A Dynamic Approach of Analytic Geometry in 3D with TI N'Spire
Enhancing an Experimental Process of Discovery

Jean-Jacques Dahan
jjdahan@wanadoo.fr
Ph. D. Math and Computer Science, Grenoble University, France
Dynamic geometry research group, IREM of Toulouse,
Paul Sabatier University, Toulouse, France

Abstract: *Everybody is aware that using technology enhances a more experimental approach of mathematics: as Thurston and Wendelin (Fields medals), we think that maths are basically experimental. We will illustrate it within the TI N'Spire environment with examples related to the basic concept of coordinates in 3D geometry. We will show how 3D coordinates can be represented in parallel perspective (and especially in military perspective) in the "Geometry" and "Graphs" applications of TI N'Spire. A dynamic approach is possible thanks to the sliders that can enhance a deep understanding in these representations (according to Duval, [1]). Dynamic numbers can help students to reach the concept of variable (Jackiw, [10]) thanks to the possible "linking" between measurements and variable. Therefore, we will play with variables, sliders and loci to create representations of solids such as cylinders, cones and other surfaces (directly or with the new 3D plotter). A more interesting problem will be modelized: the unfolding of cylinders and cones: this problem originally solved with the Cabri environments, Cabri 2 Plus and Cabri 3D is solved here in using the possible connections between algebra and geometry in TI N'Spire following the principle stated by Laborde that "the teaching of mathematics must help students learn how to adequately use various representations and to move between them if needed.". The special language of experimental maths will be used as defined in my research work about the experimental process of discovery in maths mediated by technology: exploration, generative and validative experiments, conjecture, plausibility, experimental proof … ([5]) to help teachers to understand what to teach and what to assess when they practice experimental math supported by technology.*

1. Dynamic numbers and parallel perspectives

1.1. Parallel perspectives

A parallel perspective of the 3D Euclidian space is a representation of 3D objects of this space with 2D objects: these 2D objects are obtained in transforming the 3D objects with a projection of the 3D space with respect to a given direction on a given plane.
→ When the given plane modelizes the frontal plane (vertical plane parallel to the plane of the eyes of somebody standing up) of our real world, the parallel perspective is the usual parallel perspective used in text books called in French "**la perspective cavalière**" ([2]).
→ When the given plane modelizes the horizontal plane of our real world the parallel perspective is called "**the military perspective**" ([3] and [4]). This name was given in the French army to this realistic way of representing fortresses several centuries ago.

In these two representations, we use to display first the three axes *Ox*, *Oy* and *Oz* and the units on these axes (Figure 1).
→ In **perspective cavalière**, the frontal plane contains the axes *Oy* and *Oz* and all the objects belonging to this plane or a plane parallel to this one are represented in true dimensions. So the *x* axis is represented with a ray (called the vanishing line) which position depends from the direction of the projection. What we see in practice is that the unit on this axis is multiplied by a coefficient that can reduce or enlarge the depth of the objects (Figure 1 on the left).

→ In military perspective, the horizontal plane contains the *x* and *y* axes, and all the objects belonging to this plane or a plane parallel to this one are represented in true dimensions. The *z* axis is represented with a vertical ray (the direction of the projection is perpendicular to the plane of the eyes of somebody standing on the horizontal plane). What we see in practice is that the unit on this axis is multiplied by a coefficient that can reduce or enlarge the height of the objects (Figure 1 on the right).

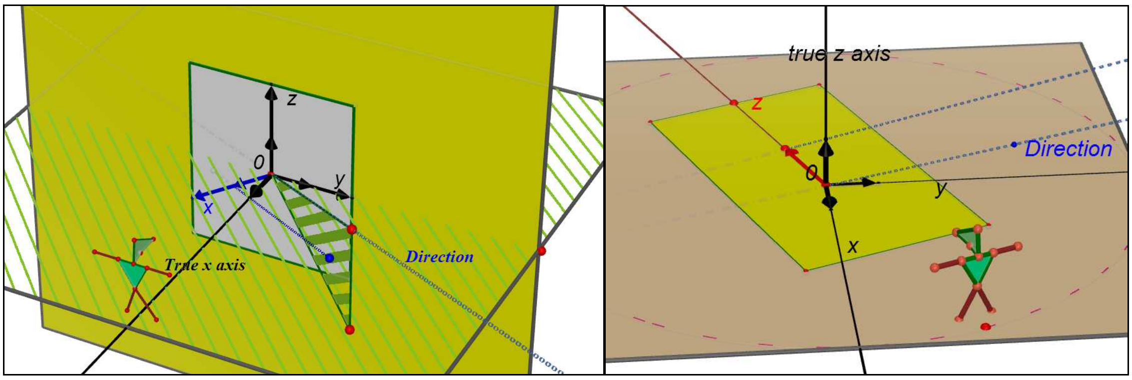

Figure 1: visualisation of the projections of the 2 perspectives presented

In Figure 2, with TI N'Spire, we have represented in a 2D figure the three axes of these two representations with the three vectors representing the units. But these representations are dynamic so in each of them we can change the coefficient of the perspective; in perspective cavalière, we can change the angle of the vanishing line, in military perspective we can rotate the system of axis (*Ox*,*Oy*) around *O* in the horizontal plane.

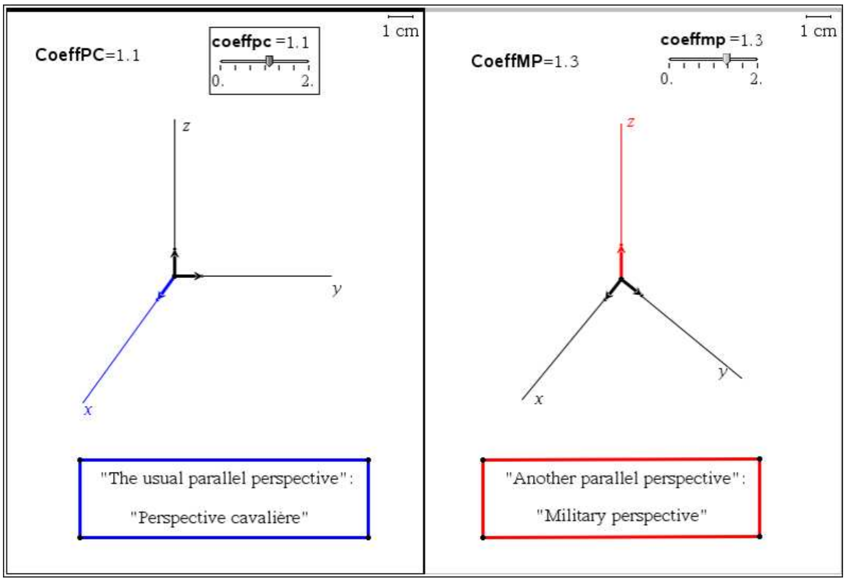

Figure 2: Two parallel perspectives

1.2. From numbers to dynamic numbers (in 3 steps)

→ In the two pages displayed in Figure 2 we have created a number with the tool "text". This number can been changed like on a piece of paper in erasing it and typing another one. It has been created to construct the unit vector with the tool "measurement transfer" and so, the length of these unit vectors changes when the number is changed.

→ Thanks to the tool "store variable", we can attach such a number to an algebraic variable: here in "**perspective cavalière**" we have stored this number in a variable called **CoeffPC**. The status of the number changes because with this name it can be used everywhere in the pages of the problem in which we are working. It become an interactive number. We will use this interactivity to change it onto a dynamic number.

→ Now we use the tool "slider" to create a slider commanding the variable **CoeffPC** and therefore the number contained in it (in the slider the name is changed onto **coeffpc** without any capital). We can, in the settings, choose the interval in which we want to command this number and also the increment. We can notice that these variables are displayed in **bold**: it is the way that the software tells us that these texts are recognized as variables. Using this slider allows us to change the length of the unit vector.

Remark: for an original work of modelling (modelling Cha Cha dance) see [6]

1.3. Representing with geometric constructions circles in these perspectives

Here is below in Figure 3, an example of representations of circles in these two perspectives showing where they are seen in true measurements and where they are represented by ellipses (obtained as loci)

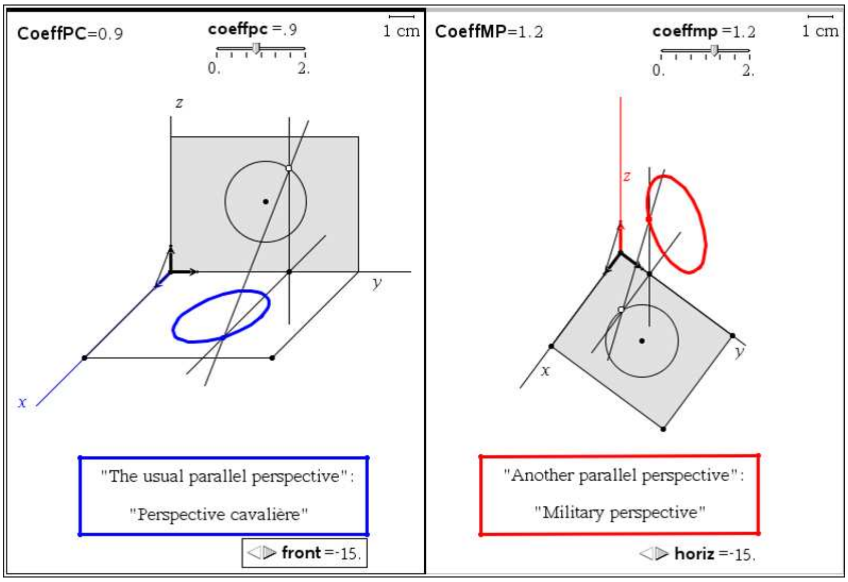

Figure3: Circles in different perspectives

In this introduction, we have presented the two perspectives we will use in this paper and the first technique to create dynamic numbers. Let us go now to analytic geometry where dynamic numbers will be used to create different solids in perspective after dealing with 3 dynamic coordinates.

2. Movable points given by their dynamic coordinates

2.1. Representing a free point commanded by sliders

In "**perspective cavalière**" we create 3 numbers; we apply to them the previous technique: store them in **xPC**, **yPC** and **zPC** and create the 3 sliders commanding these 3 variables. The settings will allow to generate numbers between -10 and 10 with an increment of 0.1. These 3 numbers are the 3 coordinates of a point *M*. We have used, "measurement transfer" to

construct points *yPC* and *zPC* and “enlargement” to construct point *xPC*. Other geometric constructions gave point M with its box. Thanks to these 3 sliders point M can be dragged in the 3 directions of the three axes.
We did the same thing in **military perspective**.
We have obtained what is displayed in Figure 4.

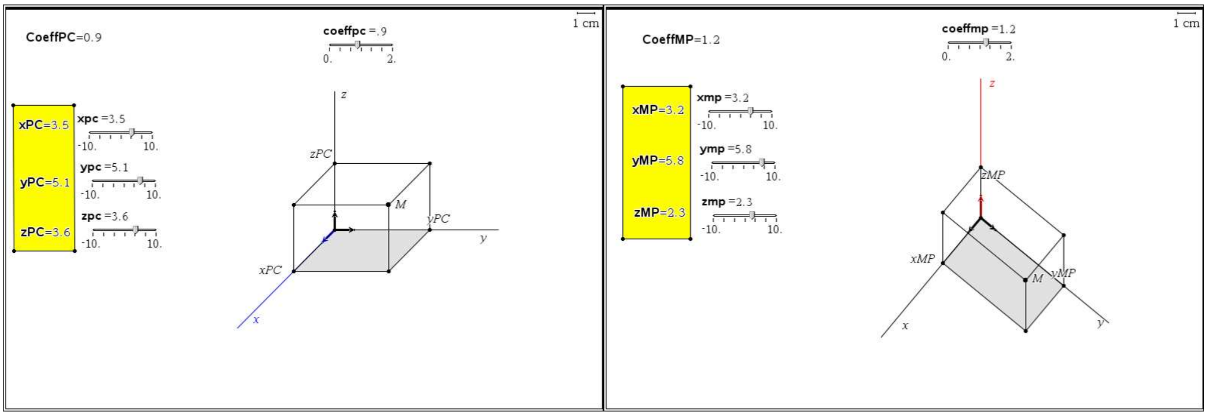

Figure 4: A point and its box commanded by 3 sliders

The very first consequence is the possibility to use these sliders to represent the trace of a motion of point *M* according to the 3 directions of the 3 axes. See below, Figure 5, where we have activated the “geometric trace” of point *M* before using the 3 sliders.

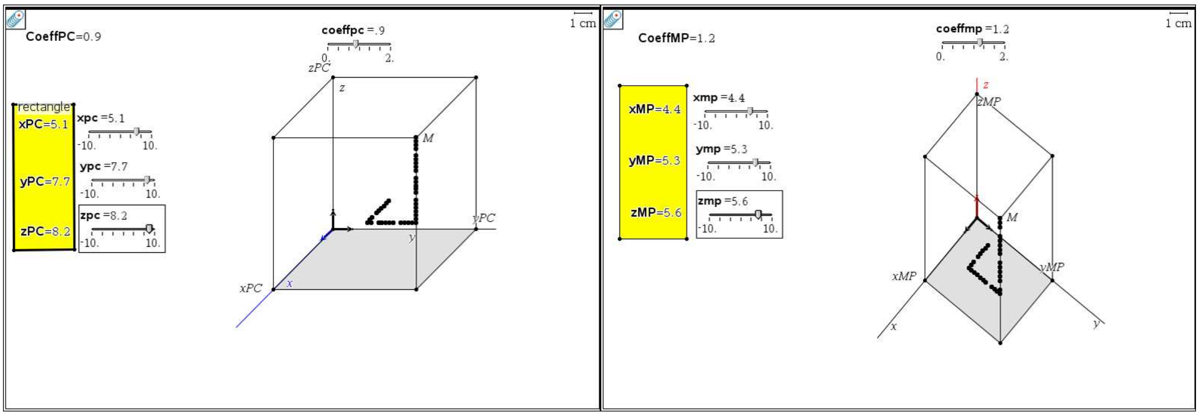

Figure 5: Traces along the directions of *Ox*, *Oy* and *Oz*

2.2. Traces of circles, cylinders and cones

We use the previous technique for 2 numbers, one stored in **r** (**r** for radius) and another stored in **a** (**a** for angle in degrees). Their sliders have special settings: **r** belongs to [0,10] and **a** belongs to [0,360]. Then with the tool “text” we create the 2 formulas “*r*cos(a)*” and “*r*sin(a)*” (not in bold because non connected to variables **a** and **r**); we use the tool “calculate” to evaluate these formulas for the values of variables **r** and **a**. We use the results we have obtained to construct the point *m* of the *xOy* plane (having these two numbers as coordinates). The trace of the circle having **r** as a radius centred in *O* in the *xOy* plane is obtained in activating the trace of point *m* and in dragging the cursor of the slider (Figure 6 on the left).
Same technique to create a variable **z** with its slider: the value of this variable allows us to define the point *S* on the *Oz* axis having this value as third coordinate. We create segment

[*Sm*], we activate its trace and we drag the cursor of its slider to obtain the trace of the cone having *Oz* as an axis and the previous circle as a base (Figure 6 on the right).

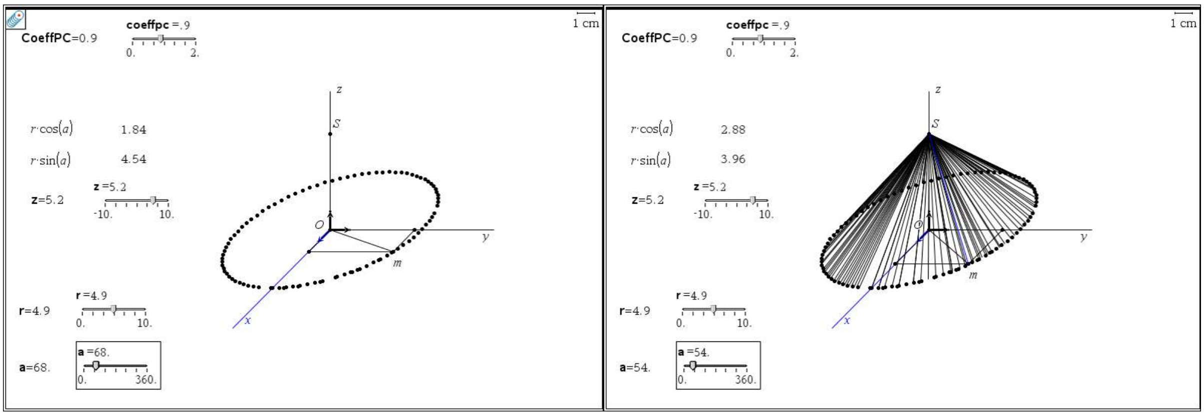

Figure 6: Traces of circles and cones in "**perspective cavalière**"

To create the trace of a cylinder (really a part of it), we construct segment [*mM*] where *M* is the translated point of *m* with respect to the translation mapping point *O* onto *S*. We activate the geometric trace of this segment and drag the cursor of the slider linked to **a** (Figure 7).

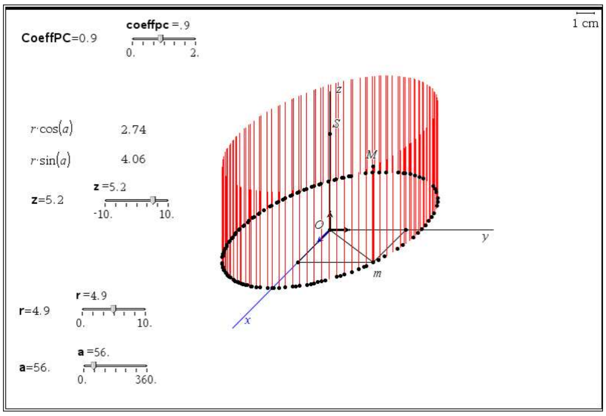

Figure 7: Trace of a cylinder in "**perspective cavalière**"

Figures 8 and 9 show the same work in "military perspective"; we can notice that the curve generated in the *xOy* plane is a circle seen in true dimension. That can be checked in drawing a circle superimposed to the trace of the curve

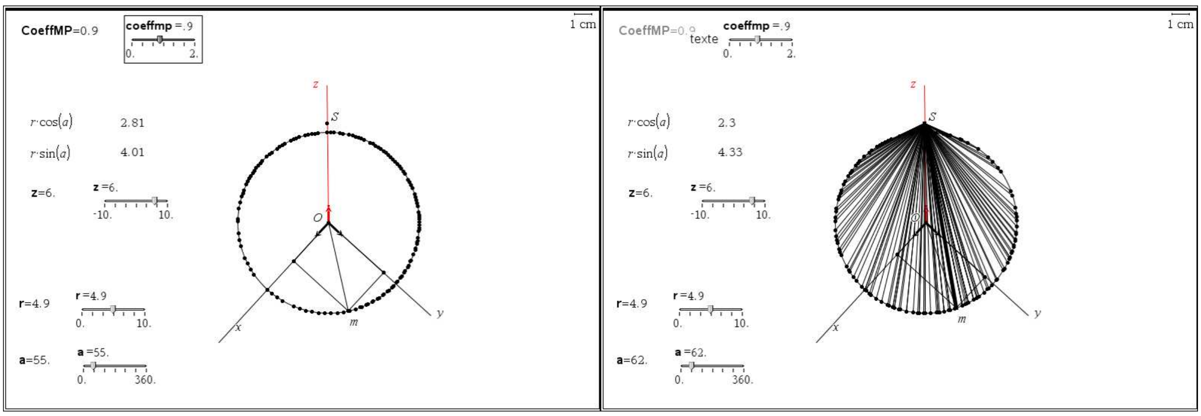

Figure 8: Traces of circles and cones in "**military perspective**"

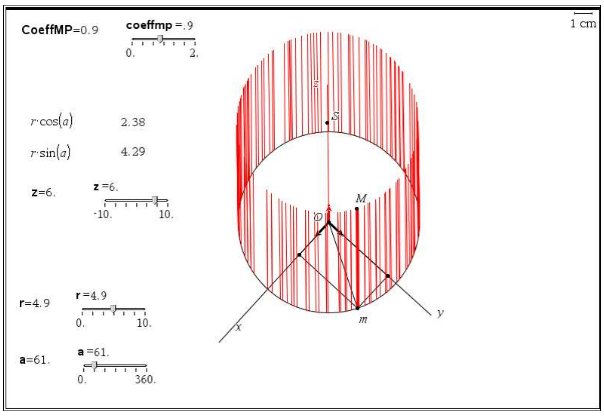

Figure 9: Trace of a cylinder in "**military perspective**"

The problem of a trace is that a trace is a dead line: it stays at the same place on the screen even if we change the position of the objects. So, to improve these representations we will improve our technique in order to replace our traces by loci.

2.3. Circles, cylinders and cones defined analytically

In Figure 10, the sliders of TI N'Spire are replaced by sliders created with segments. Each cursor is a point of each segment. The first slider commands the distance between the first point and the cursor of the first segment: this distance is attached to **ra**. The second one commands in the second segment the distance between the first point and the cursor (maximum 3.6 cm): this distance is multiplied by 100 to obtain a number attached to **an** (**an** has a maximum value of 360).

*"r*cos(a)"* and *"r*sin(a)"* are evaluated for r = **ra** and a = **an**.

The circle (the base) is obtained as the locus of point m when the second cursor moves on the second segment.

The cone is obtained as the locus of segment [Sm] when the second cursor moves on the second segment.

The cylinder is obtained as the locus of segment [mM] when the second cursor moves on the second segment.

The upper circle is obtained as the locus of point M when the second cursor moves on the second segment.

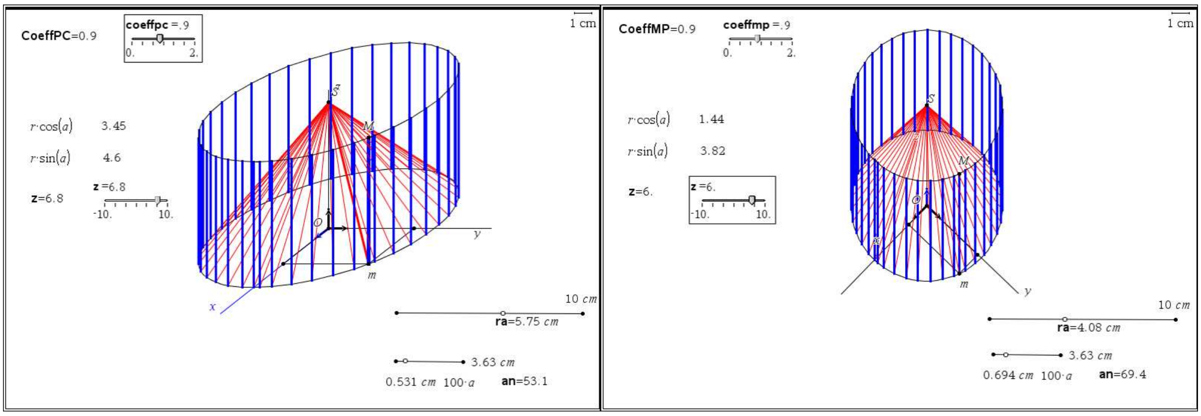

Figure 10: Circles, cylinders and cones with loci

In Figure 10, all the representations are dynamic thanks to the sliders and also because the position of Ox (in the left) can be changed. S can also be dragged in the two perspectives.

3. Modelling the unfolding of cylinders and cones

3.1. Unfolding a cylinder in military perspective

Our idea is to model a cylinder with a prism having an important number of faces. For the representation, we choose the military perspective because the base of the prism is represented in true dimensions. The only difficulty is to unfold a regular polygon in using a slider. The technique we propose is the following one ([9]):

We start with a pentagon, and later, in changing only one parameter, we will move onto a regular polygon with 20 sides.

Stage 1 (Figure 11 on the left): in **military perspective**, we have two sliders one for the coefficient of the perspective and another one for variable **z** which commands the position of point *S* on the *Oz* axis. (*C*) is a circle centred on *C* (which lies on the positive part of the *x* axis). We measure the perimeter *l* of this circle and with the displayed formula "*l*/5" we evaluate the fifth of this perimeter. We transfer this number on (*C*) from *O* to obtain *L*; *F* is the reflected point of *L* with respect to *Ox*. Therefore, [*OL*] and [*OF*] are two sides of a pentagon inscribed in (*C*).

The dotted circle centred in *O* and passing through *F* crosses *Oy* in *G*. We have created the ray passing through *O* and the midpoint of [*FG*]. This ray crosses the dotted circle in I. At last we have constructed the red arc included in the dotted circle and passing through *F*, *I* and *G*.

Stage 2 (Figure 11 on the right): we display the length of the red arc; we create a slider commanding a new variable called **k** varying between 0 and 1. With the formula "*k.d*" evaluated for *k* = **k** and *d* equal to the length of the red arc, we obtain a number transferred on the dotted circle from *F* to obtain a red point (*P1*) and a red segment [*OP1*] that can be animated from [*OF*] to [*OG*]. Another red segment, [*OP1'*] is created as the reflected one of the first one with respect to *Ox*. We have modelled the unfolding of the two first sides of a regular pentagon. Finally, this folding-unfolding is commanded by slider "**k**".

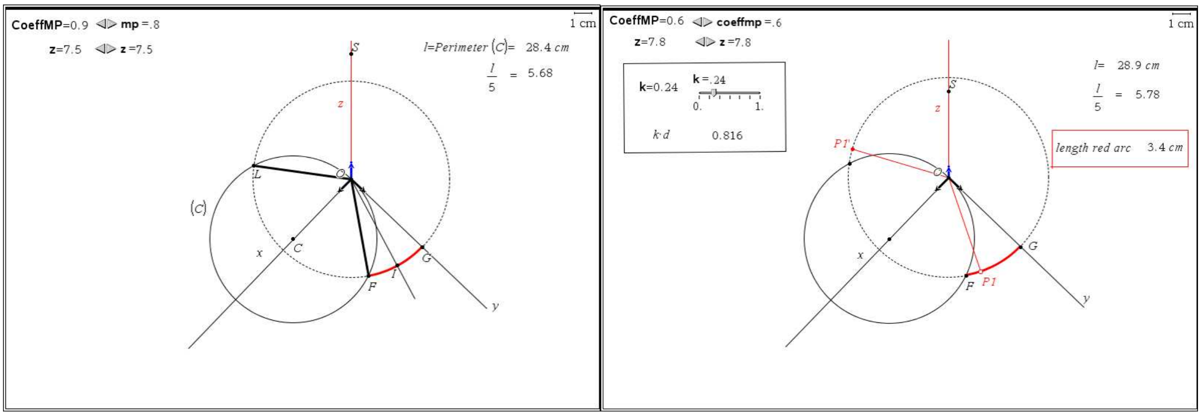

Figure 11: Unfolding a cylinder (stages 1 and 2)

Stage 3 (Figure 12 on the left): we present now the technique to obtain the other sides of the unfolded pentagon. We use the perpendicular bisector of [*OP1*] to reflect [*OP1'*] and obtain [*P1P2*]. [*P1'P2'*] is obtained in reflecting [*P1P2*] with respect to *Ox*. We iterate this process and therefore we create the unfolded pentagon with points *P2'*, *P1'*, *O*, *P1*, *P2*, *P3* and the segments joining theses points. Point *P3'* is not necessary to fold and unfold the pentagon but we construct it because really we want to unfold a polygon with more than five sides. The folding and the unfolding can be modelled with the **k** slider.

Stage 4 (Figure 12 on the right): we change the formula "*l*/5" onto the formula "*l*/10". All the data are instantaneously refreshed as well as the complete figure. Now we are on the way to model the unfolding of a regular decagon. So, to do that, we iterate the previous process to construct points from *P1* to *P7* and points from *P1'* to *P7'*.

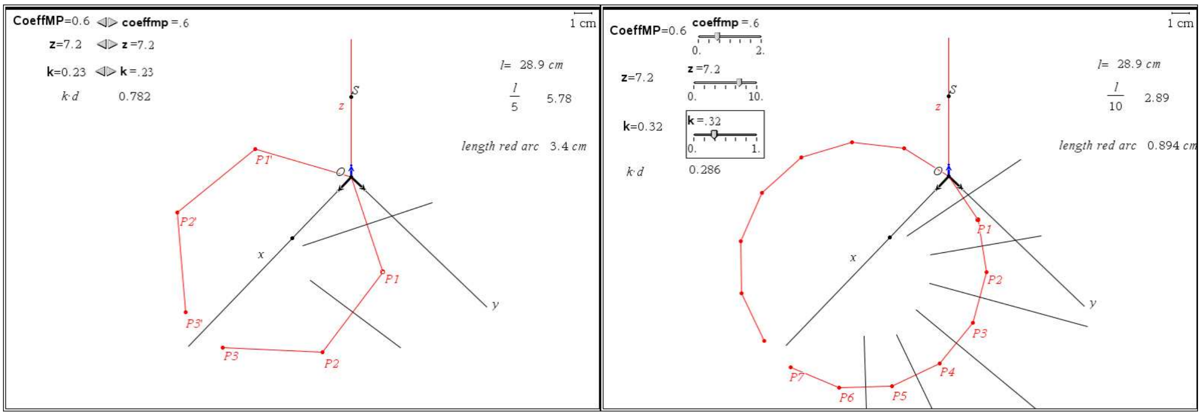

Figure 12: Unfolding a cylinder (stages 3 and 4)

Stage 5 (Figure 13 on the left and on the right): we change the formula "*l*/10" onto the formula "*l*/20". All the data are instantaneously refreshed as well as the complete figure. Now we are on the way to model the unfolding of a regular polygon having 20 sides. So, to do that, we iterate the previous process to construct points from *P1* to *P9* and points from *P1'* to *P9'* and the consecutive segments joining these points. If we change **k** with the slider and reach 0, we obtain the folded polygon (Figure 13 on the left). If we change **k** with the slider and reach 1, we obtain the unfolded polygon which is segment [*P9'*,*P9*] included in the *y* axis (Figure 13 on the right).

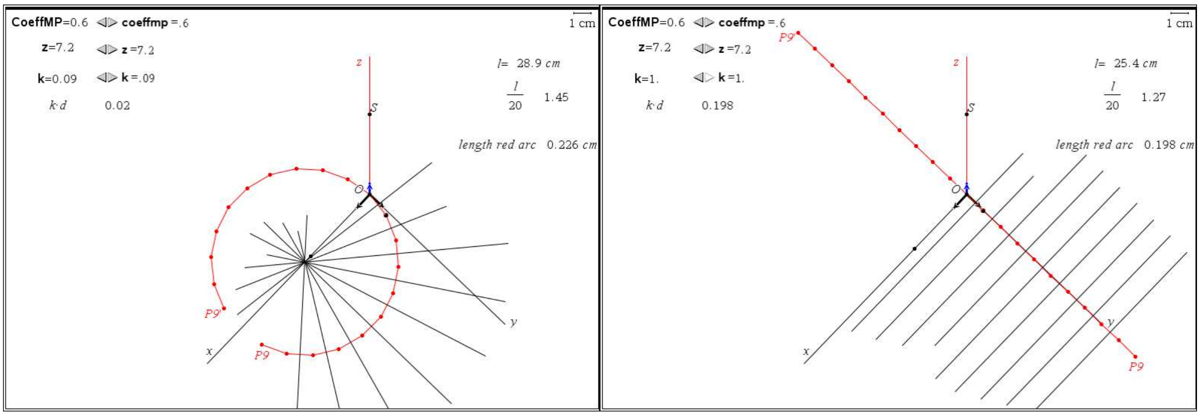

Figure 13: Unfolding a cylinder (stages 5)

Stage 6 (Figure 14 on the left and on the right) we translate all these points with the translation mapping point *O* onto point *S* (Figure 14 on the left). And then we create the 20 yellow quadrilaterals modelling the 20 faces of the prism we want to represent in military perspective (Figure 14 on the right). If we use the slider commanding **k**, we can see the folding and the unfolding of this prism. Now we need to design this file in order to provide an acceptable modelling of the cylinder.

Stage 7 and final stage (Figure 15): we have only to change the attributes of the quadrilateral in choosing the gray colour for the colour of the lines and the points. So, the edges are less perceived and the prism we have constructed can be a good visual model for our cylinder.

Using the **k** slider allows us to visualize the folding and the unfolding of a cylinder in **military perspective**.

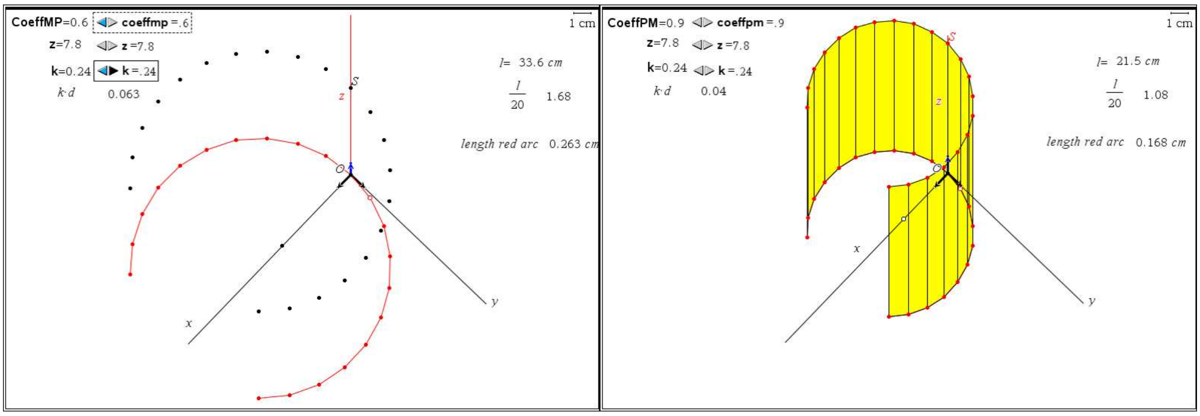

Figure 14: Unfolding a cylinder (stages 6)

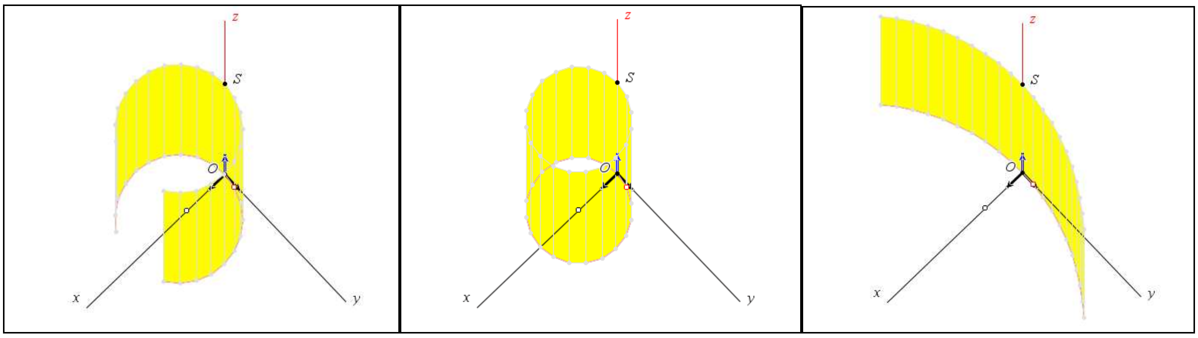

Figure 15: Unfolding a cylinder (final stage)

3.2. Unfolding a cone in military perspective ([8])

3.2.1. The choice of the military perspective:

As shown in Figure 16, if we squash a cone after cutting it along a generatrix, we can see that the base moves from a circle to an arc having as a length the perimeter of this base; the reason is that all the points of the new base are at equal distance from the summit and therefore this curve is included in the intersection between a sphere centred on the summit and having the generatrix as a radius and the *xOy* plane (we know that such a curve is a circle). As we can see in the right picture of Figure 16, the centre of this arc is necessarily the projection of the summit of the cone in the *xOy* plane. In **military perspective**, all the constructions in the *xOy* plane are done in true dimensions so it will be easy to construct this arc and therefore to model the unfolding of a cone as it will be done in the following part.

Figure 16: Squashing a cone

3.2.2. The stages of the modelling with TI N'Spire

Stage 1 (Figure 17): in **military perspective** we create first the base which is a circle centred on point *I* lying on the *x* axis. A number is created, stored in **h** and commanded by a slider (range between 0 and 10). This number represents the height of our cone. This height is represented on the *z* axis with point *H* in enlarging the unit point of this axis (enlargement centred in *O* and having **h** as a scale factor). The summit *S* of the cone is obtained in translating point *I* with the translation mapping *O* onto *H*. After that we create a generatrix of the cone [*SG*] (*G* belongs to the base). The cone is modelled by the locus of [*SG*] when *G* moves on the base. We measure the radius of the base (we can call it "*r*" but here "*r*" is only a text because not in bold). We measure the perimeter of the base (we can call it "*perim*" but here "*perim*" is only a text because not in bold). We create the text "$\sqrt{h^2+r^2}$" and we use it as a formula: we evaluate this expression for h = **h** and the current value of *r* to obtain the true dimension of *SO* (length of a generatrix) that we call *a* ("*a*" is also a text).

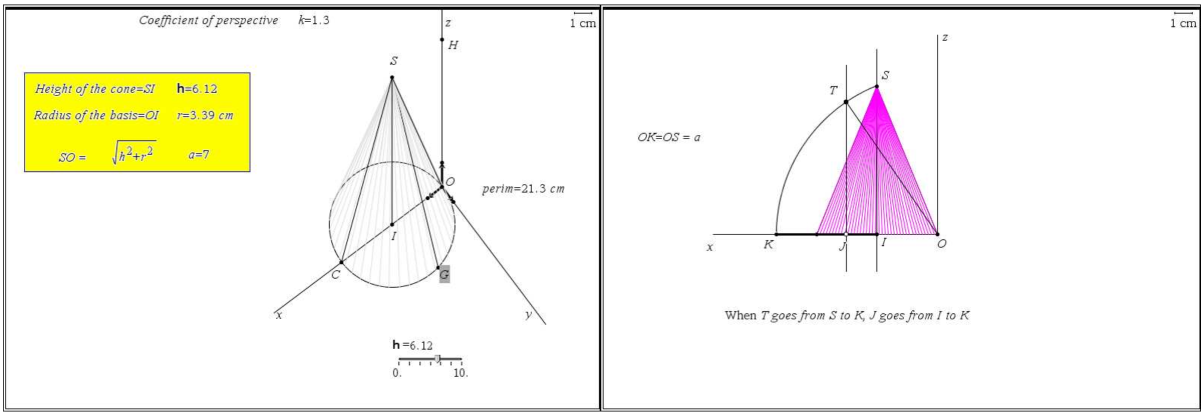

Figure 17: Unfolding a cone (stage 1)

Let us imagine that we cut the cone along generatrix [*SC*] and we squash the cone in rotating [*OS*] around the *y* axis. *S* will moves along a quarter of circle. In Figure 17 on the right, we see that this generatrix (we call it [*OT*]) has its projection *J* on the *x* axis moving from *I* to point *K* such as *OK* = *a*. Our trick is to command point *T* in dragging a point J on segment [IK]. We need to evaluate distance *JT* in order to construct *T*. The interest of this construction is also that *J* will be the centre of the squashed base (the arc having as a length the perimeter of the base of the cone.

Stage 2 (Figure 18 on the left)

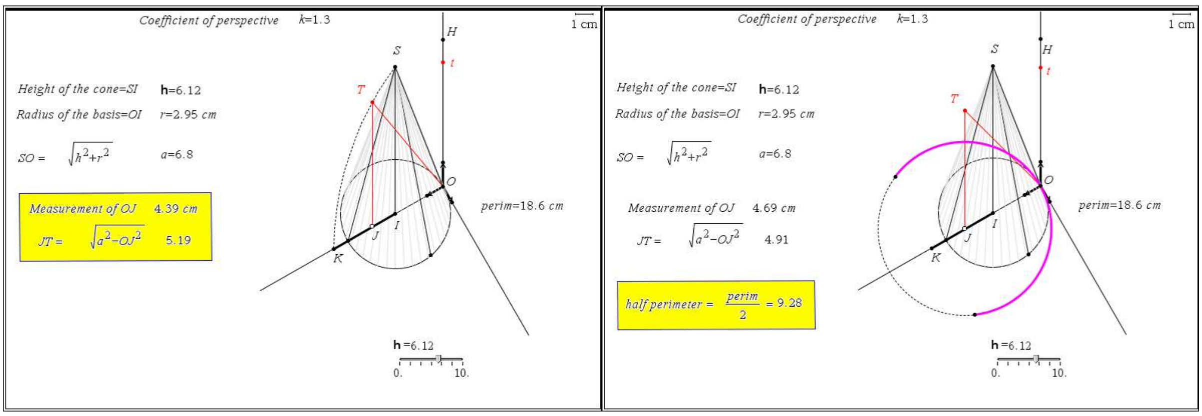

Figure 18: Unfolding a cone (stages 2 and 3)

We create K in transferring a on the x axis, the segment $[IK]$ and a point J on it. We measure OJ (true dimension in our representation); we create the text "$\sqrt{a^2-OJ^2}$" which will be used as a formula. We evaluate this formula to obtain the true dimension of JT. We construct t on the z axis with an enlargement of the unit point (centre in O and scale factor JT). Eventually, T is obtained as the translated point of J with the translation mapping O onto t.

Stage 3 (Figure 18 on the right)

We evaluate half of the perimeter of the base circle with the expression "$\frac{perim}{2}$". We create the circle centred on J and passing through O. Using the tools "measurement transfer" and "reflection", we create the two points of this circle defining with O the arc we obtain in squashing the cone. We create this arc.

Stage 4 (Figure 19)

We create segment $[TV]$ (V lies on the arc previously constructed). The unfolded cone is obtained as the locus of this segment when V moves on the arc (Figure 19 on the left). When point J reaches K, the cone is completely unfolded in the xOy plane (Figure 19 on the right).

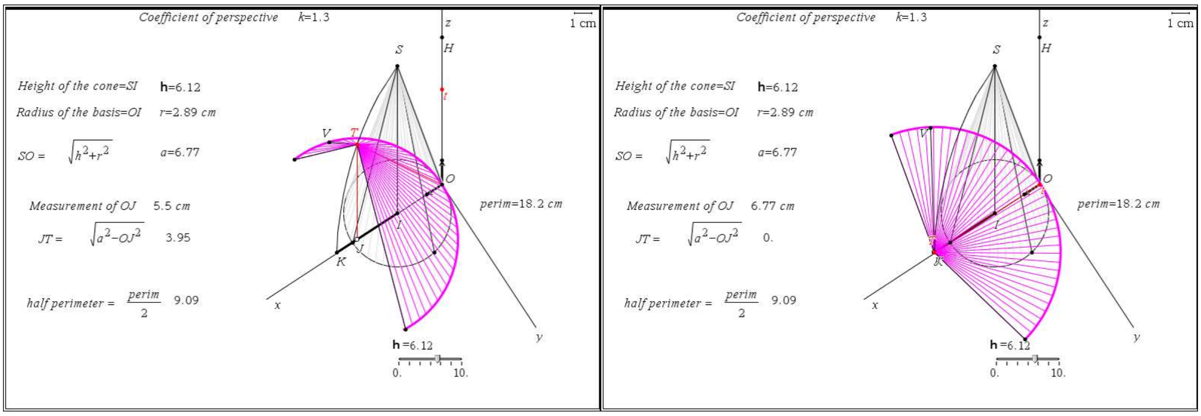

Figure 19: Unfolding a cone (stage 4)

3.3. Examples of investigations with the previous model

(see [5] for a theoretical background about experiments in maths with technology)

3.3.1. First investigation (Figure 20 on the left)

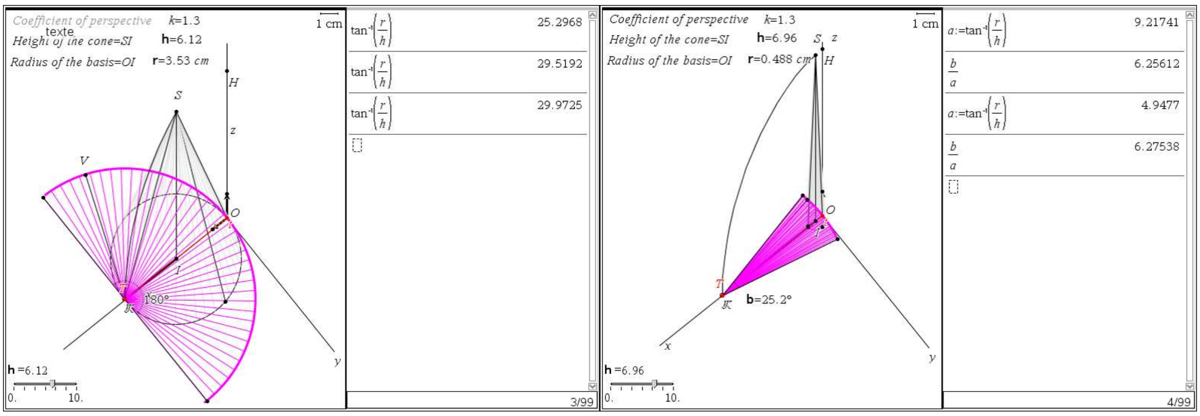

Figure 20: Investigations leading to two theorems

As we want to know the shape of a cone in order to unfold it into half a disk, we experiment in a way leading to a conjecture. We unfold the cone with our model, we measure the angle of

the unfolded cone and we drag for example I in order to change the base of the cone. We experiment (we generate data until we can interpret them positively) until we obtain half a disk (generative experiment).
We can validate the shape we have reached in measuring the angle of the unfolded cone. Then we can drag I until the number displayed for the angle reaches 180° if possible. We store the radius of the base circle in variable “**r**”.
Using the appropriate layout, we display the “Calculator” application where we evaluate in degrees angle *OSI* using the formula “ $\tan^{-1}(\frac{r}{h})$ ”. We can see than the different values generated with these experiments are approaching 30°. The conjecture telling that a cone is unfolded in half a disk only if the top angle is a 60° angle can be proven easily ([8]).

3.3.2. Second investigation (Figure 20 on the right)
We want to know what is the limit of the ratio between the angle of the unfolded cone and the angle *OSI*, when the ratio between **h** and **r** goes to infinity. The experiment in this case giving the data that can be interpreted as a conjecture is the following. We drag point I close to the origin to increase the value of the ratio $\frac{h}{r}$. We can also use the slider **h** in order to increase **h** if necessary. On the “Calculator” application, we store the angle *SOI* in variable **a**. In the “Graphs” page, we store the angle of the unfolded cone in variable **b**. So we can evaluate for each position, a and b/a. The last ratio is the one we are interested with. We can see, that the different data displayed in the “Calculator” page are approaching 2π. This conjecture obtained after a generative experiment can become more plausible after other experiments called validative experiments ([5]): we can increase more and more **h** and observe that b/a approaches more and more 2π. The proof of this conjecture can be done easily ([8]).

4. Representing surfaces *z = f(x,y)*

4.1. In the “Graphs” application (Figure 21)

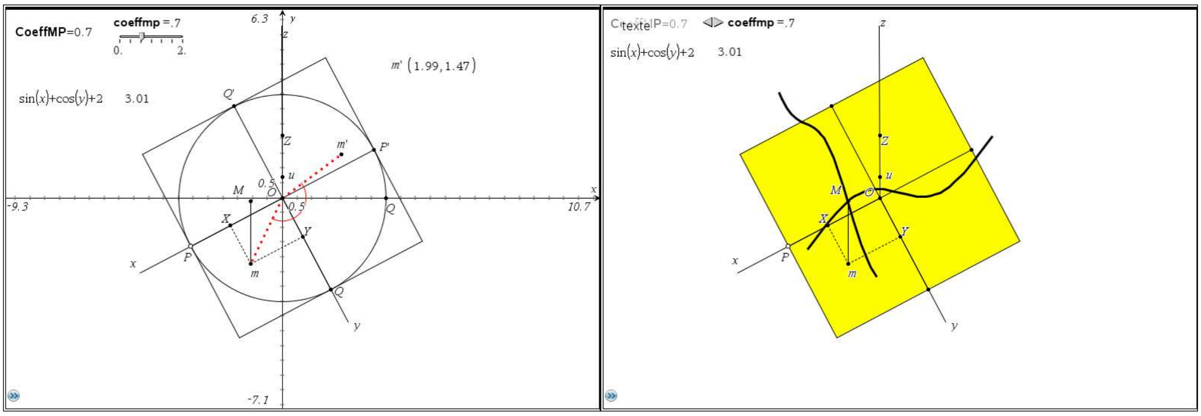

Figure 21: Surface *z(x,y) = sin(x)+cos(y)+3* in military perspective

We create the system of axes of the military perspective in a “Graphs” page with the origin on the origin of the system of axes of this page (Figure 21 on the left). The first construction is a circle centred in O and passing through a point P of Ox. Q is the intersection point between this circle and the positive part of the x axis of the system of axes of the page. We construct the square tangent to this circle passing through P which will be the domain of the function of

two variables we will represent later. We create the two segments [*PP'*] and [*QQ'*] and points *X* and *Y* respectively lying on these segments. Point *m* is defined with *X* and *Y*.
As we need the coordinates of *m* in the system of axes (*Ox*,*Oy*), we will use a special trick: we rotate *m* around *O* with the rotation mapping *P* onto *Q* to obtain *m'*. The coordinates of *m'* in the system of axes of the page given by the software are the coordinates of *m* in (*Ox*,*Oy*).
We create the text "*sin(x)+cos(y)+3*" which is the formula of the function *z(x,y)* = *sin(x)+cos(y)+3*. To represent its surface, we evaluate this expression for the coordinates of *m*. The result is *z(x,y)*. We construct *Z* on the *Oz* axis in transforming the unit point *u* of this axis with the enlargement centred in *O* and having *z(x,y)* as a scale factor. One point *M* of the surface is obtained in translating *m* with the translation mapping *O* onto *Z* (Figure 21 on the left).
To represent the two sections of this surface passing through *M* and parallel to the *xOz* and *yOz* planes, we only display the loci of M when *X* moves on [*PP'*] and *Y* moves on [*QQ'*] (Figure 21 on the right). Before doing that we have hidden most of the intermediate objects of the previous constructions.
To represent the surface with wires, we begin to represent a set of sections (20 sections) parallel to the first one. To do that, we create a set of 20 points on the second section and we display the locus of each point when *X* moves on [*PP'*] (Figure 22 on the left). We do the same thing for the other sections (Figure 22 on the right)

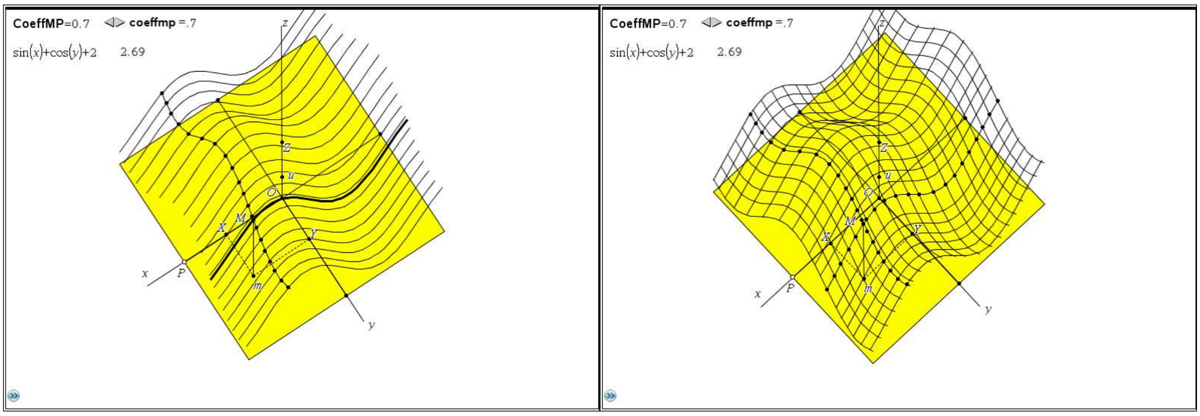

Figure 22: Surface *z(x,y)* = *sin(x)+cos(y)+3* in **military perspective**

Experimenting with this figure: it is possible in changing only the expression of *z(x,y)* to explore the possible shapes in relation with different expressions.

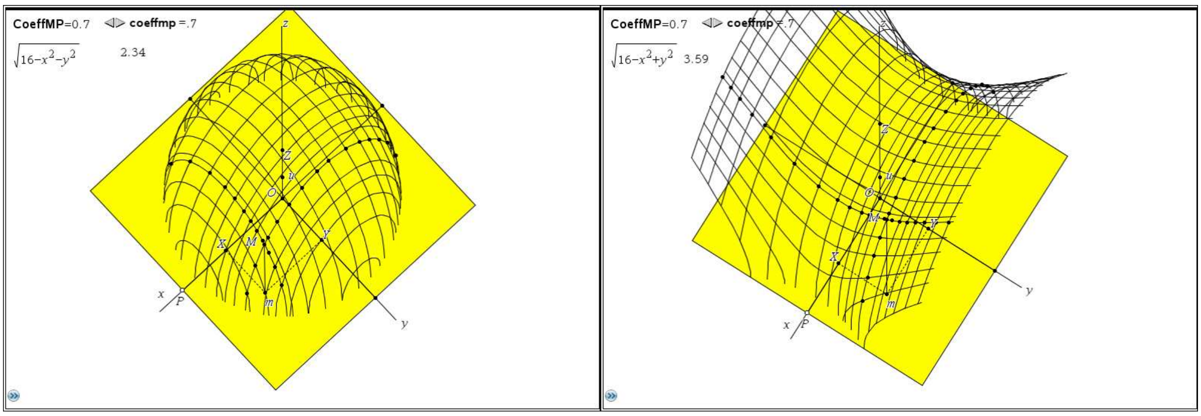

Figure 23: Exploring surfaces given by *z(x,y)* in military perspective

In Figure 23 on the left, we can conjecture that the equation is the equation of a sphere. On the right an hyperbolic paraboloid is displayed. A lot of other formulas can be experimented to lead to lots of conjectures.

Remark: changing the coefficient of perspective will change the point of view. When it approaches 0, the view gives the impression to the observer that he is just above the surface; when it increases, the view gives the impression to the observer that he is able to appreciate the difference of height (like somebody in an aircraft approaching mountains).

4.2. With the "3D Graphing" tool ([6])

In a Geometry page with the " 3D Graphing" option we create **z1**(x,y) = *sin(x)+cos(y)* and we obtain the surface of Figure 24 in the left. If we want to visualize better the positive and the negative part of the surface, one simple way is to display the horizontal plane in programming **z4**(x,y) = 0 (Figure 24 on the right). We have also changed the number of wires (resolution) in the attributes to obtain a more accurate intersection.

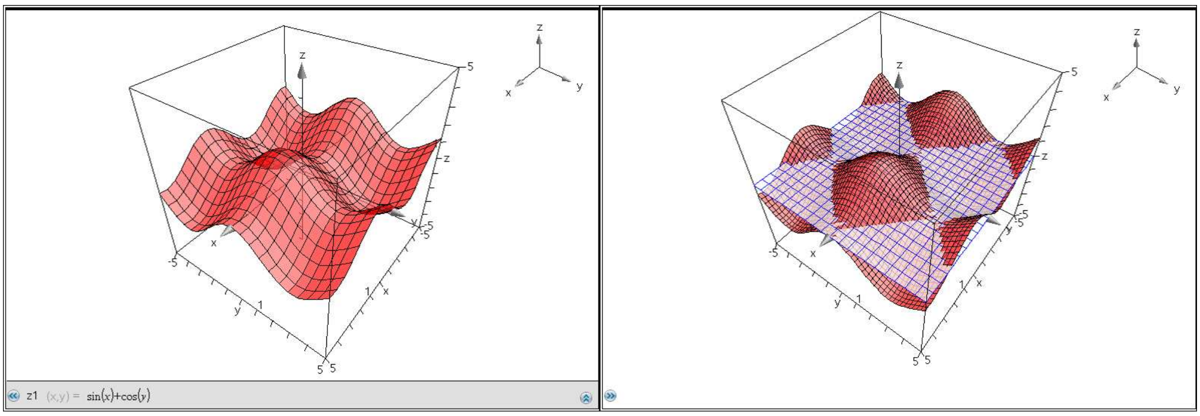

Figure 24: **z1**(x,y) = *sin(x)+cos(y)*

Here is in Figure 25 another way to represent separately the positive and the negative part of this surface.

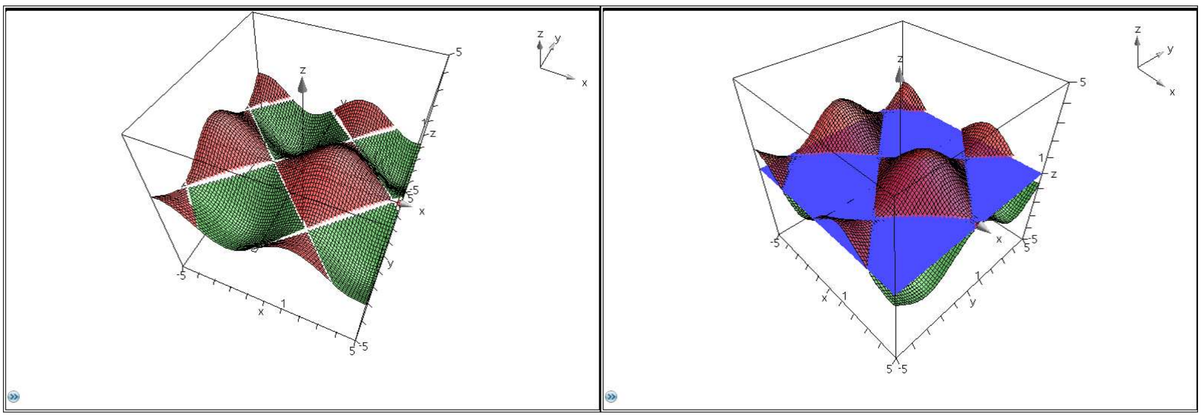

Figure 25: **z2**(x,y) = $(\sqrt{z1(x,y)})^2$ and **z3**(x,y) = $-(\sqrt{-z1(x,y)})^2$

We have programmed respectively the two functions:

z2(x,y) = $(\sqrt{z1(x,y)})^2$ and **z3**(x,y) = $-(\sqrt{-z1(x,y)})^2$. So, we obtain Figure 25 on the left where we have changed the attributes as we did previously (here the resolution is 75). In

Figure 25 on the right we have added the horizontal plane without wires and with a level of transparency equal to 0.
Remark : for the interest of such a function in teaching trigonometry ; see [7]

4. Conclusion

We have tried in this paper to show with various examples how the new TI N'Spire environment and especially the Geometry, the Graphs and the 3D Graphing applications can enhance a better approach of 3D concepts. The dynamic numbers, thanks to the tools "Store" and "Slider" allow to grab and drag variables and therefore allow a more heuristic approach of figures according to Duval ([1]). We have shown how easy it is to represent cylinders and cones and to model their unfolding in **military perspective**. Moreover, the model we have created had been used to explore experimentally some problems that cannot be explored in a paper and pencil environment to lead eventually to interesting conjectures. The experimental process of discovery was described quickly at this occasion ([5]). We have opened a window on an original use of the "3D Graphing" application thanks to the colours available in the handheld and the software. It becomes really enjoyable to enter the world of mathematics in using a tool designed like the tools used by students in their real life.

5. Bibliography

[1] DUVAL R., 1994, *Les différents fonctionnements d'une figure dans une démarche géométrique*, *Repères IREM* N° 17, 121-138.

[2] DAHAN J.J., 2001, Parallel perspective with Cabri, 13th Annual T^3 International Conference, Columbus (Ohio US).
http://www.irem.ups-tlse.fr/groupes/03mathinfo/percav/ppwithcabrids.htm

[3] DAHAN J.J., 2003, *Visualising functions of two variables with Cabri II Plus*, ATCM 2003, Chung Hua University, Hsin-Chu, (Taïwan). (Contact me to receive the paper)
http://epatcm.any2any.us/EP/EP2003/index.html

[4] DAHAN J.J., 2004, How to construct curves of functions of one variable, of parametric, polar and random functions. Surfaces in several perspectives, T3 conference, New Orleans (Louisiana, U.S.A.). (Contact me to receive the paper)

[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 (France).
http://tel.archives-ouvertes.fr/tel-00356107/fr/

[6] DAHAN J.J., 2008, *Modelling Cha Cha dance in using the "function" tools within the Cabri 2 Plus or the TI NSpire environment*, Tshwane University of Technology, Buffelspoort (South Africa).
http://rfdz.ph-noe.ac.at/fileadmin/Mathematik_Uploads/ACDCA/TIME2008/TIME08_contribs/Dahan/Dahan.pdf

[7] DAHAN J.J., 2009, *A dynamic practice of analytic geometry within the TI N'Spire environment. New perspectives for an experimental approach of teaching*, ICTMT-9, Metz (France). (Contact me to receive the paper)

http://ictmt9.org/contribution.php?id=127&lang=en

[8] DAHAN J.J., 2009, *Folding and Unfolding a Cone with Cabri 3D : a Process of Modelling Generating New Problems to Be Explored*, TIME 2010, Malaga (Spain). http://www.time2010.uma.es/Proceedings/

[9] DAHAN J.J., 2011, *Folding and unfolding 2D and 3D objects with Cabri 2D, 3D and TI N'Spire*, T3 conference, San Antonio (U.S.A.). (Contact me to receive the slideshow and the files)

[10] MACKRELL K., 2011, *Integrating number, algebra and geometry with interactive geometry software* in Proceedings of CERME 7 (in press)