

## AUTOLISP FUNCTION FOR UNFOLDING THE OBLIQUE CONE OR FRUSTUM OF OBLIQUE CONE, IN AUTOCAD

**Abstract:** In production areas dedicated to the manufacturing of small series or unique products, i.e. in sectors which require frequent calculations for geometric determinations (such as the case of obtaining the unfolded shape of oblique cones frustums from the joints of different tubes (pipes) of varying diameters and/or orientations it is useful to achieve their development within a short timeframe, accurately and with minimal effort. Regarding this, an AutoLisp function was created for the rapid development of an oblique cone frustum, capable of being used in the current working session of the drawing executed in AutoCAD, without requiring additional calculations or drawings.

**Key words:** AutoCAD, AutoLisp, oblique cone, frustum, unfolding.

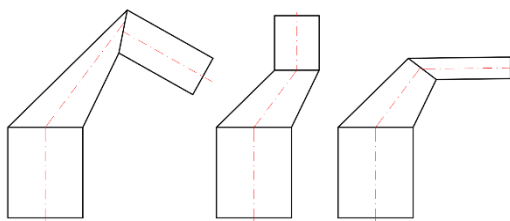
### 1. INTRODUCTION

The advantage of the implemented AutoLisp function (DCO) is that, used during the AutoCAD drawing session, it no longer requires additional calculations, measurements, or other data gathering actions about the cone from the user.

Based on the premise that, in the technical field, views or sections in overall drawings or in execution drawings are generally created to represent dimensions in their true size, the construction of oblique cones with an axis parallel to the projection plane (front plane) will be taken into account. Also, for the same reasons, the cone will be positioned so that its base is in a plane parallel with the horizontal plane (the projection of the base is a horizontal line segment).

Hence, there will be enough elements to facilitate the drawing of the unfolded shape (Figure 1).

It is not necessary that the characteristic elements of the cone frustum precisely define the outline of the geometric figure. It is enough to indicate the segments that make up the respective characteristic elements (axis, bases, or slant height). This facility is useful when the outline of the cone is drawn from segments placed in different layers or the characteristic elements have dimensions that exceed the outline of the cone or the cone frustum.

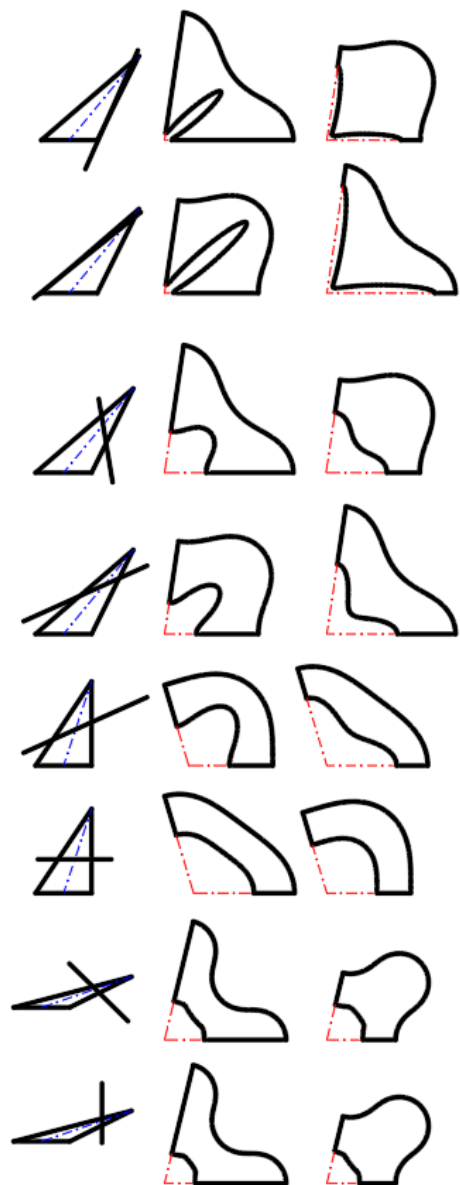


**Figure 1** Frustum of oblique cone used in the drawings.

### 2. FUNCTION DESCRIPTION

The following is an AutoLisp function created to unfold an oblique cone or an oblique cone frustum. The advantages of this function are represented by the following features: it presents the possibility to unfold

the conical surface in two ways - starting from the minimum length of the slant height or from the



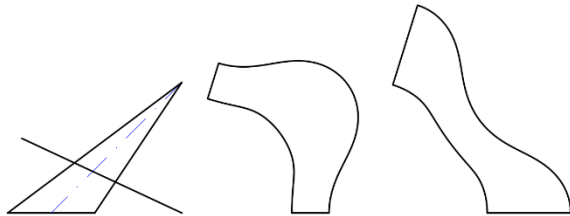
**Figure 2** Examples of unfolding the frustum of oblique cone.

maximum length (fact that has an impact on the manufacturing process in optimizing the consumption of material used); the high precision of the unfolding is controlled by the user depending on the technological limitations; the function collects its own calculation data solely through the user's accurate indication of the graphic elements of the cone; it can be run directly in the current session without leaving the work environment.

Figure 2 shows some theoretical situations to be carried out depending on the inclination of the axis of the cone trunk and the position of the sectioning plane.

From the input data required by the AutoLisp function, the length of the indicated slant height, the radius of the base circle, the position of the apex of the cone, the position of the center of the base will be determined.

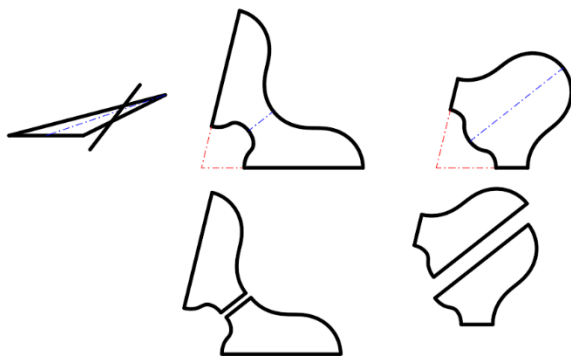
The user will enter the curve approximation precision, by indicating the number of points through which the base curve will be drawn. Directly, a first limitation arises from the working methodology in AutoCAD ("Requires an integer between -32768 and 32767"). The other limitations are technological, including the performance of the computer used.



**Figure 3** Two possibilities to unfold an oblique cone frustum.

In cases where the truncated cone or the cone itself is obtained from a single piece of material joined on a single side (slant heights), because the function offers the possibility of choosing the side with which to start unfolding (by indicating the larger or smaller portion of the slant height of the cone frustum), the material consumption can be optimized depending on the resulting geometry and the dimensions of the existing half-finished product (Figure 3).

In cases where the conical surface is obtained from two equal pieces achieved from the initial unfolded shape by cutting it in the middle (in the case of large pieces), the indication order of the slant heights does not matter.



**Figure 4** Unfolded shape made by two halves.

It is observed that no matter how the unfolding is obtained, cutting it through the middle produces two symmetrical halves in any of the situations (Figure 4).

### 3. THEORETICAL BASE OF CALCULATION

To obtain the cone unfolding, consider a certain number of points on the base circle, based on which the length of the cone's slant height will be calculated. This involves transforming the cone or the cone frustum into a pyramid or respectively, into a truncated pyramid with a number of side faces equal to that of the defined number of points.

Hence, the circle unfolding of the base will be a curve drawn through points, meaning that it will be composed of straight-line segments.

Determining the number of points can be done by two methods:

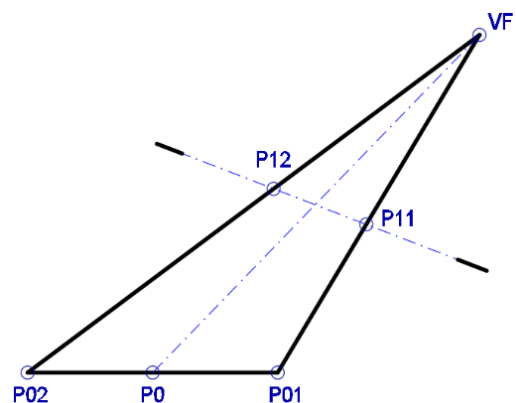
- directly, by accepting a certain predetermined number of points (100); this number can be modified by entering it from the keyboard in the specific dialogue sequence.
- indirectly, by indicating the size of the chord (segment) that makes up the curve of the unfolded base of the cone.

For enhanced accuracy and efficiency in calculations, it is recommended to directly input the number of points. Obviously, the higher the number, the closer the obtained outline will be to the real curve. In practice, the size of this number will be determined by technological limitations.

For the cone frustum, to draw on the unfolded shape the intersection curve between the section plane and the oblique cone, the point of intersection between the section plane and each slant height corresponding to the current point on the base of the cone will be determined. The distance from the apex of the cone to the determined point of intersection is then computed.

The following presents the theoretical aspects that were used in the calculation of the necessary parameters for tracing the unfolding.

In the following, the notations are correlated with the working methodology in AutoCAD [3]. The default working plane is  $xOy$  and the third dimension is given by the  $Oz$  axis, perpendicular to the  $xOy$  plane.



**Figure 5** Oblique cone sectioned by an inclined plane.

Consider an oblique cone with the base a circle located in the  $xOz$  plane, and the  $V_F P_0$  axis parallel to the  $xOy$  plane, where  $P_0$  is the center of the base circle. It is intersected by a plane perpendicular to the  $xOy$  plane. Let the triangle  $V_F P_{01} P_{02}$  be the projection of the cone on the plane  $xOy$  and  $P_{11}(x_{11}, y_{11}, z_{11})$ ,  $P_{12}(x_{12}, y_{12}, z_{12})$ , the points where the section plane intersects the slant heights  $V_F P_{01}$  and  $V_F P_{02}$  respectively (Figure 5).

The equation of the plane that passes through the points  $P_{11}$ ,  $P_{12}$  and is perpendicular to the  $xOy$  plane will be determined or, in other words, the equation of the plane that contains the points  $P_{11}$  and  $P_{12}$  and is parallel to the  $Oz$  axis will be computed.

We are interested in the case where the plane is determined by a point ( $P_{11}$  or  $P_{12}$ ) and two non-collinear vectors [1]. Consider the point  $P_{11}(x_{11}, y_{11}, z_{11})$  and the two vectors  $\overrightarrow{P_{11}P_{12}}$  and  $\overrightarrow{Oz}$ , which belong to the orthonormal basis  $\{i, j, k\}$  and which are in the form:

$$\overrightarrow{P_{11}P_{12}} = (x_{12} - x_{11}) \cdot \vec{i} + (y_{12} - y_{11}) \cdot \vec{j} + (z_{12} - z_{11}) \cdot \vec{k} \quad (1)$$

$$\overrightarrow{Oz} = 1 \cdot \vec{k} \quad (2)$$

The equation of this plan has the following form:

$$\begin{vmatrix} (x - x_{11}) & (y - y_{11}) & (z - z_{11}) \\ (x_{12} - x_{11}) & (y_{12} - y_{11}) & (z_{12} - z_{11}) \\ 0 & 0 & 1 \end{vmatrix} = 0 \quad (3)$$

Solving equation (3), we get:

$$\begin{aligned} & (x - x_{11}) \cdot (y_{12} - y_{11}) \cdot 1 + (y - y_{11}) \cdot (z_{12} - z_{11}) \cdot 0 + (x_{12} - x_{11}) \cdot 0 \cdot (z - z_{11}) - 0 \cdot (y_{12} - y_{11}) \cdot (z - z_{11}) - 0 \cdot (z_{12} - z_{11}) \cdot (x - x_{11}) - (x_{12} - x_{11}) \cdot (y - y_{11}) \cdot 1 = 0 \rightarrow \\ & \rightarrow (x - x_{11}) \cdot (y_{12} - y_{11}) - (x_{12} - x_{11}) \cdot (y - y_{11}) = 0 \rightarrow \\ & \rightarrow x \cdot y_{12} - x \cdot y_{11} - x_{11} \cdot y_{12} + x_{11} \cdot y_{11} - y \cdot x_{12} + x_{12} \cdot y_{11} + y \cdot x_{11} - x_{11} \cdot y_{11} = 0 \rightarrow \\ & \rightarrow x \cdot (y_{12} - y_{11}) + y \cdot (x_{11} - x_{12}) - x_{11} \cdot y_{12} + x_{12} \cdot y_{11} = 0 \rightarrow \end{aligned}$$

$$\rightarrow A_1 \cdot x + B_1 \cdot y + D_1 = 0, \quad (4)$$

where:

$$\begin{aligned} A_1 &= (y_{12} - y_{11}) \\ B_1 &= (x_{11} - x_{12}) \\ D_1 &= x_{12} \cdot y_{11} - x_{11} \cdot y_{12} \end{aligned}$$

To determine the intersection between the slant height corresponding to the current point on the base circle and the section plane, the equation of the slant height determined by the peak and the current point will be written in the form of the intersection of two planes. Each of these contains the two points of the line (the slant height) and is parallel to one axis of the coordinate system: one with the  $Ox$  axis and the other with the  $Oz$  axis.

Let  $V_F(x_F, y_F, z_F)$  and  $P_{0i}(x_{0i}, y_{0i}, z_{0i})$  be the points that define the current slant height.

The general equation of this line is:

$$\frac{x - x_F}{x_{0i} - x_F} = \frac{y - y_F}{y_{0i} - y_F} = \frac{z - z_F}{z_{0i} - z_F}$$

$$\begin{aligned} a) \quad \frac{y - y_F}{y_{0i} - y_F} &= \frac{z - z_F}{z_{0i} - z_F} \rightarrow y \cdot (z_{0i} - z_F) - y_F \cdot (z_{0i} - z_F) = z \cdot (y_{0i} - y_F) - z_F \cdot (y_{0i} - y_F) \rightarrow \\ & \rightarrow y \cdot (z_{0i} - z_F) - y_F \cdot z_{0i} + y_F \cdot z_F = z \cdot (y_{0i} - y_F) - z_F \cdot y_{0i} + z_F \cdot y_F \rightarrow \\ & \rightarrow y \cdot (z_{0i} - z_F) - z \cdot (y_{0i} - y_F) - y_F \cdot z_{0i} + z_F \cdot y_{0i} = 0 \rightarrow \end{aligned}$$

$$\rightarrow B_2 \cdot y + C_2 \cdot z + D_2 = 0, \quad (5)$$

where:

$$\begin{aligned} B_2 &= z_{0i} - z_F \\ C_2 &= y_F - y_{0i} \\ D_2 &= z_F \cdot y_{0i} - y_F \cdot z_{0i} \end{aligned}$$

Relation (5) is the equation of the plane containing the slant height  $V_F P_{0i}$  and is parallel to the  $Ox$  axis.

$$\begin{aligned} a) \quad \frac{y - y_F}{y_{0i} - y_F} &= \frac{x - x_F}{x_{0i} - x_F} \rightarrow y \cdot (x_{0i} - x_F) - y_F \cdot (x_{0i} - x_F) = x \cdot (y_{0i} - y_F) - x_F \cdot (y_{0i} - y_F) \rightarrow \\ & \rightarrow y \cdot (x_{0i} - x_F) - y_F \cdot x_{0i} + y_F \cdot x_F = x \cdot (y_{0i} - y_F) - x_F \cdot y_{0i} + x_F \cdot y_F \rightarrow \\ & \rightarrow y \cdot (x_{0i} - x_F) + x \cdot (y_F - y_{0i}) - y_F \cdot x_{0i} + x_F \cdot y_{0i} = 0 \rightarrow \end{aligned}$$

$$\rightarrow A_3 \cdot x + B_3 \cdot y + D_3 = 0, \quad (6)$$

where:

$$\begin{aligned} A_3 &= y_F - y_{0i} \\ B_3 &= x_{0i} - x_F \\ D_3 &= x_F \cdot y_{0i} - y_F \cdot x_{0i} \end{aligned}$$

Relation (6) is the equation of the plane containing the slant height  $V_F P_{0i}$  and is parallel to the  $Oz$  axis.

The intersection of the three planes given by equations (4), (5), (6) gives us the point of intersection between the current slant height and the sectioning plane. That is, the solution of the system of equations represents the coordinates of the point of intersection.

We have the system:

$$\begin{cases} A_1 \cdot x_{1i} + B_1 \cdot y_{1i} + D_1 = 0 \\ B_2 \cdot y_{1i} + C_2 \cdot z_{1i} + D_2 = 0 \\ A_3 \cdot x_{1i} + B_3 \cdot y_{1i} + D_3 = 0 \end{cases}$$

Solving it, the following solutions result:

$$\begin{aligned} x_{1i} &= \frac{B_3 \cdot D_1 - B_1 \cdot D_3}{A_3 \cdot B_1 - A_1 \cdot B_3}; \\ y_{1i} &= \frac{A_1 \cdot D_3 - A_3 \cdot D_1}{A_3 \cdot B_1 - A_1 \cdot B_3} \\ z_{1i} &= \frac{B_2}{C_2} \cdot \frac{D_1 \cdot A_3 - D_3 \cdot A_1}{A_3 \cdot B_1 - A_1 \cdot B_3} - \frac{D_2}{C_2} \end{aligned}$$

This results in the intersection point  $P_{1i}(x_{1i}, y_{1i}, z_{1i})$  of the current slant height with the sectioning plane.

Given that the circle is in a plane parallel to the  $xOz$  plane, the  $y$  coordinate is a constant of the form:  $Y_{1i} = Y_{0i} = Y_0$ .

Noting with  $r$  the radius of the base circle, we have  $P_0P_{01}=P_0P_{0i}=r$ . If  $\theta_i$  is the angle of the current point on the circle, then the  $x$  and  $z$  coordinates of the current point will have the following values:

$$\begin{cases} x_{0i} = x_0 + r \cdot \cos \theta_i \\ z_{0i} = z_0 + r \cdot \sin \theta_i \end{cases} \quad (7)$$

In triangle  $\Delta P_0P_{01}P_{0i}$  we have:

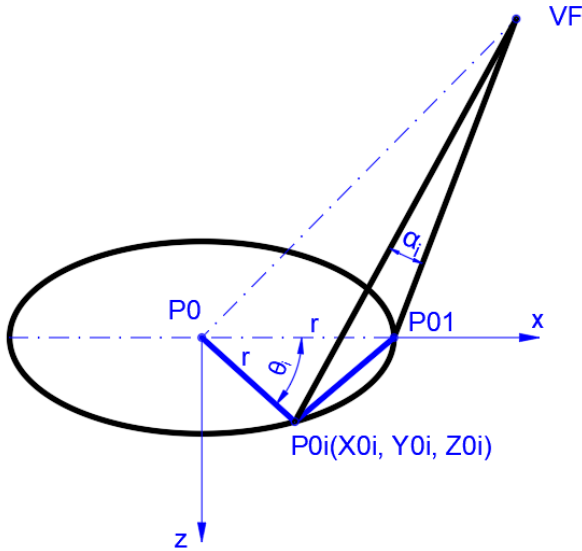
$$\angle P_0P_{01}P_{0i} = \frac{\pi - \theta_i}{2} \text{ and}$$

$$P_0P_{01} = P_0P_{0i} \cdot \cos \theta_i + P_{01}P_{0i} \cdot \cos \frac{\pi - \theta_i}{2} \rightarrow$$

$$P_0P_{01} = P_0P_{0i} \cdot \cos \theta_i + P_{01}P_{0i} \cdot \sin \frac{\theta_i}{2} \rightarrow$$

$$P_{01}P_{0i} = \frac{P_0P_{01} \cdot (1 - \cos \theta_i)}{\sin \frac{\theta_i}{2}} \rightarrow$$

$$P_{01}P_{0i} = \frac{r \cdot (1 - \cos \theta_i)}{\sin \frac{\theta_i}{2}} \quad (8)$$



**Figure 6** The elements of the cone corresponding to a current point of the base.

Because we considered the cone determined by a pyramid with a certain number (preferably as large as possible) of lateral faces, these triangular faces have the two sides starting from the tip, equal to the length of the slant heights of the cone corresponding to the points on the base. The side of the triangle on the base of the circle is equal to the chord that corresponds to the arc between the support points of these slant heights, on the circle of the base (Figure 6).

Consider the side face  $\Delta V_F P_{01} P_{0i}$ , denote the angle  $\angle P_{01} V_F P_{0i}$  by  $\alpha_i$  and applying the cosine theorem we have:

$$P_{01}P_{0i}^2 = V_F P_{01}^2 + V_F P_{0i}^2 - 2 \cdot V_F P_{01} \cdot V_F P_{0i} \cdot \cos \alpha_i \rightarrow$$

$$\cos \alpha_i = \frac{V_F P_{01}^2 + V_F P_{0i}^2 - P_{01}P_{0i}^2}{2 \cdot V_F P_{01} \cdot V_F P_{0i}} \rightarrow$$

$$\alpha_i = \arccos \left( \frac{V_F P_{01}^2 + V_F P_{0i}^2 - P_{01}P_{0i}^2}{2 \cdot V_F P_{01} \cdot V_F P_{0i}} \right) \quad (9)$$

To plot the unfolding, from the equations above, both the length of each slant height and the angle made by it with the previous slant height can be determined.

After establishing the insertion point of the unfolding, it will be drawn starting from the slant height indicated in the input data required by the AutoLisp function [2].

#### 4. LISTING OF THE FUNCTION

```
(defun C:DCO ()
  (setvar "CMDECHO" 0)
  (command "-UNITS" "2" "4" "1" "3" "" "")
  (command "UCS" "W")
  (setvar "OSMODE" 0)
  (command "OSNAP" "OFF")
  (command "LAYER" "S" "0" "")
  (command "LTSCALE" "0.5")
  (setq qwe 1)
  (while (= qwe 1)
    (setq TEST (getstring "\nEnter the type of surface
    shape to unfolding [Cone] <Frustum>:"))
    (setq TEST (ascii (strcase test)))
    (print test)
    (if (or (eq test 67) (eq test 70) (eq test 0))
      (setq qwe 0)
    )
  )
  (setq
    l1 nil
    entit1 nil
  )
  (while (eq l1 nil)
    (setq l1 (entsel "\n Select the Axis of cone:"))
    (if (eq (boundp 'l1) T)
      (setq entit1 (cdr (assoc '0 (entget (car l1)))))
    )
    (if (/= entit1 "LINE")
      (setq l1 nil)
    )
  )
  (setq
    P1 (cdr (assoc '10 (entget (car l1))))
    P2 (cdr (assoc '11 (entget (car l1))))
  )
  (setq
    l2 nil
    entit2 nil
  )
  (while (eq l2 nil)
    (while (eq l2 nil)
      (setq l2 (entsel "\n Select the base of the cone:"))
      (if (eq (boundp 'l2) T)
        (setq entit2 (cdr (assoc '0 (entget (car l2)))))
      )
      (if (/= entit2 "LINE")
        (setq l2 nil)
      )
    )
  )
)
```

```

(setq
  P3 (cdr (assoc '10 (entget (car l2))))
  P4 (cdr (assoc '11 (entget (car l2))))
)
(if (/= (cadr P3) (cadr P4))
  (progn
    (prompt "\n THE BASE MUST BE
HORIZONTAL!")
    (setq l2 nil)
  )
)
)
)
(setq
  l3 nil
  entit3 nil
)
)
(while (eq l3 nil)
  (setq l3 (entsel "\n Select the Slant height of the
cone:"))
  (if (eq (boundp 'l3) T)
    (setq entit3 (cdr (assoc '0 (entget (car l3)))))
  )
  (if (/= entit3 "LINE")
    (setq l3 nil)
  )
)
)
)
(setq
  P5 (cdr (assoc '10 (entget (car l3))))
  P6 (cdr (assoc '11 (entget (car l3))))
)
)
(setq
  P0 (inters P1 P2 P3 P4 nil); center of the big base
  X0 (car P0)
  Y0 (cadr P0)
  Z0 (caddr P0)
  P01 (inters P3 P4 P5 P6 nil); intersection between
big base and selected slant height
  X01 (car P01)
  Y01 (cadr P01)
  UBZ (angle P01 P0)
  P02 (polar P0 UBZ (distance P0 P01)); intersectia cu
baza mare a celeilalte generatoare
  VF (inters P1 P2 P5 P6 nil); varful conului
  XF (car VF)
  YF (cadr VF)
  ZF (caddr VF)
)
)
(setq
  USTART (angle P0 P01)
  r (distance P0 P01)
  lcercbazacon (* 2 PI r)
  G (distance VF P01); generatoarea selectata
)
)
(if (or (= TEST 0) (= TEST 70))
  (selectiebazamica)
)
)
)
(setq p (getint "\n Precision of unfolded [100]:"))
(if (eq p nil)
  (setq p 100)
)
)
)
(setq PDSFCO (getpoint "\n Insertion point for
unfolded:"))

```

```

P1A (polar PDSFCO 0 G)
Pdi-1 P1A
)
)
(if (or (eq TEST 70) (eq TEST 0))
  (setq Pmi-1 (polar PDSFCO 0 (distance VF P11)))
)
)
(setq
  tetai (/ (* 2 PI) p)
  i 0
  alfaicum 0
  Gi-1 G
  CC (/ (* r (- 1 (cos tetai))) (sin (/ tetai 2))); coarda
curenta ce alcatuieste curba bazei
)
)
(if (eq TEST 67)
  (COMMAND "LINE" PDSFCO P1A "")
)
)
(if (or (eq TEST 0) (eq TEST 70))
  (progn
    (COMMAND "LINE" Pmi-1 P1A "")
    (Command "LAYER" "M" "AXISFRUSTUM"
"C" "RED" "" "LT" "DASHDOT" "" "")
    (COMMAND "LINE" PDSFCO Pmi-1 "")
    (Command "LAYER" "S" "0" "")
  )
)
)
)
(while (<= tetai (* 2 PI))
  (if (= USTART 0)
    (progn
      (setq
        X0i (+ X0 (* r (cos tetai)))
        Z0i (+ Z0 (* r (sin tetai)))
        Y0i Y0
        P0i (list X0i Y0i Z0i)
        Gi (distance VF P0i)
      )
      (unghialfai)
      (setq alfaicum (+ alfaicum alai))
      (DrawBaseCurve)
      (if (or (= TEST 0) (= TEST 70))
        (drawsmallbase)
      )
      (setq
        tetai (+ tetai (/ (* 2 PI) p))
        i (+ i 1)
        Gi-1 Gi
        Pdi-1 Pdi
        Pmi-1 Pmi
      )
    )
  )
)
)
)
(if (= USTART PI)
  (progn
    (setq
      X0i (- X0 (* r (cos tetai)))
      Z0i (- Z0 (* r (sin tetai)))
      Y0i Y0
      P0i (list X0i Y0i Z0i)
      Gi (distance VF P0i)
    )
    (unghialfai)
    (setq alfaicum (+ alfaicum alai))
  )
)
)

```

```

(DrawBaseCurve)
(if (or (= TEST 0) (= TEST 70))
  (drawsmallbase)
)
(setq
  tetai (+ tetai (/ (* 2 PI) p))
  i (+ i 1)
  Gi-1 Gi
  Pdi-1 Pdi
  Pmi-1 Pmi
)
)
)
)
(if (= TEST 67)
  (Command "Line" Pdi PDSFCO "")
)
(if (or (= TEST 0) (= TEST 70))
  (progn
    (Command "Line" Pdi Pmi "")
    (Command "LAYER" "S" "AXISFRUSTUM" "")
    (Command "Line" PDSFCO Pmi "")
    (Command "LAYER" "S" "0" "")
  )
)
)
)
(defun unghialfai ()
  (setq
    cosalfai (/ (- (+ (* Gi-1 Gi-1) (* Gi Gi)) (* CC
      CC)) (* 2 Gi-1 Gi))
    sinalfai (sqrt (- 1 (* cosalfai cosalfai)))
    alfai (atan sinalfai cosalfai)
  )
)
)
(defun DrawBaseCurve ()
  (setq
    Pdi (polar PDSFCO alfaicum Gi)
  )
  (Command "Line" Pdi-1 Pdi "")
)
)
(defun drawsmallbase ()
  (setq
    A1 (- Y12 Y11)
    B1 (- X11 X12)
    D1 (- (* X12 Y11) (* X11 Y12))
    B2 (- Z0i ZF)
    C2 (- YF Y0i)
    D2 (- (* ZF Y0i) (* YF Z0i))
    A3 (- YF Y0i)
    B3 (- X0i XF)
    D3 (- (* XF Y0i) (* YF X0i))
    X1i (/ (- (* B3 D1) (* B1 D3)) (- (* A3 B1) (* A1
      B3)))
    Y1i (/ (- (* A1 D3) (* A3 D1)) (- (* A3 B1) (* A1
      B3)))
    Z1i (- (* (/ B2 C2) (/ (- (* D1 A3) (* D3 A1)) (- (*
      A3 B1) (* A1 B3)))) (/ D2 C2))
    P1i (list X1i Y1i Z1i)
    Gmc (distance VF P1i)
    Pmi (polar PDSFCO alfaicum Gmc)
  )
  (Command "Line" Pmi-1 Pmi "")
)

```

```

)
(defun selectiebazamica ()
  (setq
    l4 nil
    entit4 nil
  )
  (while (eq l4 nil)
    (while (eq l4 nil)
      (setq l4 (entsel "\n Select the section plan or
        the small base of the Frustum:"))
      (if (eq (boundp 'l4) T)
        (setq entit4 (cdr (assoc '0 (entget (car
          l4)))))
      )
      (if (/= entit4 "LINE")
        (setq l4 nil)
      )
    )
  )
  (setq
    P7 (cdr (assoc '10 (entget (car l4))))
    P8 (cdr (assoc '11 (entget (car l4))))
    P11 (inters VF P01 P7 P8 nil); intersection between
    small base and selected slant height
    X11 (car P11)
    Y11 (cadr P11)
    P12 (inters VF P02 P7 P8 nil); intersection between
    small base and the other selected slant height
    X12 (car P12)
    Y12 (cadr P12)
    P11VF (distance P11 VF)
  )
)
)

```

## 5. CONCLUSIONS

By using this function, it is possible to unfold the oblique cone or the oblique cone frustum, directly from the overall drawing without additional calculations and separate drawings by hand. Time and precision are gained and the possibility of introducing errors is eliminated. The contour obtained can be completed with the necessary dimensions and sent as a drawing to an operator or it can be used as is by a numerically controlled equipment that is part of a cutting machine.

## REFERENCES

- [1] Chiriță, S. (1989). *Probleme de matematici superioare*, Editura Didactică și Pedagogică, ISBN 973-30-0406-9, București.
- [2] Stăncescu, C. (1996). *AutoLISP Manual de programare*, Editura Fast 2000, București.
- [3] Simion, I. (2009). *AutoCAD 2010 pentru ingineri*, Editura Teora, ISBN 978-973-20-1218-5, București.

## Author:

**Teaching Assistant Iulius STROE**, Ph.D. student, University POLITEHNICA of Bucharest, Department of Engineering Graphics and Industrial Design, E-mail: iulius.stroe@gmail.com