circles external to two circles
Dynamically place 2 outer circles (data)

Figure 23. Geometric locus of the centres of circles that are external tangents to the two given circles.

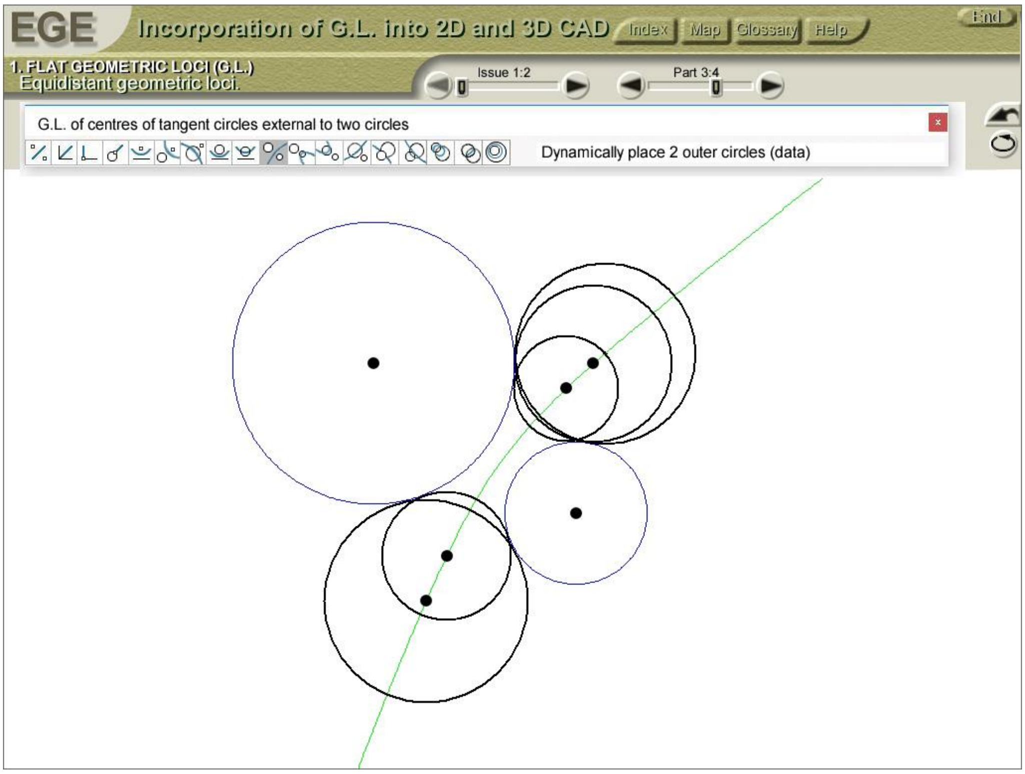
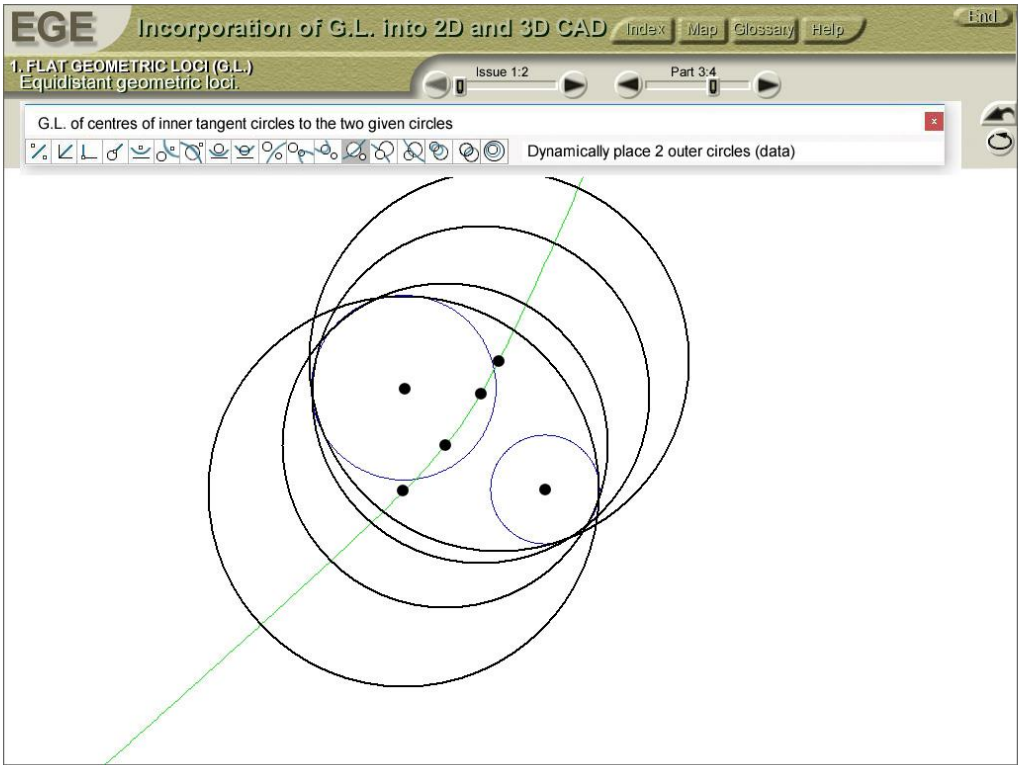

EGE
Incorporation of G.L. into 2D and 3D CAD
Index
Map
Glossary
Help
End
1. FLAT GEOMETRIC LOCI (G.L.)
Equidistant geometric loci.
Issue 1:2
Part 3:4
G.L. of centres of tangent circles external to two circles
Dynamically place 2 outer circles (data)

Figure 24. Selected output circles with their centres lying in the curve.

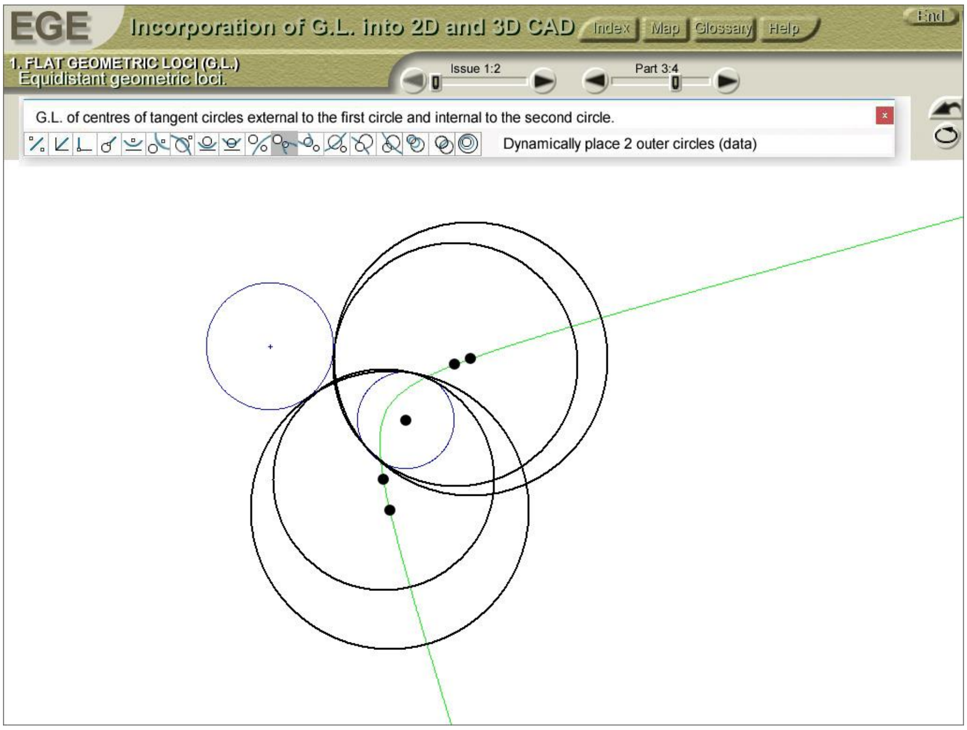
Finally, Figure 25 shows the case of internal tangents to two inscribed circles; Figure 26 that of tangent circles outside the first given circle and inside the second; and Figure 27 the opposite situation.


EGE
Incorporation of G.L. into 2D and 3D CAD
Index
Map
Glossary
Help
End
1. FLAT GEOMETRIC LOCI (G.L.)
Equidistant geometric loci.
Issue 1:2
Part 3:4
x

G.L. of centres of inner tangent circles to the two given circles

Dynamically place 2 outer circles (data)

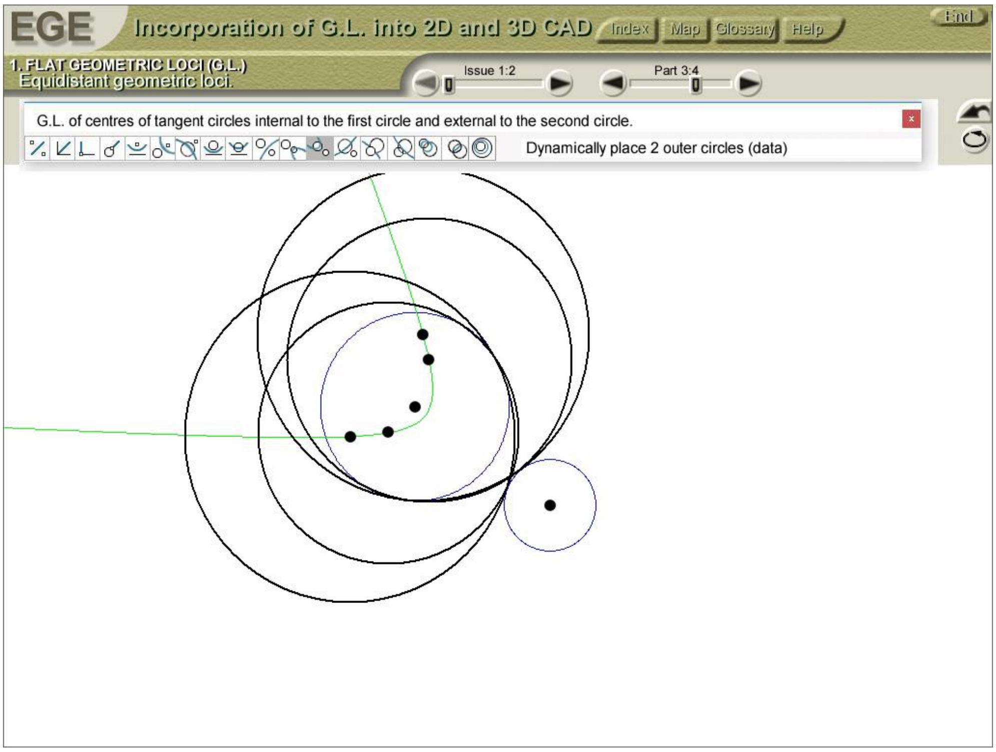
Figure 25. Geometric locus of the centres of circles that are internal tangents to two mutually external circles.


EGE
Incorporation of G.L. into 2D and 3D CAD
Index
Map
Glossary
Help
End
1. FLAT GEOMETRIC LOCI (G.L.)
Equidistant geometric loci.
Issue 1:2
Part 3:4
x

G.L. of centres of tangent circles external to the first circle and internal to the second circle.

Dynamically place 2 outer circles (data)

Figure 26. Geometric locus of the centres of circles that are external tangents to the first given circle and internal tangents to the second.



EGE
Incorporation of G.L. into 2D and 3D CAD
Index
Map
Glossary
Help
End
1. FLAT GEOMETRIC LOCI (G.L.)
Equidistant geometric loci.
Issue 1:2
Part 3:4
×

G.L. of centres of tangent circles internal to the first circle and external to the second circle.

Dynamically place 2 outer circles (data)

Figure 27. Geometric locus of the centres of circles that are internal tangents to the first given circle and external tangents to the second.

5. Conclusions

This paper features a software tool that allows the auxiliary curve containing all points which fulfil metric conditions at all possible positions (i.e., external, internal, secant or tangent), both with a whole hyperbola or only those hyperbolic segments where a solution exists, to be obtained with minimal knowledge of the geometric loci relevant to the equidistance from two circles. The solution to this problem was extended to the point−circle case in order to demonstrate that it makes sense to assume the point to be a circle of zero radius.

The software is based on geometric concepts and uses appropriate algorithms to control the curves from the relative positions of two circles or a circle and a point. The algorithms used for this purpose can be easily implemented in CAD software. The basic concepts involved are as follows:

- The maximum and minimum point-to-circle distance needed to obtain the equations of the curves corresponding to different relative positions.
- The boundary conditions needed to determine the boundaries of the zones where the hyperbolic solutions lie, discard those curve segments not fulfilling the requirements and draw the segments that do fulfil them.

This innovative work can open up new avenues for dealing with other theoretically simpler but at least as useful works, which should also be checked for performance.

Author Contributions: D.H.-D. was responsible for carrying out the conceptualization, investigation and writing of the original draft of the manuscript. F.H.-A. developed the new software, calculations and data curation. V.H.-A. performed the software tests and the software validation. R.V.-R. performed the software tests and image editing. F.J. and J.I.R.-S. conceptualized, designed the experimental methodology and coordinated and supervised the entire study. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not applicable.

Acknowledgments: We thank the anonymous reviewers of this paper for their constructive suggestions and comments.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

This section shows the programming code (Figures A1–A19) that allows one to discard a portion of the hyperbola in the case of two secant circles and the geometric locus of the centres of the circles that are external tangents to both—obviously, only a segment of the curve will fulfil the conditions since the zone shared by the two circles, where tangent circles are inevitably inscribed, contains no solutions.

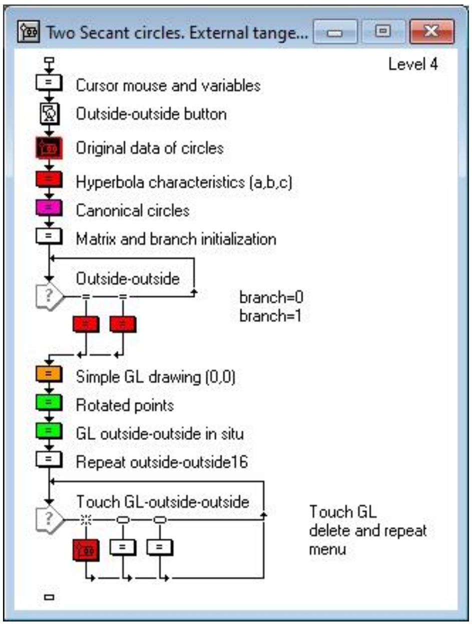
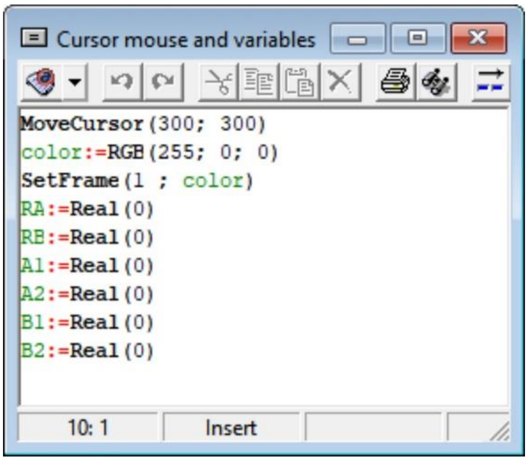


Figure A1. Two secant circles. External tangents.



```
MoveCursor(300; 300)
color:=RGB(255; 0; 0)
SetFrame(1 ; color)
RA:=Real(0)
RB:=Real(0)
A1:=Real(0)
A2:=Real(0)
B1:=Real(0)
B2:=Real(0)
```

Figure A2. Cursor mouse and variables.

Figure A3. Outside−outside button.

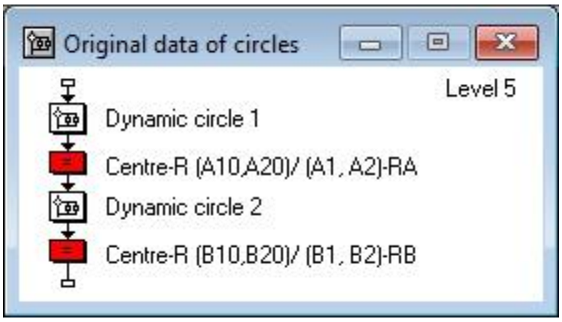


Figure A4. Original data of circles.

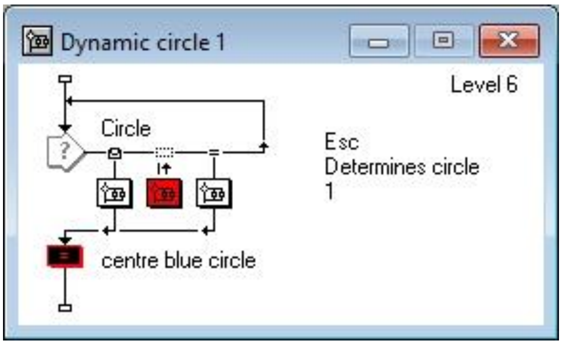
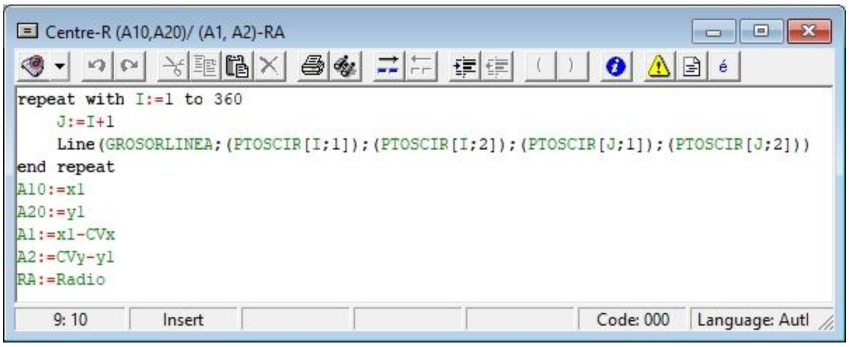


Figure A5. Dynamic circle 1.



```
repeat with I:=1 to 360
    J:=I+1
    Line(GROSORLINEA;(PTOSCIR[I;1]);(PTOSCIR[I;2]);(PTOSCIR[J;1]);(PTOSCIR[J;2]))
end repeat
A10:=x1
A20:=y1
A1:=x1-CVx
A2:=CVy-y1
RA:=Radio
```

Figure A6. Draw circle 1.

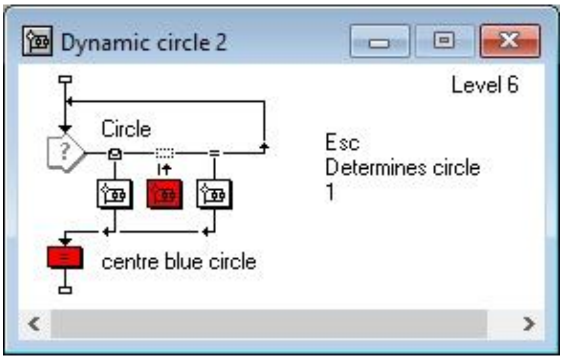
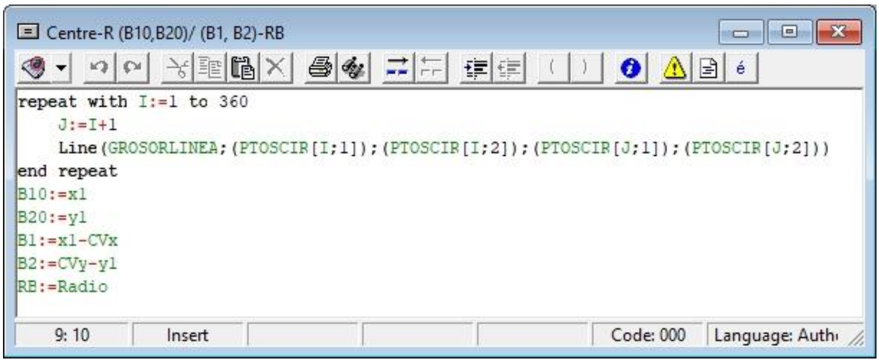
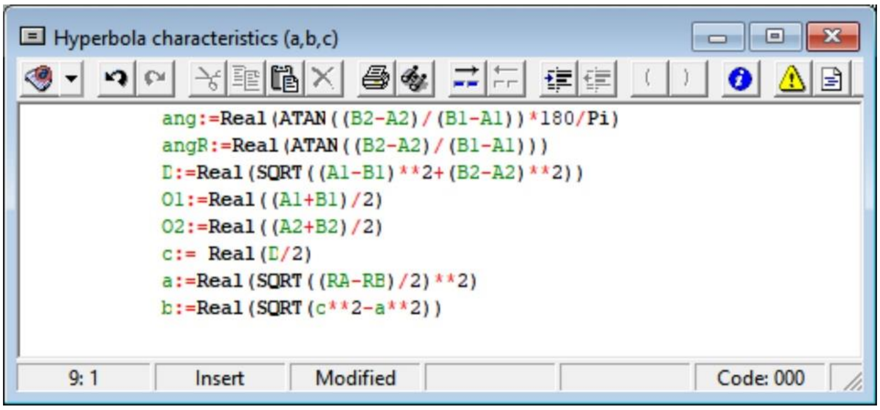


Figure A7. Dynamic circle 2.



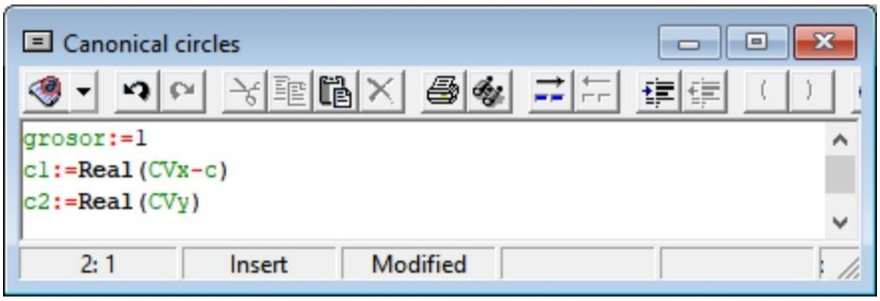
```
repeat with I:=1 to 360
    J:=I+1
    Line(GROSORLINEA;(PTOSCIR[I;1]);(PTOSCIR[I;2]);(PTOSCIR[J;1]);(PTOSCIR[J;2]))
end repeat
B10:=x1
B20:=y1
B1:=x1-CVx
B2:=CVy-y1
RB:=Radio
```

Figure A8. Draw circle 2.



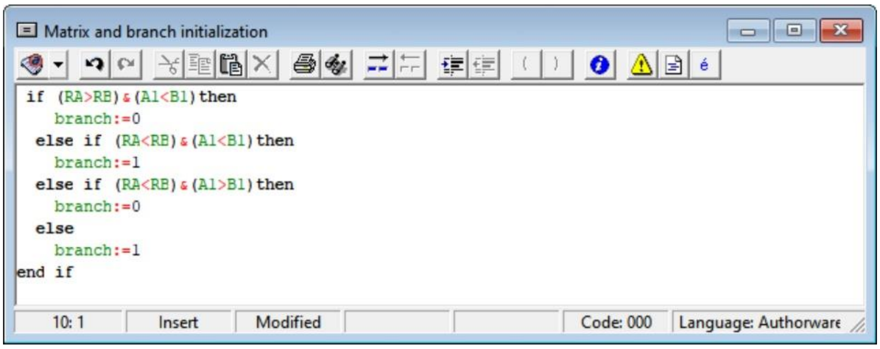
```
              ang:=Real(ATAN((B2-A2)/(B1-A1))*180/Pi)
              angR:=Real(ATAN((B2-A2)/(B1-A1)))
              D:=Real(SQRT((A1-B1)**2+(B2-A2)**2))
              O1:=Real((A1+B1)/2)
              O2:=Real((A2+B2)/2)
              c:= Real(D/2)
              a:=Real(SQRT((RA-RB)/2)**2)
              b:=Real(SQRT(c**2-a**2))
```

Figure A9. Characteristics of the hyperbola.



```
grosor:=1
c1:=Real(CVx-c)
c2:=Real(CVy)
```

Figure A10. Canonical circles.



```
 if  (RA>RB)&(A1<B1)then
     branch:=0
  else if  (RA<RB)&(A1<B1)then
     branch:=1
  else if  (RA<RB)&(A1>B1)then
     branch:=0
  else
     branch:=1
end if
```

Figure A11. Matrix and branch initialization.

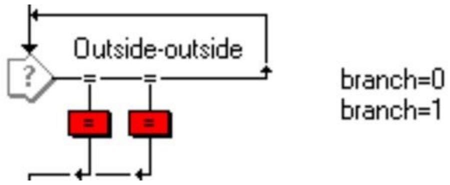
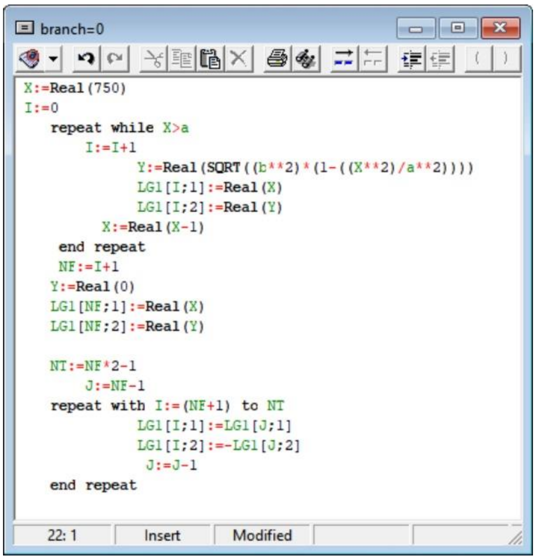
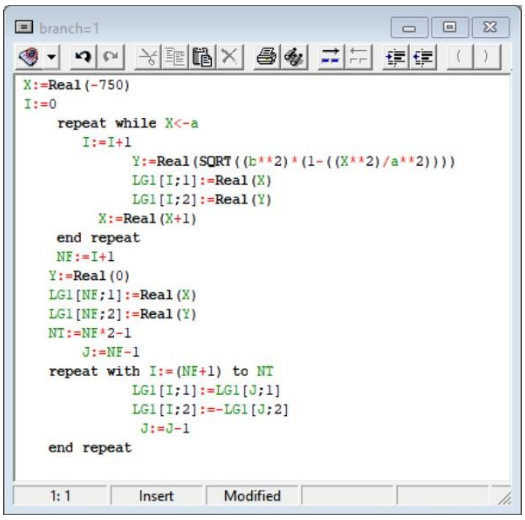


Figure A12. Loop branch selection.



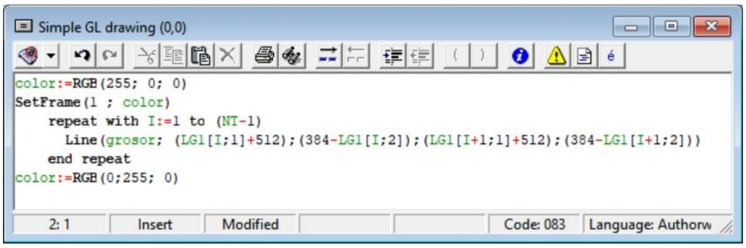
```
X:=Real(750)
I:=0
   repeat while X>a
       I:=I+1
             Y:=Real(SQRT((b**2)*(1-((X**2)/a**2))))
             LG1[I;1]:=Real(X)
             LG1[I;2]:=Real(Y)
         X:=Real(X-1)
    end repeat
    NF:=I+1
   Y:=Real(0)
   LG1[NF;1]:=Real(X)
   LG1[NF;2]:=Real(Y)

   NT:=NF*2-1
       J:=NF-1
   repeat with I:=(NF+1) to NT
             LG1[I;1]:=LG1[J;1]
             LG1[I;2]:=-LG1[J;2]
              J:=J-1
   end repeat
```

Figure A13. Branch 0.



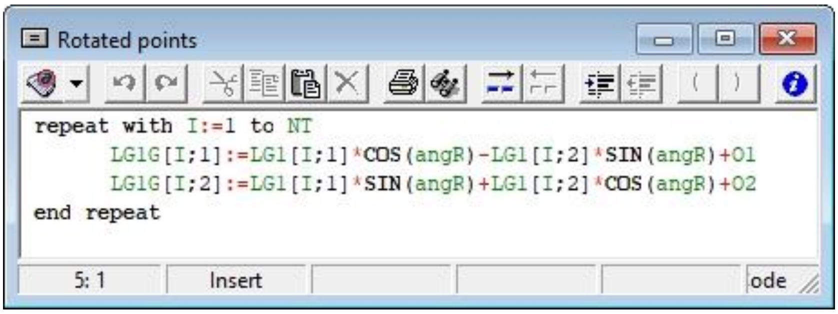
```
X:=Real(-750)
I:=0
    repeat while X<-a
       I:=I+1
             Y:=Real(SQRT((b**2)*(1-((X**2)/a**2))))
             LG1[I;1]:=Real(X)
             LG1[I;2]:=Real(Y)
         X:=Real(X+1)
    end repeat
    NF:=I+1
   Y:=Real(0)
   LG1[NF;1]:=Real(X)
   LG1[NF;2]:=Real(Y)
   NT:=NF*2-1
       J:=NF-1
   repeat with I:=(NF+1) to NT
             LG1[I;1]:=LG1[J;1]
             LG1[I;2]:=-LG1[J;2]
              J:=J-1
   end repeat
```

Figure A14. Branch 1.



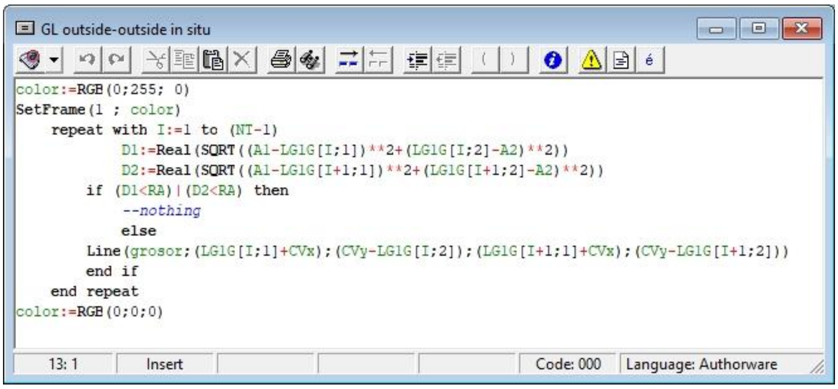
```
color:=RGB(255; 0; 0)
SetFrame(1 ; color)
    repeat with I:=1 to (NT-1)
      Line(grosor; (LG1[I;1]+512);(384-LG1[I;2]);(LG1[I+1;1]+512);(384-LG1[I+1;2]))
    end repeat
color:=RGB(0;255; 0)
```

Figure A15. Visual checking of geometric loci.



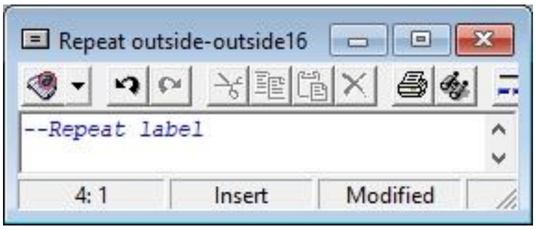
```
repeat with I:=1 to NT
      LG1G[I;1]:=LG1[I;1]*COS(angR)-LG1[I;2]*SIN(angR)+O1
      LG1G[I;2]:=LG1[I;1]*SIN(angR)+LG1[I;2]*COS(angR)+O2
end repeat
```

Figure A16. Coordinates of rotated points.



```
color:=RGB(0;255; 0)
SetFrame(1 ; color)
   repeat with I:=1 to (NT-1)
         D1:=Real(SQRT((A1-LG1G[I;1])**2+(LG1G[I;2]-A2)**2))
         D2:=Real(SQRT((A1-LG1G[I+1;1])**2+(LG1G[I+1;2]-A2)**2))
      if (D1<RA)|(D2<RA) then
         --nothing
         else
      Line(grosor;(LG1G[I;1]+CVx);(CVy-LG1G[I;2]);(LG1G[I+1;1]+CVx);(CVy-LG1G[I+1;2]))
      end if
   end repeat
color:=RGB(0;0;0)
```

Figure A17. In situ generation of outside–outside geometric loci.



```
--Repeat label
```

Figure A18. Repeat label.

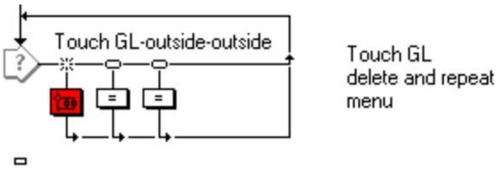


Figure A19. Decision loop.

References

1. Cummins, C.; Seale, M.; Macente, A.; Certini, D.; Mastropaolo, E.; Viola, I.M.; Nakayama, N. A separated vortex ring underlies the flight of the dandelion. *Nature* **2018**, *562*, 414–418. [CrossRef] [PubMed]
2. Rojas-Sola, J.I. Advances in engineering graphics: Improvements and new proposals. *Symmetry* **2021**, *13*, 827. [CrossRef]
3. Llach, D.C. Reconstructing "sketchpad" and the "coons patch": Toward an archaeology of CAD. *Leonardo* **2018**, *51*, 429–430.
4. Rojas-Sola, J.I.; Del Río-Cidoncha, G.; Ortíz-Marín, R.; López-Pedregal, J.M. Design and development of sheet-metal elbows using programming with visual basic for applications in CATIA. *Symmetry* **2020**, *13*, 33. [CrossRef]
5. Pan, Z.; Wang, X.; Teng, R.; Cao, X. Computer-aided design-while-engineering technology in top-down modeling of mechanical product. *Comput. Ind.* **2016**, *75*, 151–161. [CrossRef]
6. Wang, H.; Zeng, Y.; Li, E.; Huang, G.; Gao, G.; Li, G. "Seen Is Solution" a CAD/CAE integrated parallel reanalysis design system. *Comput. Methods Appl. Mech. Eng.* **2016**, *299*, 187–214. [CrossRef]
7. Bonneau, G.P.; Bartoň, M.; Nelaturi, S. SPM 2021 Editorial. *Comput. Aided Des.* **2022**, *143*, 103138. [CrossRef]
8. Chang, Y.S.; Chen, M.Y.C.; Chuang, M.J.; Chou, C.H. Improving creative self-efficacy and performance through computer-aided design application. *Think. Ski. Creat.* **2019**, *31*, 103–111. [CrossRef]

9. Flores, R.L.; Belaud, J.P.; Negny, S.; Le Lann, J.M. Open computer aided innovation to promote innovation in process engineering. *Chem. Eng. Res. Des.* **2015**, *103*, 90–107. [CrossRef]
10. Husig, S.; Kohn, S. Computer aided innovation—State of the art from a new product development perspective. *Comput. Ind.* **2009**, *60*, 551–562. [CrossRef]
11. Zanni-Merk, C.; Cavallucci, D.; Rousselot, F. An ontological basis for computer aided innovation. *Comput. Ind.* **2009**, *60*, 563–574. [CrossRef]
12. Li, J.; Li, Y. Research and application of computer aided design system for product innovation. *J. Comput. Methods Sci. Eng.* **2019**, *19*, S41–S46. [CrossRef]
13. Bonnici, A.; Akman, A.; Calleja, G.; Camilleri, K.P.; Fehling, P.; Ferreira, A.; Hermuth, F.; Israel, J.H.; Landwehr, T.; Liu, J.; et al. Sketch-based interaction and modeling: Where do we stand? *AI EDAM-Artif. Intell. Eng.Des. Anal. Manuf.* **2019**, *33*, 370–388. [CrossRef]
14. Bobenrieth, C.; Cordier, F.; Habibi, A.; Seo, H. Descriptive: Interactive 3D shape modeling from a single descriptive sketch. *Comput. Aided Des.* **2020**, *128*, 102904. [CrossRef]
15. Gryaditskaya, Y.; Sypesteyn, M.; Hoftijzer, J.W.; Pont, S.; Durand, F.; Bousseau, A. OpenSketch: A richly-annotated dataset of product design sketches. *ACM Trans. Graph.* **2019**, *38*, 232. [CrossRef]
16. Sun, L.; Xiang, W.; Chai, C.; Yang, Z.; Zhang, K. Designers' perception during sketching: An examination of creative segment theory using eye movements. *Des. Stud.* **2014**, *35*, 593–613. [CrossRef]
17. Sun, L.; Xiang, W.; Chai, C.; Wang, C.; Huang, Q. Creative Segment: A descriptive theory applied to computer-aided sketching. *Des. Stud.* **2014**, *35*, 54–79. [CrossRef]
18. Veisz, D.; Namouz, E.Z.; Joshi, S.; Summers, J.D. Computer-aided design versus sketching: An exploratory case study. *AI EDAM-Artif. Intell. Eng. Des. Anal. Manuf.* **2012**, *26*, 317–335. [CrossRef]
19. Willis, K.D.D.; Jayaraman, P.K.; Lambourne, J.G.; Chu, H.; Pu, Y.W. Engineering sketch generation for computer-aided design. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Nashville, TN, USA, 19–25 June 2021; pp. 2105–2114. [CrossRef]
20. Zhong, Y.; Qi, Y.G.; Gryaditskaya, Y.; Zhang, H.G.; Song, Y.Z. Towards practical sketch-based 3D shape generation: The role of professional sketches. *IEEE Trans. Circuits Syst. Video Technol.* **2021**, *31*, 3518–3528. [CrossRef]
21. Velichová, D. Geometry in engineering education. *Eur. J. Eng. Educ.* **2002**, *27*, 289–296. [CrossRef]
22. Voronina, M.V.; Tretyakova, Z.O.; Moroz, O.N.; Folomkin, A.I. Descriptive geometry in educational process of technical university in Russia today. *Int. J. Environ. Sci. Educ.* **2016**, *11*, 10911–10922.
23. Ullah, A.M.M.S.; Harib, K.H. Tutorials for integrating CAD/CAM in engineering curricula. *Educ. Sci.* **2018**, *8*, 151. [CrossRef]
24. Liapi, K.A. Geometry in architectural engineering education revisited. *J. Archit. Eng.* **2002**, *8*, 80–88. [CrossRef]
25. Ranger, F.; Vezeau, S.; Lortie, M. Traditional product representations and new digital tools in the dimensioning activity: A designers' point of view on difficulties and needs. *Des. J.* **2018**, *21*, 707–730. [CrossRef]
26. Papachristou, E.; Kyratsis, P.; Bilalis, N. A comparative study of open-source and licensed CAD software to support garment development learning. *Machines* **2019**, *7*, 30. [CrossRef]
27. Blažek, J.; Pech, P. Locus Computation in Dynamic Geometry Environment. *Math. Comput. Sci.* **2019**, *13*, 31–40. [CrossRef]
28. Kovács, Z. Achievements and Challenges in Automatic Locus and Envelope Animations in Dynamic Geometry. *Math. Comput. Sci.* **2019**, *13*, 131–141. [CrossRef]
29. Rojas-Sola, J.I.; Hernandez-Diaz, D.; Villar-Ribera, R.; Hernandez-Abad, V.; Hernandez-Abad, F. Computer-Aided Sketching: Incorporating the locus to improve the three-dimensional geometric design. *Symmetry* **2020**, *12*, 1181. [CrossRef]
30. Hernandez-Abad, F.; Hernandez-Abad, V.; Ochoa-Vives, M. *Lugares Geométricos: Su Aplicación a Tangencias*; UPC: Barcelona, Spain, 1993. (In Spanish)
31. Hernández-Abad, F.; Rojas-Sola, J.I.; Hernández-Abad, V.; Ochoa-Vives, M.; Font-Andreu, J.; Hernandez-Diaz, D.; Villar-Ribera, R. Educational software to learn the essentials of engineering Graphics. *Comput. Appl. Eng. Educ.* **2012**, *20*, 1–18. [CrossRef]
32. Hernández-Abad, F.; Rojas-Sola, J.I.; Hernández-Abad, V.; Ochoa-Vives, M.; Font-Andreu, J.; Hernandez-Diaz, D.; Villar-Ribera, R. Interactive educational software of textile design. *Comput. Appl. Eng. Educ.* **2012**, *20*, 161–174. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.