A Classification of Tritangent Conics: The Power of Geometric Macros in Dynamic Geometry

Jean-Jacques Dahan
jjdahan@wanadoo.fr
IRES of Toulouse
Paul Sabatier University
Toulouse FRANCE

Abstract: *Based on our knowledge of conics and my previous work, I will detail the algorithms for constructing conics tangent to the three sides of a triangle, internally and externally. These constructions developed in a dynamic geometry environment (here the new Cabri) largely using the "Macro Construction" tool (which is none other than a program of this environment) will make it possible to visualize all these conics in motion and to highlight evidence of some surprising properties of these families of conics: in particular, we will be led to conjecture a classification of conics tangent to the three sides of a triangle according to the position of one of their foci. This work requires for each type of conic an introduction concerning the construction algorithms of their characteristic elements as well as of their tangents lines within a dynamic geometry environment.*

1. First work around ellipses

1.1. Construction algorithms of the center, axes and foci of an ellipse

For an ellipse given in a dynamic geometry environment, here are the different constructions needed to obtain:

1.1.1. The center of the ellipse (Figure 1 left): (E) is a given ellipse and A, B and C are three points of this ellipse. An algorithm to construct the center O of this ellipse is as follows:

1. Construct line (AB)
2. Construct line (CD) parallel to (AB) through C
3. Line (IJ) (I and J midpoints of [AB] and [CD]
4. Construct the points L and R intersection points between (E) and (IJ)
5. construct O midpoint of [LR] which is the center of ellipse (E)

This construction is recorded as a macro construction called **center ellhyp** for which the initial object is an ellipse (or a hyperbola) and the final object is the center of the given conic.

1.1.2. **The axes of the ellipse** (Figure 1 center): (E) is a given ellipse and the previous macro **center ellhyp** is available. An algorithm to construct the axes of this ellipse is as follows:

1. Construct center O of ellipse (E) in using the macro **center ellhyp**
2. Select any points $r1$ and $r2$ on (E)
3. Evaluate the distance between $r1$ and $r2$: d
4. Display number 90 as the result of 90 $* \frac{d+1}{d+1}$
5. Ellipse (E') image of ellipse (E) with rotation centered at O and which angle is the previous calculated number 90
6. Points $i1$, $i2$ and $i3$ *three first intersection points between* (E) and (E')
7. $j1$ and $j2$, midpoints of [$i1$ $i2$] and [$i2$ $i3$]
8. Lines *(O $j1$)* and *(O $j2$)* are the axes of ellipse (E)

This construction is recorded as a macro construction called **axes ellipse** with the initial object an ellipse and the final objects the axes of that given ellipse. Note that this construction does not work to display the axes of a hyperbola.

1.1.3. **The two foci of the ellipse** (Figure 1 right): (E) is a given ellipse and the previous macros **center ellhyp** and **axes ellipse** are available. An algorithm to construct the axes of this ellipse is as follows:

1. Construct center O of ellipse (E) in using the macro **center ellhyp** 2. Construct the axes of ellipse (E) in using the macro **axes ellipse** 3. Create points $s2$, $r2$, $t1$ and $t2$, intersection points between these axes and the ellipse 4. Display distances $Os2$ (a) and $Ot2$ (b) and evaluate $c = \sqrt{c^2}$ where $c^2 = \sqrt{(a^2 - b^2)^2}$	5. Create triangle $Os2t2$ and m midpoint of [$s2t2$] 6. Create point n intersection between the triangle $Os2t2$ and the perpendicular to $s2t2$ at m 7. Create vector On 8. Measurement transfer of c (as calculated in the table below) to get point $f1$ which is a focus of (E) 9. $f2$, symmetric point of $f1$ with respect to O is the second focus

This construction is recorded as a macro construction called **foci ellipse** with initial object an ellipse and final objects the foci of that given ellipse.

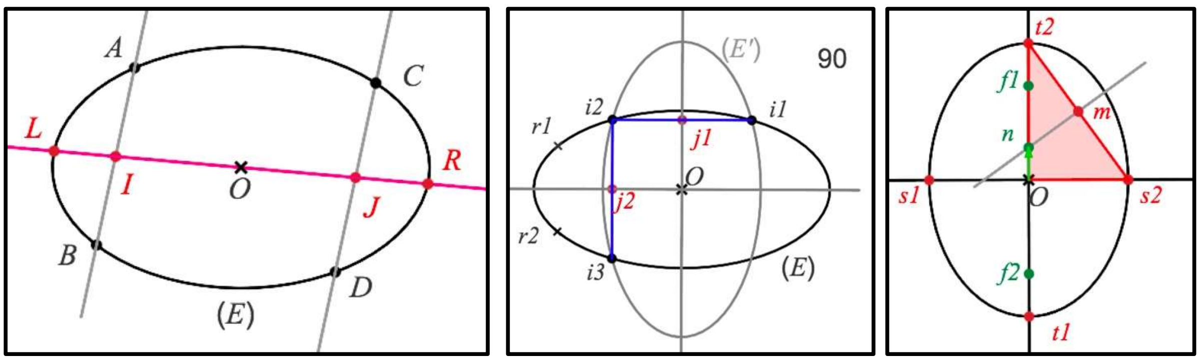

Figure 1: Center, axes and foci of an ellipse

1.2. Construction algorithm of the tangent lines to an ellipse from a given point (Figure 2 left)**:** (E) is a given ellipse and the previous macro **axes ellipse** is available. M is a given point outside the ellipse. An algorithm to construct the two tangent lines to this ellipse passing through M is as follows:

1. Construct the two foci $f1$ and $f2$ of ellipse (E) in using the macro **foci ellipse** 2. Construct line ($f1f2$) and points $e1$ and $e2$ its intersection points with (E) 3. Create segment [$e1$ $e2$] and cercle ($C1$) centered at $f2$ of radius $e1e2$ 4. Create circle (C) centered at M passing through $f1$	5. Create points $h1$ and $h2$, intersection points between ($C1$) and (C) 6. ($T1$) is the perpendicular bisector of [$f1h1$] 7. ($T2$) is the perpendicular bisector of [$f1h2$] 8. $t1$ is the intersection point between ($T1$) and segment[$f2h1$] 9. $t2$ is the intersection point between ($T2$) and segment[$f2k$] where k is the symmetric point of $f2$ with respect to ($T2$)

This construction is recorded as a macro construction called **tangent lines ellipse** with initial objects an ellipse and a point and final objects the two tangent lines to the ellipse passing through the given point.

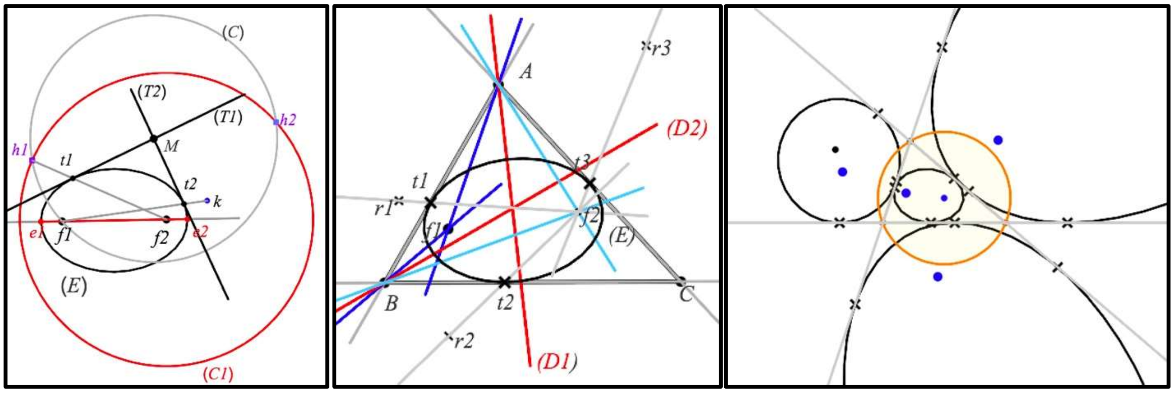

Figure 2: Tangent lines to an ellipse through a given point

1.3. Little Poncelet Theorem. Consequences

The Little Poncelet Theorem states that the angle bisector of *f1Mf2* (Figure 2 left) is also the angle bisector of the two tangent lines (*T1*) and (*T2*). Therefore, the angle bisector can be interpreted as an axis of symmetry. We can use this property for the following constructions.

1.3.1. Ellipses tangent to the three sides of a given triangle (Figure 2 center and right):
Here is an algorithm allowing such construction leading to a powerful macro construction

1. Construct the first focus *f1,* lines *(f1A)* and *(f1B),* the angle bisectors of *CAB* and *CBA, (D1)* and *(D2)* 2. Construct line (*f1A*) and line (*f1B*) 3. Symmetric line of *(f1A)* with respect to *(D1)* and symmetric line of *(f1B)* with respect to *(D2)* 4. Create *f2* their intersection point	5. Create points *r1*, *r2* and *r3* symmetric points of *f1* with respect to *(AB)*, *(BC)*) and *(CA)* 6. *t1* intersection point between (r*1f2*) and *(AB)* 7. *t2* intersection point between (r*2f2*) and *(BC)* 8. *t3* intersection point between (r*3f2*) and *(CA)* 9. *(E)* is the ellipse which foci are *f1* and *f2* passing through *t1*

This construction is recorded as a macro construction called **ellipse tritangent** with initial objects a triangle and a point and final objects the ellipse tangent to the three sides of the given triangle, a second point which is the second focus of this ellipse (the first focus being the first given point) and the three contact points with the sides (in reality the lines supporting the sides).
Important remark: The proposed algorithm works when the first point lies **inside the triangle** and also **outside the triangle but only if outside the circumcircle** of the triangle. Figure 2 right displays the use of this macro for a point inside the triangle and three points outside the triangle but inside the circumcircle
Another remark: it seems that it is impossible to construct an ellipse tangent to the three sides of a triangle when the first given focus lies outside the triangle and inside the circumcircle.

1.3.2. Ellipses tangent to two given rays:
We show now how to construct the ellipse tangent to two given rays, one given focus and a given contact point on one of the two given rays. Here is an algorithm allowing this construction (Figure 3 left):

1. Construct two rays (*L1*) and of (*L2*), a point *t1* on (*L1*) and a point *f1* 2. Angle bisector *(B)* of the two rays (*L1*) and (*L2*) 3. Symmetric line of *(Of1)* with respect to *(B): (S)* 4. Symmetric point of *f1* with respect to (*B*): *r1*	5. Intersection point between (*r1t1)* and *(S)*: *f2* 5. Symmetric point of *f2* with respect to (*L2)*: *r2* 7. Intersection point between (*f1r2)* and *(L2)*: *t2* 8. Ellipse *(E)* which foci are *f1* and *f2* passing through *t1*

This construction is recorded as a macro construction called **ellipse tgt to 2 rays** with initial objects two rays, a contact point on one ray and the first focus of the expected ellipse, and final objects the ellipse tangent to the two given rays, the second contact point and the second focus of the ellipse.
Figure 3 right displays three red ellipses constructed with this macro: the given focus is a random point between the two rays and the chosen contact point is a contact point of one of the ellipses constructed with the macro **ellipse tritangent**.

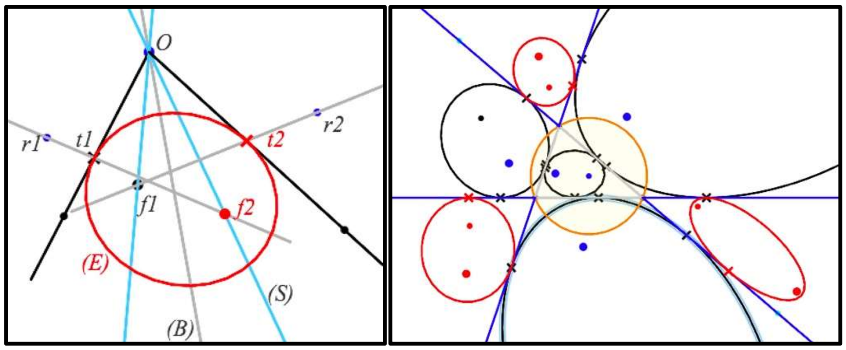

Figure 3: Ellipses tangent to two lines

1.4. Tangent line to an ellipse. Conjugate directions

In this paragraph, we show an algorithm to construct a tangent line to an ellipse at one of its points and then its conjugate directions (images by an affinity of two perpendicular diameters of a circle).
The first case when the ellipse is defined by two foci and one of its point (Figure 4 left): Here is the algorithm to construct a tangent line.

1.4.1. Tangent line to an ellipse

1. Construct line ((*f1f2)* and its intersection points with *(E):* *e1* and *e2*
2. Segment [*e1e2*]
3. Point *m* on *(E)*
5. Circle *(C)* centered at *f1* and of radius *e1e2*
6. Ray *[f1m)* intersecting *(C)* at *n*
7. *Line Tm* perpendicular bisector of [*f2 n*]

This construction is recorded as a macro construction called **tangent ellipse 1** with initial objects the two foci and the point defining the ellipse and the point of the ellipse where we expect the tangent line and final object this tangent line.
Second case when the ellipse is defined by five points: the algorithm construction is exactly the same as the previous one on the condition of adding a preliminary stage of construction of the foci of the ellipse in using the macro **foci ellipse**.
Then, this construction is recorded as a macro construction called **tangent ellipse 2** with initial objects the five points defining the ellipse, and the point of the ellipse where we expect the tangent line and final object this tangent line.
1.4.2. Conjugate directions (Figure 4 center): Given an ellipse defined by five points and a point *m1* on this ellipse, here is below is an algorithm to construct ray [*Om1*) and its conjugate [*Om2*)
This construction is recorded as a macro construction called **conjugate directions** with initial objects the five points defining the ellipse and a point *m1* of this ellipse and the final object [*Om1*) and its conjugate [*Om2*) where *m2* lies on the ellipse.

1. We use the macro "foci ellipse" to construct the two foci of *(E)*: *f1* and *f2*
2. Line ((*f1f2)* and one of its intersection points with *(E)*: *e2*
3. Midpoint *O* of [*f1f2*] and circle *(C)* centered at *O* and passing through *e2*
4. Point *m1* on *(E)*. **Ray *[Om1)***
5. Line perpendicular to *(f1f2)* through *m1* intersecting *(f1f2)* at *h1*
6. Ray *[h1m1)* intersecting *(C)* at *n1*
7. Evaluate distance *d* between *f1* and *f2*. Display number 90 as the evaluation of d-d+90
8. Rotation of *n1* around *O* (angle: the previous 90): *n2*. Ray [*On2*)
9. Line perpendicular to *(f1f2)* through *n2* intersecting *(f1f2)* at *h2*
10. Ray *[h2n2)* intersecting *(E)* at *m2*. **Ray [*Om2*)**

1.4.3. Parallelogram circumscribed to an ellipse defined by five points and a point *m1* on this ellipse (Figure 4 right). Here is an algorithm to construct the parallelogram circumscribed to the ellipse containing a point m1 and with sides parallel to [*Om1*) and its conjugate direction.

1. Create point *m1* on *(E)*.
2. Use the macro "**conjugate directions**" to construct the two conjugate directions *Om1* and *Om2*
3. Point *p1* symmetric of *m1* with respect to *O*
3. Point *p2* symmetric of *m2* with respect to *O*
4. Lines passing through *m1* and *p1* parallel to [*Om2)*
5. Lines passing through *m2* and *p2* parallel to [*Om1)*
6. Parallelogram defined by these four lines

This construction is recorded as a macro construction called **circum parallelogram** with initial objects the five points defining the ellipse and a point *m1* of this ellipse and final object the parallelogram circumscribed to the given ellipse, tangent at *m1*.

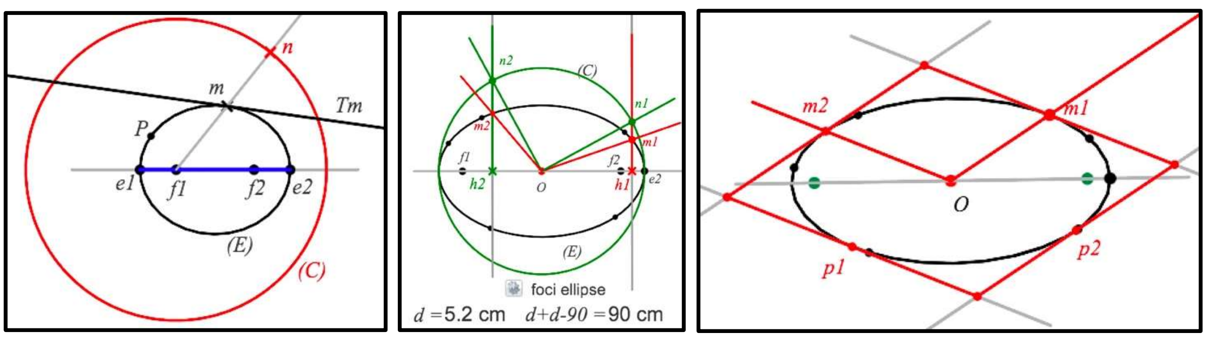

Figure 4: Tangent lines and conjugate directions of an ellipse

1.5. Steiner ellipse

1.5.1. Reminder of the construction of the Steiner ellipse (Figure 5 left)

The Steiner ellipse (*E*) of a triangle *ABC* is the image of the inscribed circle of an equilateral triangle, and justifies the algorithm of its construction detailed below:

1. Create medians [*AA1*], [*BB1*] and [*CC1*]
2. Their intersection point *I*
3. *j* midpoint of [*IA*] and *k* midpoint of [*IB*] and *l* midpoint of [*IC*]
3. Conic (E) passing through *A1, B1, C1, j,* and *k. (E)* is the Steiner ellipse passing also through *l* centered at *I*

This construction is recorded as a macro construction called **steiner ellipse** with initial object a triangle and the final objects the Steiner ellipse, its center (centroid of the given triangle) and the contact points (midpoints of the sides of the given triangle)

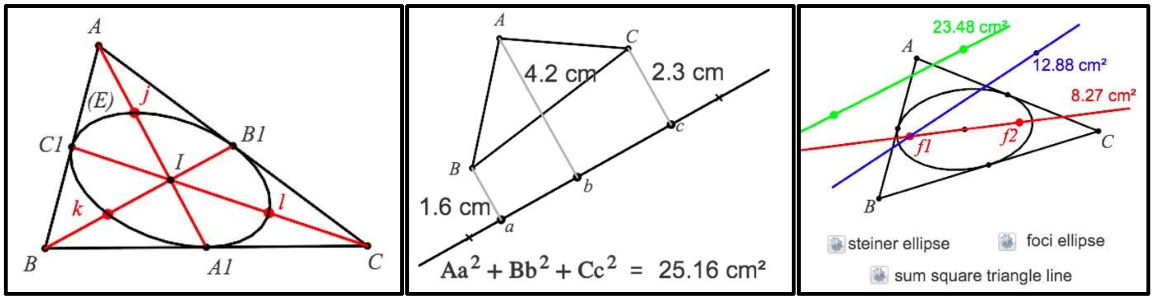

Figure 5: Steiner ellipse

1.5.2. A regression property of the Steiner ellipse

We create first a macro called **sum square triangle line** with initial objects a triangle and a line and the final object the number evaluating the sum of the squares of the distances between the vertices of the triangle and the given line (Figure 5 center).

Figure 5 (right) illustrates how it is possible to investigate in order to conjecture that the line joining the foci of the Steiner ellipse of the given triangle *ABC* is the one minimizing the sum of the squares of the distances between the vertices of *ABC* and a line. For the given triangle, use first the macro **steiner ellipse** to display the Steiner ellipse. Apply to this ellipse the macro **foci ellipse** to display its foci *f1* and *f2*. Create line (*f1f2*) to which we apply macro **sum square triangle line**: we obtain the number 8.27 cm^2. We create then a line with the number obtained with the same macro; Trying to decrease the value of this number in changing the position of the line leads to approach the position of (*f1f2*). This investigation can be conducted for any triangle and always leads to the same conjecture. In fact, the result stated by this conjecture is a known property of the Steiner ellipse.

1.6. Isoptic curves of an ellipse

1.6.1. Definition of an isoptic of an ellipse

Set of points from which an ellipse can be seen under a given angle.

1.6.2. Construction algorithm of the isoptics of an ellipse given by two foci, a point, and an angle beween 0° and 180°. Here is an algorithm for this construction:

1. Create a point *t1* on the given ellipse and the tangent line (*T1*) to the ellipse at *t1* in using the macro **tangent ellipse 1**	9. Create *Ray 1* = [*q*,*s1*)
2. Measurement of *f1f2* and display number 90 as the result of *f1f2 - f1f2 + 90*	10. *Ray 2* image of *Ray 1* by the translation mapping *q* onto *O*
3. *p* image *of O* (center ellipse) by the rotation centered at *t1* and of angle the previous calculated number *90*	11. *c1* intersection point between *Ray 2* and the ellipse
4. *p1,* orthogonal projection of *p* on *(T1)*	12. Hide all the previous constructions. Dont hide the ellipse, *t1*, (*T1*) and *c1*
5. *Create vector t1p1*	13. Create 5 points on the ellipse in order to use macro **conjugate directions** and obtain rays [*O,c1*) and [*O,c2*)
6. Point *q,* measurement transfer of *f1f2* on the previous vector from *t1*	14. (*T2*) is the parallel line to [*O,c1*) through *c2*
7. Create a slider whose boundaries are 0 and 180 commanding a number so so-called **angle**	15. *m* intersection point between (*T1*) and *(T2)*
8. *s1* image *of t1* by the rotation centered at *q* and of the angle the previous number **angle**	16. Measurement of angle *t1mc2* (equal to **angle**)
	17. Locus of point *m* when *t1* moves along the ellipse provides the isoptic related to the **angle** commanded by the slider

Stages from 1 to 11 are illustrated by Figure 6 left. Stages from 12 to 16 are illustrated by Figure 6 center. Stage 17 is illustrated by Figure 6 right.

Remark: in Figure 6 right, changing the value commanded by the slider automaticaly changes the shape of the isoptic. Especially, when **angle** is equal to 90°, we obtain a circle. To check experimentally this result, construct a conic passing through five points of the displayed isoptic, then the software recognizes a circle.

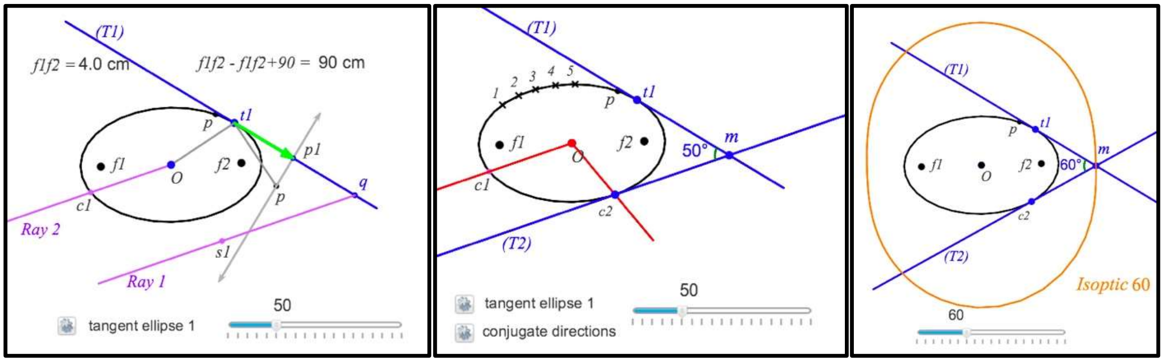

Figure 6: Isoptic curves of an ellipse

2. Second work around parabolas

2.1. Algorithm construction of the focus, the axis and the directrix of a parabola

Here is an algorithm for this construction (Figure 7 left):

1. Create three points *m1*, *m2* and *m3* on (*P*) and segment [*m1*, *m2*] 2. Construct *m4* such as *(m1 m2) // (m3 m4)* 3. *i1* midpoint of [*m1, m2*] and *i2* midpoint of [*m3, m4*] 4. Line (*i1 i2)* intersects *(P)* at *t* 5. *(T)* parallel line to [*m1, m2*] through *t* 6. Line (*i1, i2)* and its perpendicular (*Pe*) through *m3* cutting *(P)* at *m5*	7. Perpendicular bisector (*Pb*) of [*m3 m5*] 8. Line *(R)* symmetric line of (*i1 i2*) with respect to *(T)* 9. *f* (focus) of *(P)* intersection point of *(R)* and *(Pb)* 10. *s* intersection of *(P)* and *(Pb)* 11. *h* symmetric of *f* with respect to *s* 12. *(D)* (directrix of *(P)*) perpendicular to (*Pb)* at *h*

This construction is recorded as a macro construction called **param parab** with initial object a parabola and final objects the focus, the directrix, the summit and the symmetric point of the focus with respect to the summit.

2.2. Algorithm construction of the two tangent lines to a parabola through a given point

In Figure 7 center, a parabola *(P)* is given by its focus *f* and its summit *s*. The following algorithm describes how to obtain the two tangent lines *(T1)* and *(T2)* to *(P)* through the given point *m*. We also obtain the contact points *t1* and *t2* of the tangent lines with *(P)*.

This construction is recorded as a macro construction called **2 lines tg parab** with initial objects the points *f*, *s* and *m* and final objects the two tangent lines through *m* to the parabola with focus *f* and summit *s* and the two contact points.

1. Create parabola (*P*) with its focus *f* and its summit *s* 2. Create *m* outside *(P)* 3. Create circle *(C)* centered at *m* passing through *f* 4. Perpendicular line (D*)* to *(fs)* through *h* symmetric point of *f* with respect to *s*	5. *s1* and *s2* intersection points between *(C)* and *(D)* 6. *(P1)* perpendicular line to *(D)* through *s1,* cutting *(P)* at *t1* 7. *(P2)* perpendicular line to *(D)* through *s2,* cutting *(P)* at *t2* 8. *(T1)* perpendicular bisector of [*f s1]* 9. *(T2)* perpendicular bisector of [*f s2]*

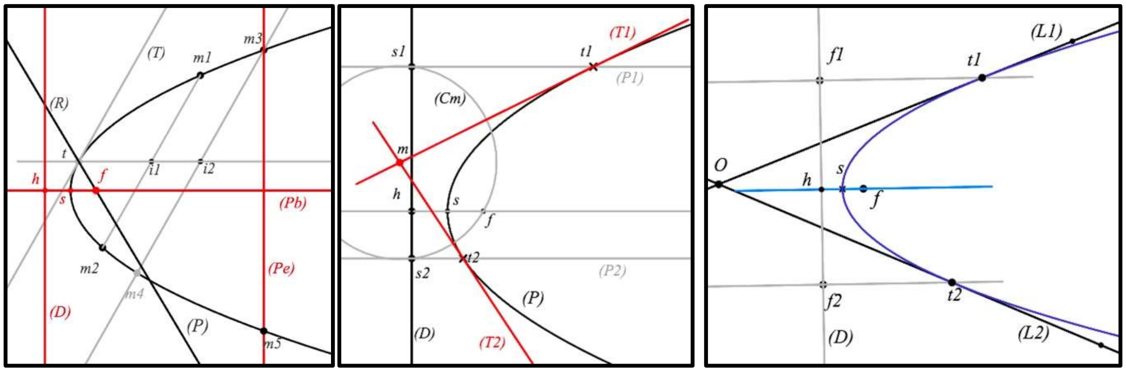

Figure 7: Parabolas

2.3. Construction algorithms of the parabolas tangent to two given lines (Figure 7 right)

Given two lines *(L1)* and *(L2)* crossing at *O* and a point *f* not lying on these lines, the following algorithm returns the parabola with focus *f* that is tangent to *(L1)* and *(L2)* at *t1* and *t2*.

1. Create lines *(L1)* and *(L2)* passing through *O* and a point *f*	6. *(Axis)* cuts *(D)* at *h*
2. *f1* symmetric of *f* with respect to *(L1)*	7. Construct *s* midpoint of [*h f*]
3. *f2* symmetric of *f* with respect to *(L2)*	8. Construct parabola *(P)* with *f* as a focus and *s* as a summit
4. Line (*D*) = *(f1f2)*	9. Perpendicular line to *(D)* at *f1* intersects *(L1)* at *t1*
5. Line *(Axis)* perpendicular line to *(D)* through *f*	10. Perpendicular line to *(D)* at *f2* intersects *(L1)* at *t2*

This construction is recorded as a macro construction called **parab bitg** with initial objects two lines *(L1)* and *(L2)* passing through point *O*, a point *f* and two points *t1* and *t2* on *(L1)* and *(L2),* and final objects the parabola with focus *f* tangent to these two lines at *t1* and *t2*.

2.4. Construction algorithm of the parabolas tangent to the three lines supporting the three sides of a triangle (relation with the circumcircle and the Simson and Steiner lines)

2.4.1. *Reminder* (Figure 8 left): Given a triangle *ABC*, its circumcircle *(C)* and a point *M*. Let us call *H1*, *H2,* and *H3* the orthogonal projections of *M* respectively on the three sides of the triangle and *M1*, *M2,* and *M3* the symmetric points of *M* with respect to the three sides of the triangle. We know this result ([5]):

Points *H1*, *H2,* and *H3* (respectively *M1*, *M2,* and *M3*) are colinear if and only if *M* belongs to *(C)*. In this case, the line joining *H1*, *H2,* and *H3* (respectively *M1*, *M2,* and *M3*) is called the **Simson line** of M (respectively the **Steiner line** of M) for the triangle *ABC*.

Remark: The Steiner line contains the orthocenter of the triangle

Creating Figure 8 left gives the opportunity to create two other macros:

Macro **Circumcircle** that returns the circumcircle of a triangle with its center.

Macro **Steiner line** that returns the Steiner line of a given triangle and a given point (chosen on the circumcircle).

2.4.2. The construction algorithm (Figure 8 center)

1. Create a point *f* on the circumcircle of the triangle *ABC* defined by the three given lines	6. *s* midpoint of [*fr*]
2. Create a triangle *ABC*	7. *f1, f2,* and *f3* symmetric of *f* with respect to the three given lines (on the Steiner line)
3. Use the macro **Steiner line** to obtain the Steiner line *(S)* of point *f* for *ABC*	8. Perpendicular line to *(S)* at *f1* cuts *(L1)* at *t1*
4. Perpendicular line *(Axis)* to *(S)* through *f*	9. Perpendicular line to *(S)* at *f2* cuts *(L2)* at *t2*
5. *r* intersection point between *(S)* and *(Axis)*	10. Perpendicular line to *(S)* at *f3* cuts *(L3)* at *t3*
	11. Parabola *(P)* which focus is *f* and summit *r*

This construction is recorded as a macro construction called **parab tg 3 lines** with initial objects three lines (defining a triangle) and a point on the circumcircle of the triangle and final objects the parabola tangent to these three lines, its summit, and the three contact points.
Another macro can also be recorded in changing the initial objects of the three lines with a triangle. In this case, the macro is called **parab tg triangle**.
Figure 8 right displays three parabolas obtained thanks to macro **parab tg 3 lines** from the three points *focus 1*, *focus 2* and *focus 3* chosen on the circumcircle of the triangle defined by the three given lines.

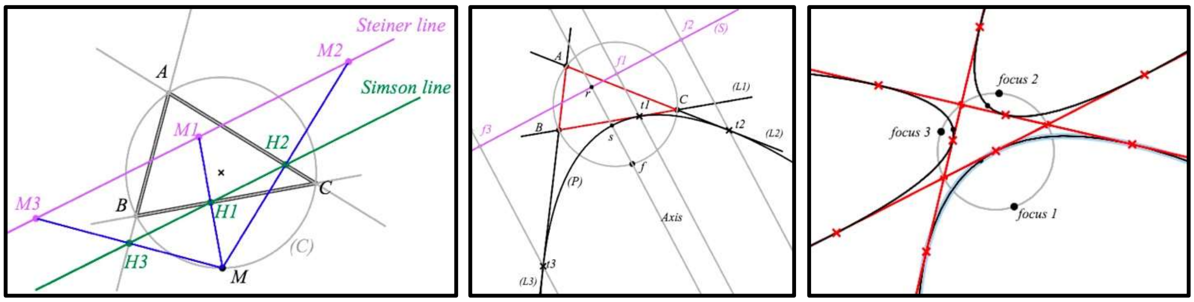

Figure 8: Parabolas tangent to three given lines

2.5. Isoptic curves of a parabola

The following algorithm will make it possible to construct the set of points from which a parabola can be seen under an angle given by a slider commanding a number that can vary from 0° to 180°. The parabola *(P)* is given with its focus *f* and its summit *s*.
This construction algorithm shows how to obtain two points *l1* and *l2* from where the parabola *(P)* is seen under a constant angle defined by the slider (Figure 9 left). Then we construct the loci of *l1* and *l2*, called *(H1)* and *(H2)*. By combining these two loci, we obtain the sought isoptic (Figure 9 center). A glimpse of the displayed result allows us to conjecture that this isoptic could be a branch of hyperbola. We will explain how to reach such conjecture and how to find the parameters of such a hyperbola (foci and directrix).

1. Ray *(R1)* = [*sf*)	13. Create a slider (from 0 to 180) and call the number displayed ***slider***
2. Evaluate *sf* and then *90* as the result of *sf-sf+90*	14. Evaluate *ang* = *sf* - *sf* +*slider* and then *-ang*
3. *(R2)* image of *(R1)* by the rotation centered at *s* and of angle *sf-sf+90*	15. *t2* image of *t1* by the rotation centered at *t* and of angle *-ang*
4. *m* point on *(R1)* and *(R3)* image of *(R2)* by the translation mapping *s* onto *m*	16. Ray *(S1)* = [*t t2*) intersecting *(P)* at *u*
5. *t* intersection point between *(P)* and *(R3)*	17. Ray *(S2)* image of *(S1)* by the translation mapping *t* onto *t1*
6. *h* symmetric point of *f* with respect to *s*	18. *v* and *w* intersection points between *(P)* and *(S2)*
7. Perpendicular line (*D*) to *(R1) through h*	19. *i* and *j* midpoints of [*t u*] and [*v w*]
8. *r* orthogonal projection of *t* on *(D)*	20. Line *(ij)* intersects *(P)* at *k*
9. *(L1)* perpendicular bisector of [*rf*], (tangent line to *(P)* at *t*	21. Line *(L2)* parallel to *(S1)* at *k*
10. Evaluate 4.*sf.* Create a point *p* on *(L1)*	22. *l1* intersection between *(L1)* and *(L2)*
11. *q* obtained by measurement transfer of 4.*sf* on *(L1)* from *p*	23. *(H1)* locus of *l1* (commanded by *m)*
12. *t1* image of *t* by translation mapping *p* onto *q*	*24. (H2)* locus of *l2* (commanded by *m*) where *l2* is the symmetric *of l1 with respect to (R1)*

In using the slider, we can check that for a value of 90° given by the slider, *l1* belongs to *(D)* which is a known result and the isoptic for 90° is not an hyperbola but a line, the directrix of the parabola.
In Figure 9 right, we have created point *e* intersection between *(H1)* and ray *(R1')* which is the symmetric ray of *(R1)* with respect to *s*. Then we create a point *f'* on the ray *(R1')*. At last we construct

the hyperbola *(H)* whose foci are *f* and *f'* and passing through e. We can state that, in changing the position of *f'*, there is a moment when *(H)* can be superimposed to *(H1)* and *(H2)* which means that the isoptic we have constructed is possibly a hyperbola. The idea we could have at this moment of our investigation is that this hyperbola has one of its foci which is the focus of the parabola and one of its directrix which is the directrix of the parabola. This conjecture will be corroborated below.

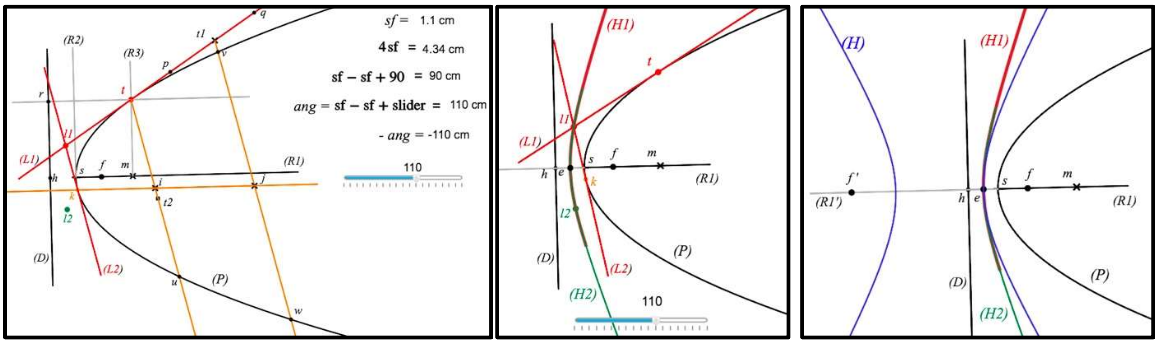

Figure 9: Isoptic curves of a parabola

Other investigations to grab the final property experimentally (Figure 10 left):

A useful formula: with the notations of Figure 10 left representing a hyperbola whose foci are *f* and *f'*, whose center is *O* and one summit *e*, we know that:

$oh = \frac{oe^2}{of}$ and as $oe = oh + he$ and $of = oh + hf$ we can deduce that $ho = \frac{he^2}{he-2h}$. This formula is available when e is on the right side of h. In the other case, the formula becomes $ho = \frac{he^2}{he+2hf}$. A formula encompassing the two previous cases could be $ho = \frac{he^2}{he+2.sl.hf}$ where $sl = sign(90 - \boldsymbol{slider})$. Note that **slider** is a number displayed by a slider of the software, more than 90 when e is on the right of h and less than 90 when e is on the left of h.

From that formula we can create a construction algorithm for the center O of an hyperbola knowing one focus *f*, one summit e and the foot of the directrix h.

The construction algorithm:

1. Create two points *h* and *f* and ray [*f h)* 2. Create *e* on this ray 3. Create expression *sign(90-x)* 4. Create a slider (bounds: 0 and 180). Number displayed: **slider** 5. Evaluate the previous expression for **slider** to get **sl** 6. Evaluate *he* and *hf* and then $ho = \frac{he^2}{he+2.\boldsymbol{sl}.hf}$	7. Construct ray *(R)* symmetric of ray [*hf)* with respect to *h* 8. Circle *(C)* centered at *h* and of radius *ho* 9. *O* intersection point between *(C)* and *(R)* 10. *f '* symmetric point of *f* with respect to *O* 11. *(H)* hyperbola defined by the two foci *f* and *f '* and passing through *e*

This construction is recorded as a macro construction called **hyper from h f e** with initial objects three points h, *f*, e, a number between 0 and 180 and the expression *sign(90-x)* and final objects the hyperbola whose foci are *f* and *f'* and passing through *e* and *f'*.

Final investigations: in Figure 10 right, we start from a simplified version of Figure 9 center. We kept parabola *(P)*, its focus *f*, its summit *s*, its directrix *(D)* and the point *h* of *(D)* colinear with *f* and *s*. We kept also *(H1)* and *(H2)* which combination represents the isoptic of *(P)* corresponding to the number displayed by the slider (here 110). Eventually we also kept point *e* which is the intersection point between *(H1)* and line *(fs)*. Now we apply macro **hyper from h f e** to the points *h*, *e* and *f* to get the hyperbola *(H)* with foci *f* and *f'*, centered at *O* and passing through *e*. We can state immediately that *(H)* seems to be superimposed to *(H1)* and *(H2)*. This observation persists when we change the value of the slider. To increase the level of the corroboration we create a point q on *(H)* to which we apply macro **2 lines tg parab** to get the two tangent lines to *(P)* passing through q. We measure and display the angle between these two lines and we obtain the same number as the one displayed by the slider; this observation persists when we move point q along the right branch of hyperbola *(H)*. Same observations can be made for other values returned by the slider.

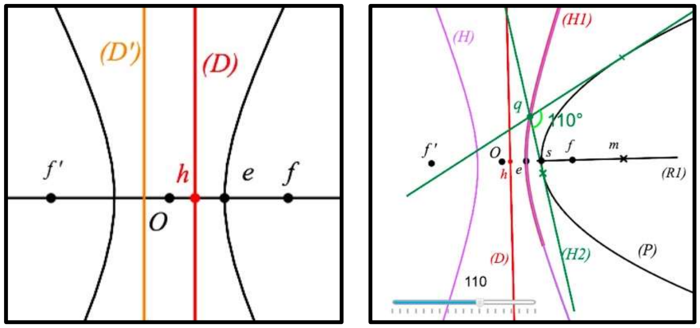

Figure 10: Corroboration of the conjecture about isoptic curves of a parabola

3. Third work around hyperbolas

3.1. Construction algorithm of the axes of a hyperbola

(H) is a given hyperbola, the following construction algorithm allows us to obtain its axes and its center (Figure 11 left).

1. Apply macro **center ellyp** to get the center O of *(H)*
2. Circle *(C)* centered at O passing through a point l chosen on the left branch of the hyperbola
3. n symmetric point of l with respect to O
4. m third point of intersection between *(C)* and *(H)*
5. i midpoint of $[lm]$ and j midpoint of $[mn]$
6. $Axis1$ is line *(Oi)* and $Axis2$ is line *(Oj)*

This construction is recorded as a macro construction called **axes hyper** with initial object a hyperbola and whose final objects are the two axes and the center of the given hyperbola.

3.2. Construction algorithm of the foci (and the center) of a hyperbola

For *(H)* is a given hyperbola, the following construction algorithm allows us to obtain its foci (Figure 11 center):

1. Apply macro **axes hyper** to get the two axes of *(H)* and its center O. Construct summits $A1$, $A2$
2. *(C)* centered at O passing through a point q chosen on the left branch of *(H)*
3. Xq orthogonal projection of q on $Axis1$
4. Measure $OXq = x$, $OA1 = a$ and $Oq = r$
5. Evaluate $a^2.\frac{r^2-a^2}{x^2-a^2}$ *which is* $\boldsymbol{b^2}$
6. Evaluate $\sqrt{b^2}$ which $is\ b$
7. B intersection point between $Axis2$ and circle centered at O and of radius b
8. C image of $A2$ by translation mapping O onto B
9. $f1$ and $f2$ intersection points between $Axis1$ and circle centered at O and passing through B

This construction is recorded as a macro construction called **foci hyper** whose initial objects are a hyperbola and a point on its left branch and whose final objects are the two foci of the given hyperbola. We can check on Figure 11 right that line *(OC)* is an asymptotic line of the hyperbola. The second one is its symmetric with respect to *Axis1*.

So, we have recorded the macro construction called **asympt hyper** whose initial objects are a hyperbola and a point on its left branch and final objects the asymptotic lines of the hyperbola.

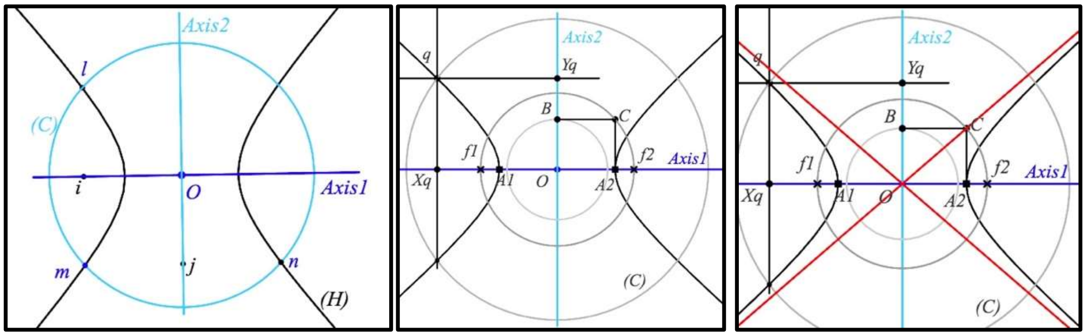

Figure 11: Construction of center, axes, foci and asymptotic lines of a hyperbola

3.3. Construction algorithms of the tangent line at a point of a hyperbola (two cases)

A hyperbola *(H)* is given by its two foci *f1* and *f2* and a point *q* defining the left branch of *(H)*.
Here are the algorithms for construction of the tangent line *(T2)* at a point *t2* of the right branch of the hyperbola (algorithm 1, Figure left) and the one of the tangent line *(T1)* at a point *t1* of its left branch (algorithm 2, Figure right).

Algorithm 1:	**Algorithm 2**
1. Line *(f1f2)* and its intersection points *a1* and *a2* with *(H)*	1. Line *(f1f2)* and its intersection points *a1* and *a2* with *(H)*
2. *g2* symmetric point of *f2* with respect to *a2*	2. *g1* symmetric point of *f1* with respect to *a1*
3. Circle *(C1)* centered at *f1* passing through *g2*	*3.* Circle *(C2)* centered at *f2* passing through *g1*
4. A point *t2* on the right branch of *(H)*	*4.* A point *t1* on the left branch of *(H)*
5. Ray *[f1 t2*) intersecting *(C1)* at *m1*	*5.* Ray *[f2 t1*) intersecting *(C2)* at *m2*
6. *(T2)* perpendicular bisector of [*f2 m1*]	6. *(T1)* perpendicular bisector of [*f1 m2*]

These constructions are recorded as two macro constructions:
The first one is called **tgt hyper right** with initial objects the two foci of a hyperbola, the point defining it (same side of the first foci: left side) and a contact point on the right branch of the hyperbola and final object the tangent line to the hyperbola at this last point.
The second one is called **tgt hyper left** with initial objects the two foci of a hyperbola, the point defining it (same side of the first foci: left side) and a contact point on the left branch of the hyperbola and final object the tangent line to the hyperbola at this last point.

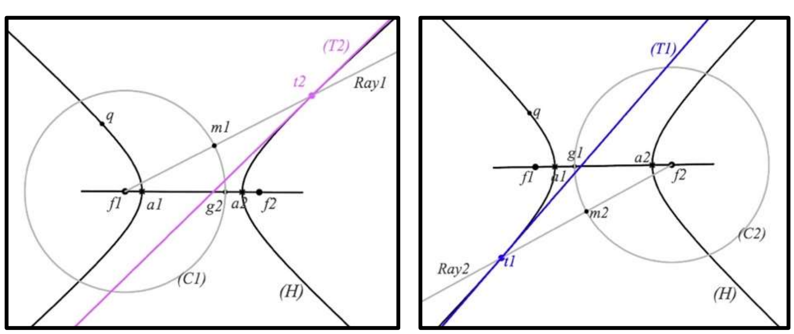

Figure 12: Tangent line at a point of a hyperbola

3.4. Construction algorithms of the tangent lines to a hyperbola from a given point (two cases)

Here are the two algorithms (Figure 13):

Case 1 (from *U1*)	Case 2 (from *U2*)
1. Hyperbola *(H)* from *f1, f2* and *q*	1. Hyperbola *(H)* from *f1, f2* and *q*
2. Apply macro **asympt hyper** to get the asymptotic lines *Asympt1* and *Asympt2*	2. Apply macro **asympt hyper** to get the asymptotic lines *Asympt1* and *Asympt2*
3. Circle *(C)* centered at *f1* passing through *g2,* symmetric point of *f2* with respect to *a2*	*3.* Circle *(C)* centered at *f1* passing through *g2,* symmetric point of *f2* with respect to *a2*
4. Circle *(C1)* centered at a point *U1 passing through f2* intersecting *(C)* at *r* and *s*	*4.* Circle *(C2)* centered at a point *U2* passing through *f2* intersecting *(C)* at *r* and *s*
6. *(T1)* perpendicular bisector of [*f2 r*]	6. *(T2)* perpendicular bisector of [*f2 s*]
7. *t1* intersection of *(T1)* and *(f1 r)*	7. *t2* intersection of *(T2)* and *(f1 s)*

These algorithms are recorded as two macro constructions:

Macro **tgt hyp 1 from pt** whose initial objects are the two foci of a hyperbola *(H)*, one of its point *q* on the left branch and a point *U1* and final objects a tangent line *(T1)* to *(H)* at *t1*, tangent to the right branch if *U1* is located under the right asymptotic line (*Asympt1*), to the left branch of *(H)* if *U1* is located above the right asymptotic line (*Asympt1*).

Macro **tgt hyp 2 from pt** whose initial objects are the two foci of a hyperbola *(H)*, one of its point *q* on the left branch and a point *U2* and final objects a tangent line *(T2)* to *(H)* at *t2*, tangent to the left branch if *U2* is located under the left asymptotic line (*Asympt2*), to the right branch of *(H)* if *U2* is located above the left asymptotic line (*Asympt2*).

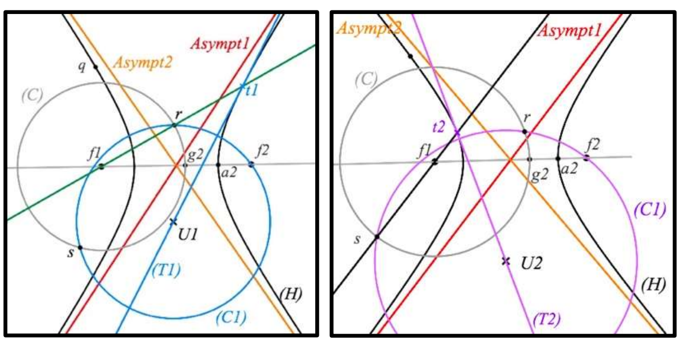

Figure 13: Tangent lines to a hyperbola from a given point

Below in figure 14, are represented the four different positions of the tangent lines of *(T1)* and *(T2)* regarding the position of point *U* relativelely to the asymptotic lines of the hyperbola

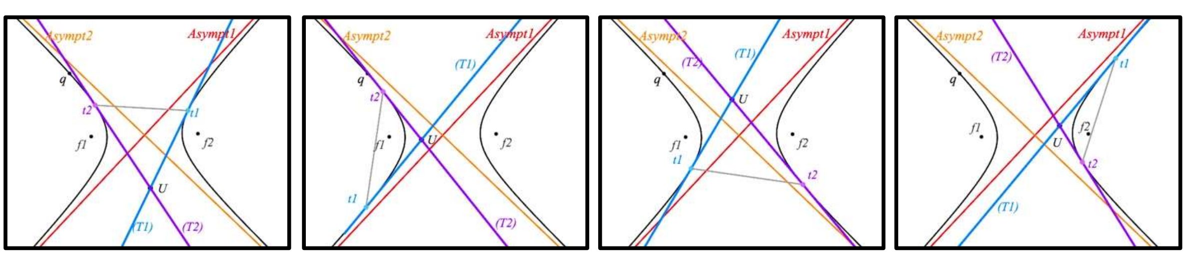

Figure 14: Tangent lines to a hyperbola according to the position of their origin

3.5. Hyperbolas tangent to the three sides of a given triangle (Figure 2 center and right):

3.5.1. Existence of such hyperbolas: using the previous construction algorithm (macro **tgt hyp right**), it is possible to construct three tangent lines to the same branch of a given hyperbola, these three lines defining a triangle *ABC*. It is easy to check that one of the foci of this hyperbola is always inside the circumcircle of triangle *ABC*. (See Figure 15 left).

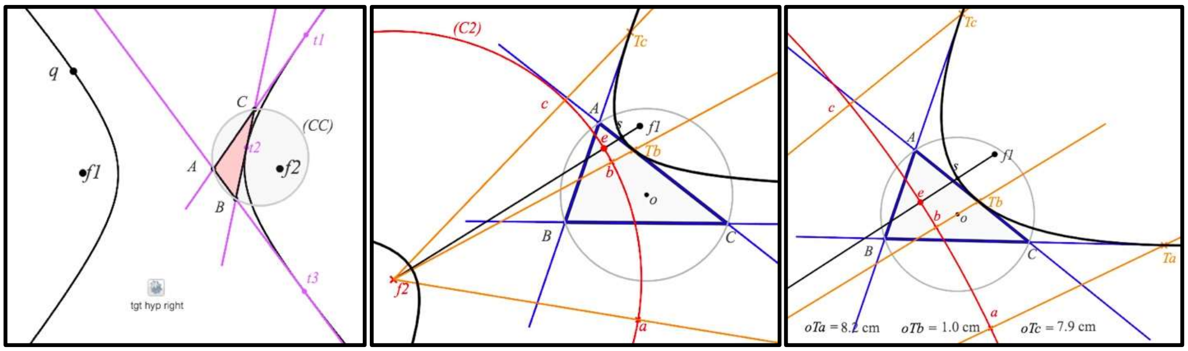

Figure 15: Hyperbolas tangent to the three sides of a triangle

3.5.2. Construction algorithm of a branch of hyperbola tangent to the three sides of a given triangle *ABC* is a given triangle and *f1* a given point outside the triangle (Figure 15 center). If a branch of hyperbola having *f1* as one of its foci is tangent to the three lines supporting the sides of ABC, necessarily the symmetric points of *f1* with respect to these lines, *a*, *b* and *c* belongs to the director circle *(C2)* ssociated with the second focus *f2* which is the center of this circle. Necessarily, *f1* must be inside the circumcircle of *ABC* because if not, *f*2 would be located inside (*C2*) which is impossible. If this branch of hyperbola exists the three contact points would be respectively the intersection points between ray [*f2 a*) and line (*BC*) for *Ta*, between ray [*f2 b*) and line (*AC*) for *Tb* and between ray [*f2 c*) and line (*AB*) for *Tc*.

Figure 15 right shows a case where the position of *f2* allows the construction of an expected hyperbola.

The hyperbola solution of our problem is defined by its two foci *f1* (given point) and *f2* (center of director circle) and one summit s (midpoint of [*e f2*] where *e* is the intersecion between [*f1 f2*] and *(C2)*.

To check positions of *f1* allowing the existence of the three points *Ta*, *Tb* and *Tc* and by the way the existence of a branch of hyperbola tangent to the thee lines *(AB)*, *(BC)* and *(CA)*, we measure and display the distances *oTa*, *oTb* nd *oTc* and move *f2* until a position where these three distances exist.

3.5.3. Construction algorithm (Figure 15 center)

1. Triangle *(ABC)* and point *f1* 2. Lines *(AB)*, *(BC)* and *(CA)* *3. Points c,* a *and b* symmetric of *f1* with respect to these lines *4.* Circle *(C2)* centered at a point *f2* passing through *a, b and c* 5. *e* intersection of *(C2)* and *[f1 f2]*	6. *s* midpoint of *[e f1]* 7. Hyperbola with foci *f1* and *f2* passing through *s* 8. *Ta* intersection of *[f2 a)* and *(BC)* 9. *Tb* intersection of *[f2 b)* and *(AC)* 10. *Tc* intersection of *[f2 c)* and *(AB)*

This construction is recorded as a macro construction called **tri tgt hyp** with initial objects a triangle and a point inside its circumcircle (but outside the triangle) and whose final objects are the hyperbola which first focus is the given point and the the triangle linking the contct points

3.5.4. Possible locations of the first focus

If we move point *f1* where it is allowed by the previous macro we can state quickly that the hyperbola does not always exist: in fact, this result can be reached by observing when the triangle linking the contact points appears or disappears. **I had the idea to move point *f1* along segments parallel to the sides of the given triangle.** That was an amazing idea because I could quickly conjecture that the hyperbola exists when *f1* is located inside the circumcircle but outside a special triangle which seemed to be similar to the given triangle. The measurements taken during my experiments led to obtain a ratio close to 1.60 (I suspected the golden ratio) and the center of the dilation transforming the given triangle onto this one being the orthocenter of the given triangle. To corroborate this conjecture, starting from a triangle *ABC*, I constructed its orthocenter *h* and transformed it by the dilation centered at *h* and with ratio equal to $\frac{1+\sqrt{5}}{2}$ (close to 1.618). Then, I applied the previous macro to ABC and a point *f1* in the previous suspected part of the plane where I expected the hyperbola to exist. And it works!

3.5.5. Final conjectures (Figure 16 right)

About hyperbolas tangent to the three sides of a triangle

ABC is a triangle, *(C)* its circumcircle, *h* its orthocenter, *A'B'C'* the image of triangle *ABC* by the dilation centered at *h* and of ratio, the golden ratio $\frac{1+\sqrt{5}}{2}$. Each point belonging to *(C)* but outside *A'B'C'* is one of the foci *f1* of a hyperbola tangent to the three lines supported by the sides of *ABC*.

About conics tangent to the three sides of a triangle

ABC is a triangle, *(C)* its circumcircle, each point *f1* of the plane except the points of the sides of the triangle and the points of the three portions of planes opposite to angles $\angle A$, $\angle B$ and $\angle C$ are the first focus of a conic tangent to the three lines supported by the sides of *ABC*. More precisely:

→This conic is an ellipse when *f1* is inside *ABC* or outside its circumcircle

→This conic is a parabola when *f1* is on its circumcircle

→This conic is a hyperbola when *f1* is inside its circumcircle but outside the image of triangle ABC by the dilation centered at the orthocenter of *ABC* and of ratio, the golden ratio $\frac{1+\sqrt{5}}{2}$.

The plane portions corresponding to these three cases are visible in figure 16 right.

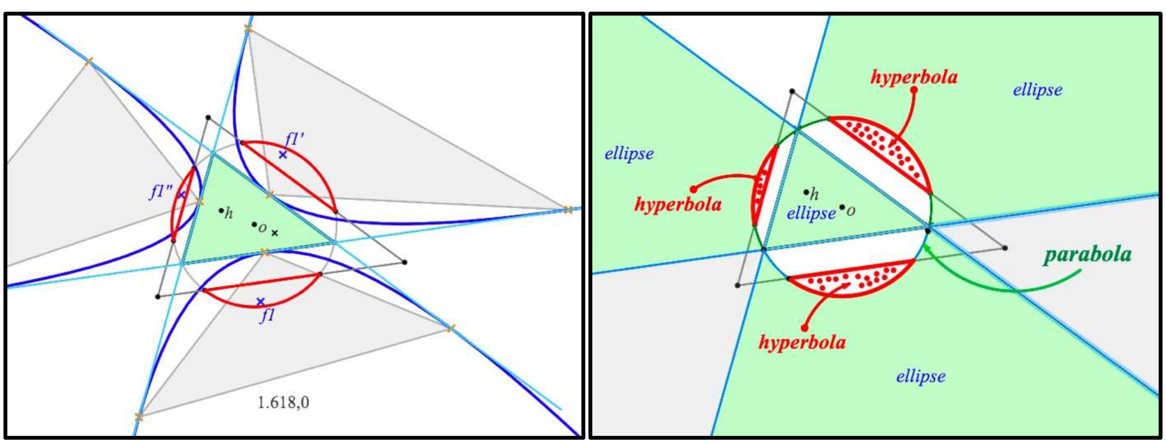

Figure 16: Locations of hyperbolas tangent to a "triangle" and final conjectures

3.6. Isoptic curves of a branch of hyperbola

The following algorithm will make it possible to construct the set of points from which a branch of hyperbola is seen under a given angle.

1. Points *A1* and *A2* intersection of *(H)* with *[f1 f2]*	10. Create a point *s2* on (*T2*) by Measurement transfer of *h* (on the right of *t2)*
2. Asymptotic lines in applying macro **asymp hyper**	11. Create a slider between 0 and 180
3. Edit a number *d* and evaluate 2*d*	12. Evaluate the opposite of the number displayed by the slider
4. Create points *e* and *g* by measurement transfer of *d* and *2d* on vector *f1O*	13. Use this last number to rotate *s2* around *t2* to get *w2*
5. Ray [*e g*) and a point *m* on it	14. Ray [*t2 w2*) and its intersection *t2′* with (*H*)
6. *B2* intersection between *Asympt1* and the perpendicular to *(f1O)* at *A2*	15. Midpoint *i* of [*t2 t2′*]
7. Ray *[m b2*) where *b2* is the image of *B2* by translation mapping *A2* onto *m*	16. Intersection *j* between *(H)* and ray [*O i*)
8. Apply macro **tgt hyp right** to get the tangent line (*T2*) to (*H*) at *t2* (on *(H)* and [*m b2*))	17. *(T2)* parallel to (*t2 t2′*) through *j*
9. Evaluate and display distance between *f2* and (*T2*) called *h*	18. Intersection ***l*** between *(T2)* and *(T2′)*

Figure 17 left illustrates the stages from 1 to 11 and Figure 17 center the stages from 12 to 18.
The locus of point ***l*** is the part of the isoptic generated by the tangent lines to *(H)* associated to the points *m* of ray [*e g*). Points ***l*** exist until ray [*t2 w2*) becomes parallel to *Asympt 2*. The position of *e* (commanded by *d*) allows to avoid positions of *m* where (*T2'*) does not exist.
To be sure to obtain the complete isoptic related to the right branch of the hyperbola, we complete the locus of *l* by the locus of *l'* its symmetric point with respect to *(f1f2)*: see Figure 17 right where different isoptic curves are visible, obtained by changing the values of the slider without choosing a value superior to the angle between the two asyptotic lines and letting their trace be active.

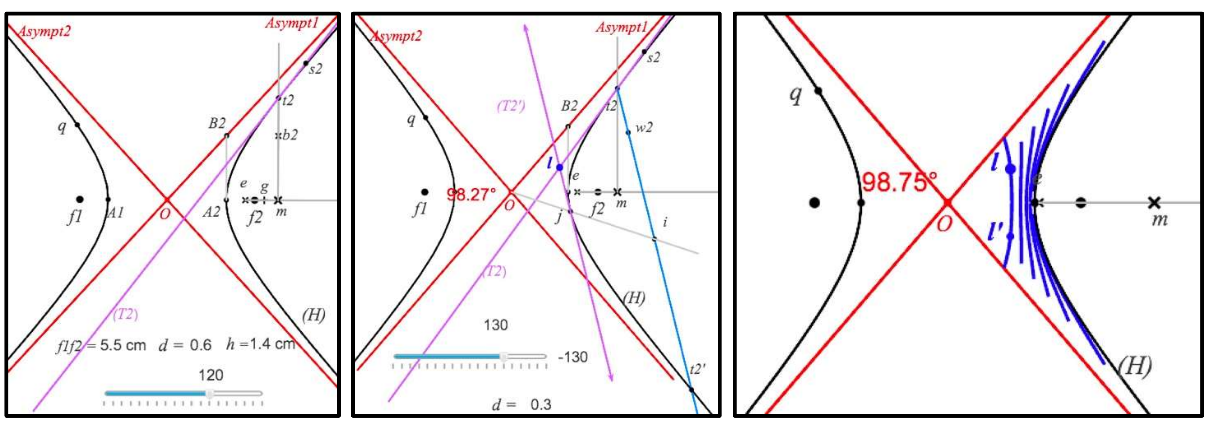

Figure 17: Isoptics of hyperbolas

About the limit position of point *e*:

The equation of the hyperbola *(H)* is $\frac{x^2}{a^2}-\frac{y^2}{b^2}-1=0$ in a system of axes centered at O. This equation is equivalent to $y=\frac{b}{a}.\sqrt{x^2-a^2}$ and we know that $\frac{dy}{dx}=\frac{b}{a}.\frac{x}{\sqrt{x^2-a^2}}$. Therefore, the slope of the

tangent line at M_0 (x_0, y_0) is $\frac{b}{a}.\frac{x_0}{\sqrt{{x_0}^2-a^2}}$ = *tan(u)***.** Let us evaluate the slope of [*t2 w2)* (*v* is the angle displayed by the slider) which is:
tan(u-v)=$\frac{\tan(u)-\tan(v)}{1+\tan(u).\tan(v)}$. If we call *m* the value -$\frac{b}{a}$ which is the slope of *Asympt2,* we want to find the position of M_0, when [*t2 w2)* is parallel *to Asympt2*, that is to say:
$\frac{\tan(u)-\tan(v)}{1+\tan(u).\tan(v)} = m$ or $tan(u) = \frac{m+\tan(v)}{1-m.\tan(v)}$ or $\frac{b}{a}.\frac{x_0}{\sqrt{{x_0}^2-a^2}} = \frac{-\frac{b}{a}+\tan(v)}{1+\frac{b}{a}.\tan(v)}$, from which we obtain
$\frac{x_0}{\sqrt{{x_0}^2-a^2}} = \frac{-ab+a^2.\tan(v)}{ab+b^2.\tan(v)} = M$ equivalent to ${x_0}^2 = \frac{M^2a^2}{M^2-1}$ and $x_0 = \sqrt{\frac{M^2a^2}{M^2-1}}$.

Eventually the limit value of x_0 to construct a point viewing the right branch under an angle of *v* is the previous value. x_0 is the distance *Oe.*

4. Conclusion

This article was the occasion of an original visit of the conics centered on the problem of their tritangency. Almost every construction gives the opportunity to create a detailed macro construction which will be used for the following investigations. The purely geometric construction of tritangent conics led to the discovery of two original conjectures on the classification of such conics, one including the golden ratio. Once again, the experimental approach mediated by dynamic geometry has shown its power for the illustration of known results with some of their lesser known consequences and the discovery of original and highly plausible conjectures. The reader will find there detailed all the algorithms of the purely geometric constructions used in order to realize that the initiation to programming can and must go through the stage of geometric macro constructions before approaching more complex formalizations.

References

[1] POLYA G., 1945, *How to solve it*, Princeton, University Press
[2] LAKATOS I., 1984, *Preuves et réfutations Essai sur la logique de la découverte,* Hermann,
[3] DAHAN J.J., 2002, *How to teach Mathematics in showing all the hidden stages of a true research. Examples with Cabri, in Proceedings of ATCM 2002,* Melaca, Malaysia
[4] DAHAN J.J., 2005, *La démarche de découverte expérimentalement médiée par Cabri-géomètre en mathématiques,* PhD thesis, Université Joseph Fourier, Grenoble, France http://tel.archives-ouvertes.fr/tel-00356107/fr/
[5] DAHAN J.J., 2019, *Steiner Ellipse and Marden's Theorem, in Proceeding 24th Asian Technology Conference in Mathematics, pp 1-13, Leshan, China*
[6] DAHAN J.J., 2020*, The Role of Technology to Build a Simple Proof: The Case of the Ellipses of Maximum Area Inscribed in a Triangle, in Electronic Proceedings 25th Asian Technology Conference in Mathematics, Virtual Format*

Software

Cabri Author (Version 4.10) by Cabrilog at http://www.cabri.com