The richness of a multi software approach: solving trigonometric equations with Cabri, TI NSpire and Autograph and therefore modelling refraction

Jean-Jacques Dahan
jjdahan@wanadoo.fr
IREM of Toulouse
Paul Sabatier University Toulouse
France

Abstract: *Usually, teachers use software chosen by their school or the department of their college. Sometimes they have been trained on the use of this software, and are convinced that they can enrich their teaching in using it. I will show that even an expert can miss a lot of opportunities to change the points of view for a given problem. We will start with the equation sinx=siny and conjecture its solutions with the technique of soft loci in Cabri. We will also look at the possibility of locking variables using TI Nspire. Another point of view will be presented by using Autograph to display the intersection between the surface z = sin(x)-sin(y) and the plane z = 0. We will extend this study to the family of equations "sinx = k.siny" in order to present an existing geometric model of refraction together with an analytic model I have created by using the power of the CAS embedded in Autograph and TI N'Spire.*

1. Introduction

1.1. Equation *sin(x) = sin(y)* in textbooks

This equation is introduced as a method to research angles x in radians by verifying the equation $sin(x) = a$ where a is a given number between -1 and 1. Geometrically, it is shown that this problem is the same as looking at the angle x such as $sin(x)=sin(x_0)$ where angle x_0 is a particular solution found on the unit circle. The solutions are presented very often like this:

$$sin(x)=sin(x_0) \Leftrightarrow x = x_0 + 2k\pi \text{ or } x = \pi - x_0 + 2k\pi,\ k \in Z$$

and illustrated by a figure like figure 1 on the left.

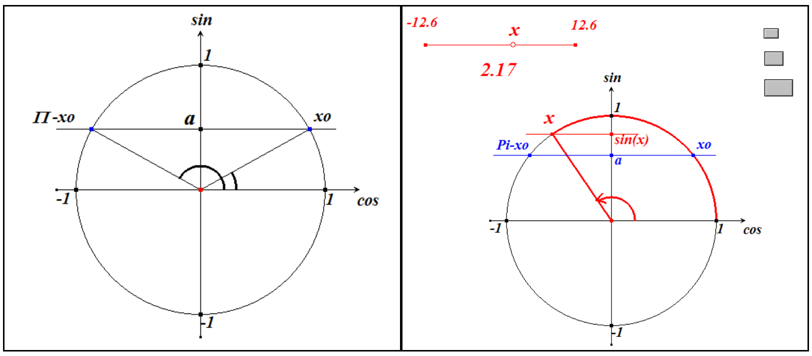

Figure 1

1.2. Improvement with a dynamic visualisation

We can use a dynamic geometry software like Cabri to create a figure where a slider can command the value of the angle x in radians (between a negative bound and a positive bound). We can display the corresponding arc-angle (bold and red in figure 1 on the right) in order to find experimentally the values of x that have a given number a as a sine. The first experimental value is called x_0. The

investigation leads us to verify the result given by textbooks. Dragging point *x* along the slider (figure 1 on the right) moves the point *x* along the unit circle $sin(x)= a$.
This dynamic visualisation is called "monstration" ([1]). The aim of this process is to give to the observer data he will believe in immediately.

2. Experimental solving with soft loci

2.1. Visualising the set of solutions by "tracing" them with conditional constructions (Cabri 2 Plus)

Nearly twenty years of work with the Cabri environment has led to the well-known results concerning conditional constructions (J. M. Laborde and Y. Martin) and the very powerful and dynamic visualisations of polyhedra in parallel perspectives (G. Tulloue). I have already presented about one of the applications with soft loci during ATCM 2002 ([3]). I will use it again to investigate the previous problem and obtain the solutions geometrically. Here below are the experiments we did:

2.1.1 First experiment

Preparation of the figure: in a Cabri page, we display the axes, the expression "*sin(x)-sin(y)*", a free point *M* and its coordinates *x* and *y*. We evaluate the previous expression for the coordinates of *M*. The number we obtain can be positive or negative, depending on its position in the plane. Using the technique of conditional constructions, we create two new points, superimposed to *M* : the first one, *p*, appears only when this number is positive and the second, *n*, only when this number is negative. These points are coloured with two different colours (*p* in blue and *n* in red).
Protocol of the experiment: as we want to find the points of the plane verifying $sin(x)\text{-}sin(y)=0$, the experiment will consist of activating the traces of these two points and dragging them everywhere. The plane will be coloured with these two chosen colours and the border of the two visualised sets will be the set of solutions.
Generative experiment: as planned in the protocol we drag point *M* everywhere and we obtain figure 2 on the left which seems to be a tiling of the plane with squares having the same sides. Therefore, this observation generate a conjecture about the solutions to our problem (that is the role of a generative experiment: see in [5]).

Conjecture: as the border seems to be a double set of perpendicular lines of the plane, parallel to lines $y=x$ and $y=-x$, we conjecture that the solutions could be given by; $y=x+2k.v$ or $y=-x+v+2k.v$ where *k* belongs to Z and *v* is a number.

Validative experiment: the aim of such an experiment is to confirm the conjecture, to increase its plausibility (could be also to reject it in observing properties which contradict the conjecture). We construct lines that seem to define the border that we want to discover, and we observe that, visually, with the accuracy of the constructed traces, these lines seems to be a very good approximation of our border.

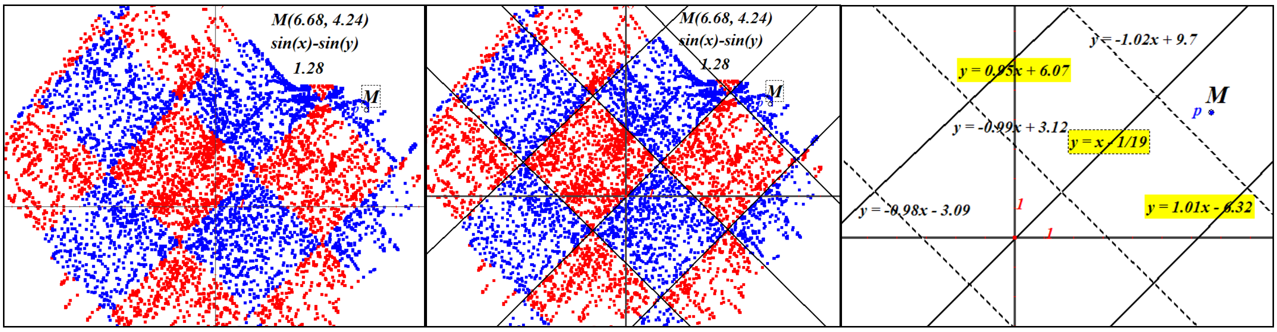

Figure 2

2.1.2. Second experiment

If we display the equations of these lines as shown in figure 2 on the right, we can guess that the result could be the previous one with $v=\pi$. The basis of this experiment is the same figure as the previous one. The protocol changes: we have only to display the equations. We improve our conjecture only by observing carefully the displayed equations, and hence, generate a new conjecture which gives the final good result. But at this stage of our research nothing can tells us that we have got the real result. Experimentally we have not reached **truth,** but only **plausibility** ([5]).

2.2. Visualising the set of solutions by “tracing” them and by locking a variable (TI N’Spire)

2.2.1 First experiment

The Cabri version embedded in the TI N’Spire software contains a new tool called “lock variable” which allows the locking of a value of the number obtained for *sin(x)-sin(y).* We drag point *M* only on positions where this value had been locked. So the experiment is simpler than in Cabri 2 Plus because this tool avoids to use conditional constructions.

The protocol now is this one: drag point M until a position where *sin(x)-sin(y)*is very close to zero. Lock this value. Put the trace of point P on. Try to drag point M everywhere.

The generative experiment gives the trace of points M and leads to the same conjecture but this allows us to visualise directly the set of points that verify approximately our equation (figure 3 on the left).

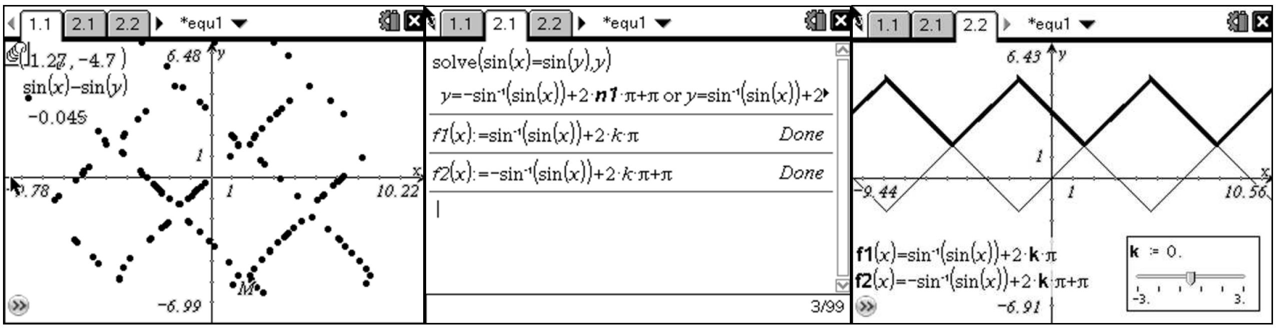

Figure 3

2.2.2 Solving the equation with the CAS embedded in TI N’Spire

In this software, which is also available in handhelds (the screenshots of figure 3 have been created in the handheld version of the software), we can create a multipage document where pages are connected. In figure 3 on the centre, we have used a different page to solve our equation and surprisingly, the CAS used does not give the expected solutions. In fact when we solve the equation *sin(x)=sin(y)* with respect to the unknown *y* we obtain:

$$y=-\sin^{-1}(\sin(x))+2.\boldsymbol{n1}.\pi+\pi \text{ or } y=\sin^{-1}(\sin(x))+2.\boldsymbol{n1}.\pi.$$

What we did in this page could have been considered as **a validative experiment**, because we wanted to confirm the conjecture done in the previous page. In fact, it is such an experiment, but the result of this experiment (the displayed solutions) does not seem to be a double set of linear functions. As a remark we can say that the user of such a CAS must know the interpretation of ***n1*** (which is a number in Z). Nevertheless this experiment leads to some doubt.

So, in order to understand the meaning of this strange formula, we store $\sin^{-1}(\sin(x))+2.k.\pi$ in f_1 and $-\sin^{-1}(\sin(x))+2.k.\pi+\pi$ in f_2 and display their curves for values of k belonging to Z.

2.2.3. A generative experiment in a Graphs & Geometry page

In such a page, we create a slider to control the values of k and start with $k=0$. We also take care to chose a step of 1 in the settings of the slider. The figure we obtain (figure 3 on the right) is the beginning of the validation of our conjecture. We can increase the plausibility of this conjecture by changing the values of k with the slider and observe dynamically that the set of our strange curves is exactly the same as the one got in the conjecture with linear functions. Let us remark that somebody with good knowledge of trigonometry would have guessed that the curve of f_1 is the one displayed by the software, because the expression $\sin^{-1}(\sin(x)$is a periodic one defined on R and equal to x only between $-\pi/2$ *and* $\pi/2$; such a person must know that the expression $\sin(\sin^{-1}(x)$is only defined between -1 and 1 and equal to x between these bounds.

3. A 3D approach of the problem

3.1. Direct solving with the technique of graph plotting in 2D (Autograph)

In Autograph, it is possible to obtain the graph of the curve defined implicitly by sin(x)-sin(y)=0 which seems to be the set we have discovered previously (see figure 4 on the left)

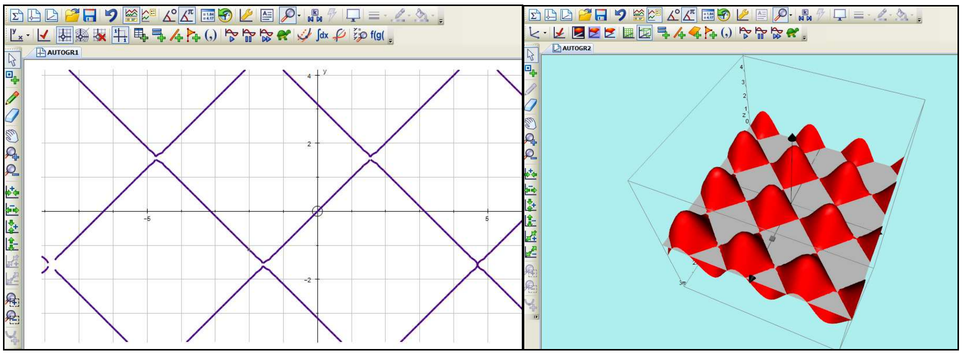

Figure 4

3.2. Solutions as the intersection points between a surface and the *xOy* plane

The same software contains a 3D graph plotter that we use in plotting the surface $z=\sin(x)-\sin(y)$ and the surface $z=0$. We can visualise the intersection points of these two surfaces: the double set of lines in the horizontal plane (see figure 4 on the right)

3.3. Technique of the bicolour visualisation of a surface in military perspective (Cabri)

We know how to represent in military perspective a surface given by an equation $z=f(x;y)$ in Cabri 2 Plus ([4]) by using the special tools "Locus" and "Locus of loci". We have represented (figure 5 on the left) the military perspective of the surface $z=\sin(x)-\sin(y)$. The technique is simple: we construct one point M (point having $z=\sin(x)-\sin(y)$ as the third coordinate) of the surface, the loci of *M* giving the sections of this surface by planes parallel to *xOz* and *yOz*, and finally the two loci of these sections.

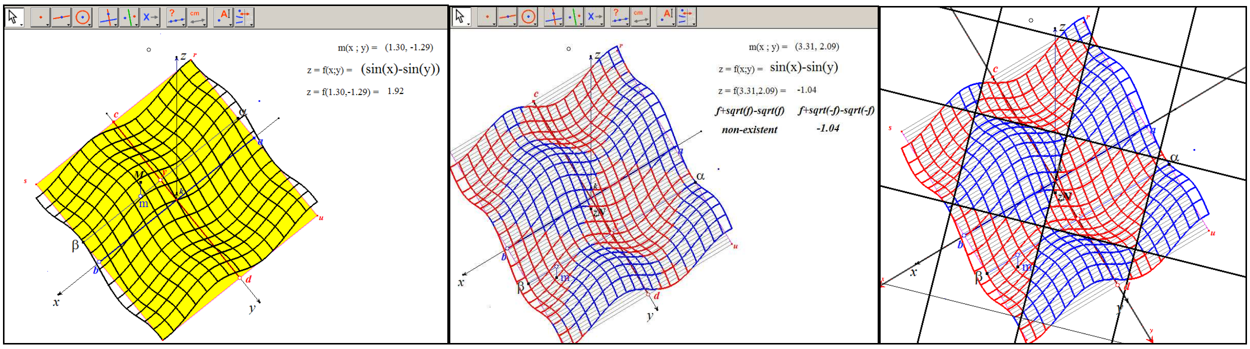

Figure 5

An algebraic trick achieved with Cabri 2 Plus, related to the tool “Expression”, allows us to improve the previous representation in colouring in blue, points verifying $z>0$, and in red the other ones. This is the trick: we create two new expressions, *f+sqrt(f)-sqrt(f)* and *f+sqrt(-f)-sqrt(-f)* and we evaluate these expressions for $f=z$; so, the first expression is equal to z when $z>0$ and does not exist when $z<0$; the second expression is equal to z when $z<0$ and does not exist when $z>0$; we construct the point *M* corresponding to the first expression and so the first surface in blue, the point M corresponding to the second expression and so the second surface in red (figure 5 on the centre).
Finally (figure 5 on the right) we have displayed the curves of functions $y=x$, $y=x+6{,}28$, $y=x-6{,}28$, $y=-x+3{,}14$, $y=-x-3{,}14$… and we check that these curves define the border of blue and red surfaces, so we have obtained experimentally the solutions of our problem. This validative experiment is called the experimental proof.

4. Investigations for a more general equation: *sin(x)= k.sin(y)*

4.1. Direct solving with the technique of dynamic graph plotting in 2D (Autograph)

We plot as we did in 3.1 the equation *sin(x)= k.sin(y)* and Autograph returns the screen we obtained for the equation *sin(x) = sin(y)* (figure 6 on the left) because as soon as a parameter appears in an equation, like *k*, the software recognizes it and with the “constant controller” we can change manually or automatically the values of this parameter. We can say that the software creates automatically a slider for each parameter appearing in any equation. So this “constant controller”, in the hand of a researcher quickly becomes a powerful tool of investigation.
We investigate by changing the values of *k* and our observations (like any generative experiment) lead to some conjectures about the shape of the solution curves when $0<k<1$ (figure 6 on the centre) and when $k>1$ (figure 6 on the right).
Other conjectures:
When $k=0$ we can observe a set of vertical lines having $x=k.\pi$ as equations which is the confirmation of elementary knowledge.
When *k* increases and goes to infinity we can conjecture that the solution curves approach horizontal lines having $y= k.\pi$ as equations, which can be proven easily if the equation *sin(x)=k.sin(y)*is written *sin(y)= (1/k).sin(x)*.So the problem to solve is the previous one because *1/k* approaches *0* when *k* goes to infinity.

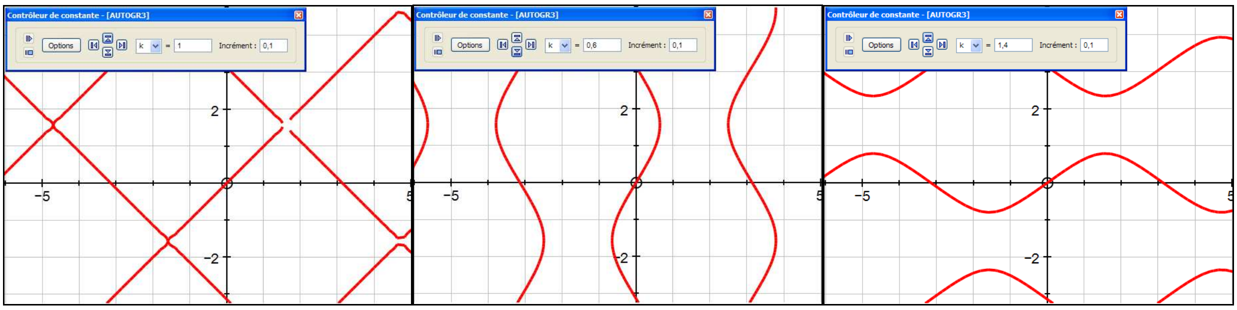

Figure 6

4.2. Dynamic validations with three softwares (TI N'Spire, Autograph and Cabri)

4.2.1. Investigations and validations with TI N'Spire

In a calculator page, we solve the equation $sin(x)=k.sin(y)$ to obtain:
$y=-sin^{-1}(sin(x)/k)+2.\mathbf{n1}.\pi+\pi$ and $-1\leq sin(x)/k\leq 1$ or $y=sin^{-1}(sin(x)/k)+2.\mathbf{n1}.\pi$ and $-1\leq sin(x)/k\leq 1$.
We store the expression $-sin^{-1}(sin(x)/k)+2.n.\pi+\pi$ in $f_1(x)$ and the expression $sin^{-1}(sin(x)/k)+2.n.\pi$ in $f_2(x)$ (as we did in 2.2.2.).
We display the curves of theses two functions in a Graphs & Geometry page after creating two sliders, one for k and another one for n (the step chosen for n in the settings of the slider is 1). We retrieve two sets of curves for a given value of k which are only a part of our solution because we have to change n to get the other ones (figure 7 on the left for $n=-1$, figure 7 on the right for $n=0$)

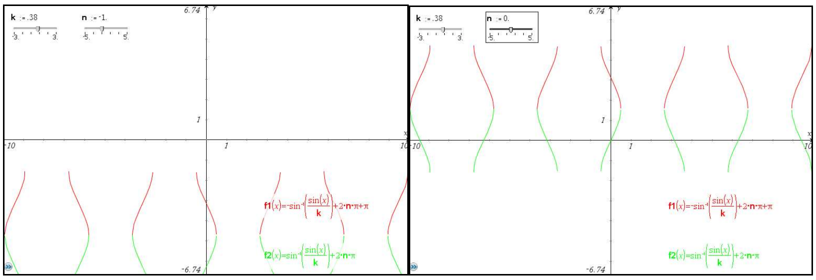

Figure 7

4.2.2. Validations with Autograph

In this case this software is very powerful because we have only to edit the equation $sin(x)=k.sin(y)$ as we did in 4.1 and $y=arcsin((sin(x))/k)+r$ and $y=-arcsin((sin(x))/k)+s$, to obtain figure 8 on the left. With the constant controller, we can explore the values of r and s to superimpose the two last curves to the set of solutions given by the solutions of $sin(x)=k.sin(y)$(see figure 8 on the right).

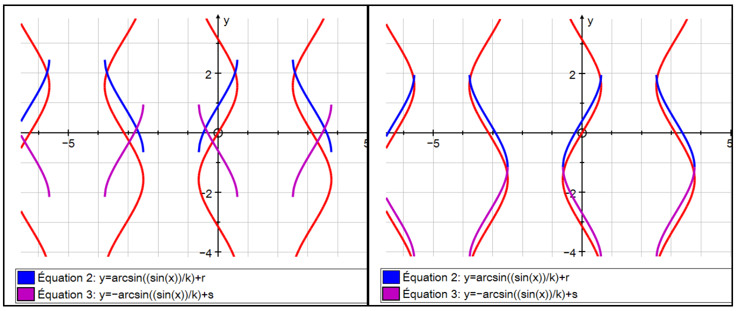

Figure 8

4.2.3. Investigations with Cabri

Here we display the expressions *arcsin((1/k)*sin(x))+2*n*pi* and *-arcsin((1/k)*sin(x))+2*n*pi+pi*. We create two sliders like the ones we used in TI N'Spire, one for *k* and another one for *n*. We obtain the same type of validation we had in 4.2.1. (figure 9)

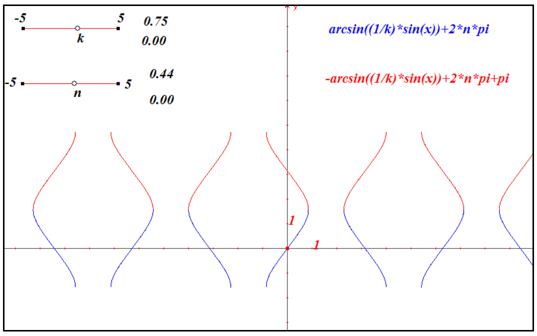

Figure 9

5. The refraction phenomenon

5.1. "*sin(x)= k.sin(y)*" as the equation of refraction

Given n_1 and n_2 the coefficients of refraction of air (or something else) and water (or something else), we know that we can determine the position of a refracted ray from the position of the incident ray with the Fermat rule, $n_1.sin(x)= n_2.sin(y)$, where x is the angle between the incident ray and the normal and y the angle between the refracted ray and the normal (figure 10). This equation can be written $sin(x)= k.sin(y)$ if $k=n_2/n_1$. We know also the Fresnel construction allowing us to construct easily the refracted ray from the incident one (figure 10), or to find y knowing x.

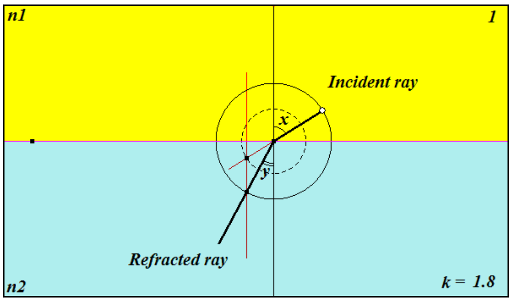

Figure 10

5.2. The geometrical Laugier-Martin model and its applications

Jean-Marie Laugier is a physics specialist, and Yves Martin a geometry specialist. They have worked together on a difficult problem which has arisen from their interest in modelling refraction with Cabri. The problem is this one: if two points are given (point "Eye" and point "Object"), one in the air and a second in the water, where is the "Contact" point on the plane between air and water where the incident ray coming from the point in the air crosses the refracted ray reaching the point in the water? (figure 11 on the left). They solved this problem ([2]) by using the Apollonius hyperbola and created a wonderful macro construction allowing the construction of this special point between air and water with initial objects the two given points, the line between air and water and the coefficient *k*. The purpose of this construction is to model where the "Eye" will see the "Object": not where it lies really but somewhere on the dotted line (figure 11 on the left). To construct the exact position where the "Eye" sees the "Object", we repeat the previous construction: the second time is for a second "Eye" close to the first one, as modelled in figure 11 on the right.

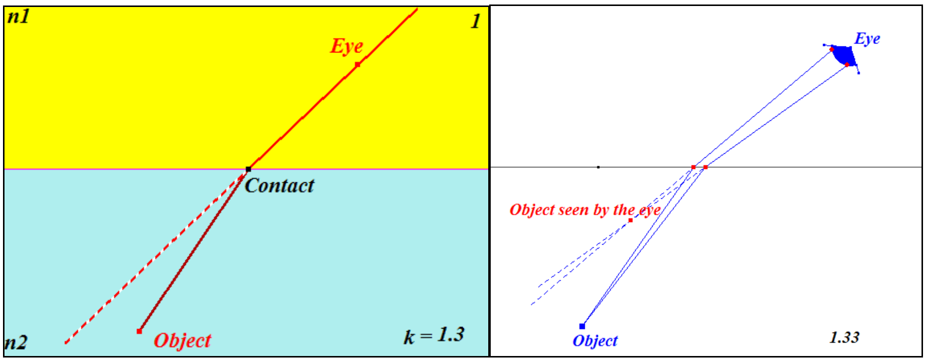

Figure 11

Application: when a fisherman on a boat looks at a fish in the water, he does not see the fish at its real position. I have used the Laugier-Martin macro and the tool Locus in Cabri to model this special problem as they did it when they solved their problem (figure 12). We can see the position of the real fish in black and the position of the one seen by the fisherman in red above the real one.

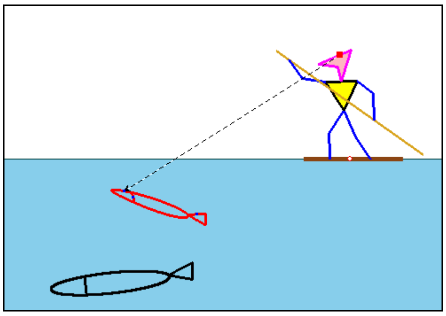

Figure 12

It is possible to explore and to get unexpected observations in dragging the real fish everywhere especially when it dives deeper and deeper.

5.3. My analytic model and its defaults

I wanted to create a macro needing less knowledge than the Laugier-Martin one. As Cabri makes it possible to construct curves of functions even with a lot of parameters, and as TI N'Spire makes it possible to solve rather complicated equations, I have explored this problem analytically.

5.3.1. The new problem to solve

In relation with figure 13 on the left, b, c and l are given, and we have to find the position of I in finding a such as $sin(x)=k.sin(y)$(constraint). In this figure the "Eye" is point o, the "Object" is point O and the contact point is I. The constraint can be written: $\frac{(l-a)}{\sqrt{(l-a)^2+c^2}}=\frac{k.a}{\sqrt{a^2+b^2}}$.

My problem is: solve this equation and use its solution to create a macro allowing the construction of I and therefore the construction of the virtual point seen by o instead of the real one O.

5.3.2. Starting with TI N'Spire

We solve the previous equation with respect to *a*, store *x* in *a* and store the expression we get in $f_1(x)$ which is given by the formula: $(k^2\text{-}1).x^4\text{-}2.(k^2\text{-}1).l.x^3\text{-}((b^2\text{-}c^2.k^2\text{-}(k^2\text{-}1).l^2).x^2+2.b^2.l.x\text{-}b^2.l^2$

In another page (figure 13 on the right), we construct *O* on the y axis (below), *o* as a free point (with both coordinates positive); the abscissa of *o* is commanded by the slider "*l*" . The curve of function f_1 is displayed in bold red and point *I* is the intersection point between this curve and the *x* axis (abscissa between *0* and *l*)

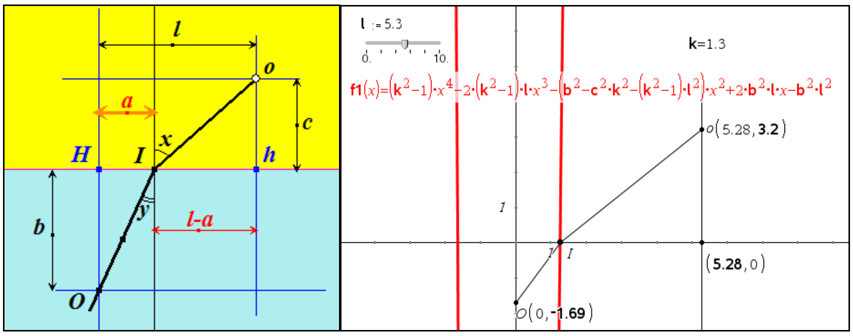

Figure 13

5.3.3. My solution with Cabri 2 Plus

In a Cabri page (figure 14 on the left) we show the curve corresponding to the previous expression in the system of axis. We create point *J* and its coordinates, especially its abscissa, which is the value of *a* from which we need to construct *I* on the line between air and water. This construction needs a Cabri geometrical trick in order to be correct even when *O* is located on the right side of *o*. We construct the vector from *H* to *h* and, with the compass tool, the circle centred on *H* and having the abscissa of *J* as a radius. I is the intersection point between this circle and this vector. With this figure I can create a macro having as initial objects, *o*, *O*, the line between air and water and the number *k* (like the Laugier-Martin one), but also the system of axis and the expression of f_1 (two more initial objects) to get as a final object point *I*.

When I add a number as accuracy (figure 14 on the right), I can create another macro giving *O'*, virtual object seen from two points of our "Eye", *o* and *o'* (*oo'*= accuracy and *(oo')*//refraction line)

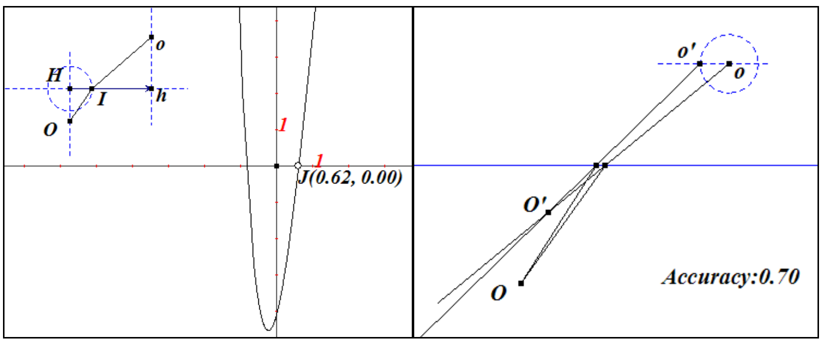

Figure 14

5.3.4. Testing my macros with Filou fishing in Cabri 2 Plus

We have chosen an accuracy of *0,005* and constructed the virtual object *O'* with my second macro. But the experiment led in figure 15 on the left is very disappointing because, if we use the traces of point *o* and *O* and drag *o* along a line parallel to the refraction line, we can observe that the construction is not stable at all when point *H* approaches point h. The explanation could be probably this one: we have a problem when *l* is nearly equal to zero so the coefficients of f_1 are a first source of error. The curve of f_1 that we used is a locus in Cabri, and *J* is the intersection point between a locus and the *x* axis: that is a second source of error. We can observe (figure 15 on the centre) that the virtual fish in red has nothing to see with the one we expect (we have chosen a position where the fisherman is just above the fish). So my investigation was limited by the limits of computation of the software and until now the geometrical Laugier-Martin tools still remains accurate, stable and therefore powerful (figure 15 on the right) for modelling constructions of refracted and dynamic objects.

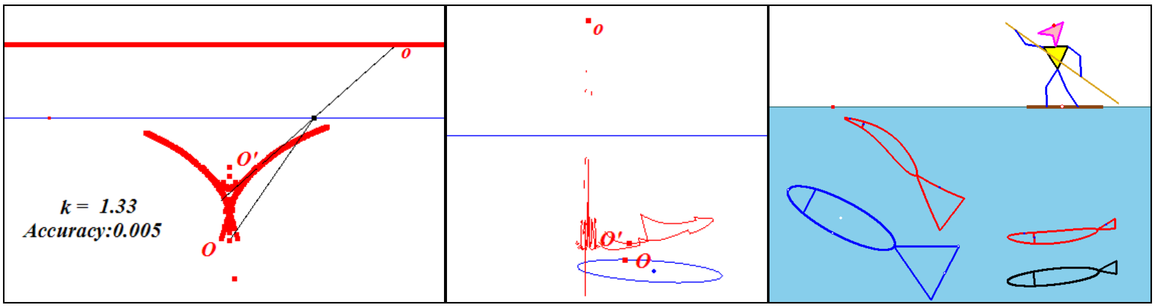

Figure 15

6. Conclusion

The use of different software packages to solve a problem can improve the process if the user is an expert in all of them. It enables unexpected points of view, and this can enhances the understanding of the concepts encountered during the research. From 2D to 3D, from geometric to algebraic frames, from numerical to graphic frames, from math to physics: a multi-software approach can enrich the interest, and justify the power of ICT in the daily practice of mathematics. Mathematics is experimental, especially with ICT, where experiments can be combined with the theoretical framework of my research ([5]). The process of modelling is also enhanced by a multi-software approach ([6]).

References

[1] M.A. Johsua, S. Johsua,1987, Les fonctions didactiques de l'expérimental dans l'enseignement scientifique (première partie), in Recherches en didactiques des mathématiques,8,(3), 231-266

[2] J.M., Laugier, 2001, *Cabri-Géomètre en sciences physique : Autour de Fermat, in Proceedings of Cabriworld 2001*, Montréal, Canada

[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., 2003, *Using the new tools of Cabri 2 Plus to teach functions, in Proceedings of the T3 International Conference, Nashville, USA*

[5] Dahan J.J., 2005, *La démarche de découverte expérimentalement médiée par Cabri-géomètre en mathématiques. Un essai de formalisation à partir de l'analyse de démarches de résolutions de problèmes de boîtes noires*, thèse de doctorat, Université Joseph Fourier, Grenoble. http://tel.archives-ouvertes.fr/tel-00356107/fr/

[6] Dahan J.J., 2008, *Modeling Cha Cha dance within the TI Nspire and Cabri environments in Proceeding*s TIME 2008, Tshwane University of Technology, South Africa