The new stuffs not listed in the previous versions of this manual or updated parts are marked with ♣.

This manual describes the new volume-shapes that are not listed in the Epics manual. More than twenty new shpapes are defined and simple "config" files are given in *EPICS*/prog/NewVol/Config. The basic subprograms to treat the new volume-shapes are stored in the *EPICS*/prog/NewVol directory. Those for drawing the volumes are in the *EPICS*/Util/DrawNewVol directory.

♣It is assumed that environmental variables COSMOSTOP and EPICSTOP are both correctly assigned.

1 How to use

The procedure to use some of the new volume-shapes is the same as described in the Epics manual and repeated here. Suppose you are using "sccyl" (sliced cut cylinder) and "sqpipe" (square pipe).

- You should define the new volumes in your configuration file which would include the following lines somewhere:

```
#news   new-1  sccyl
...
#news   new-4  sqpipe
...
1 ...
2   sccyl  Fe   0  0   / 0  0  0  3 5    7.071067812E-01 ...
..
6   sqpipe Al  0 0 /  ...
```

- In *EPICS*/Util, you have to issue,

  ./usenewvol *config*

  where *config* is the path to your config file.

  ♣You can use the 'usenewvol' command in any directory where there is a target 'config' file. The 'configMenu' command will show the list of similar commands which can be invoked in a directory where there is a config file. For details,
  `http://cosmos.n.kanagawa-u.ac.jp/EPICSHome/EPICSupdate2013.pdf`

- To draw the new configuration, in *EPICS*/Util

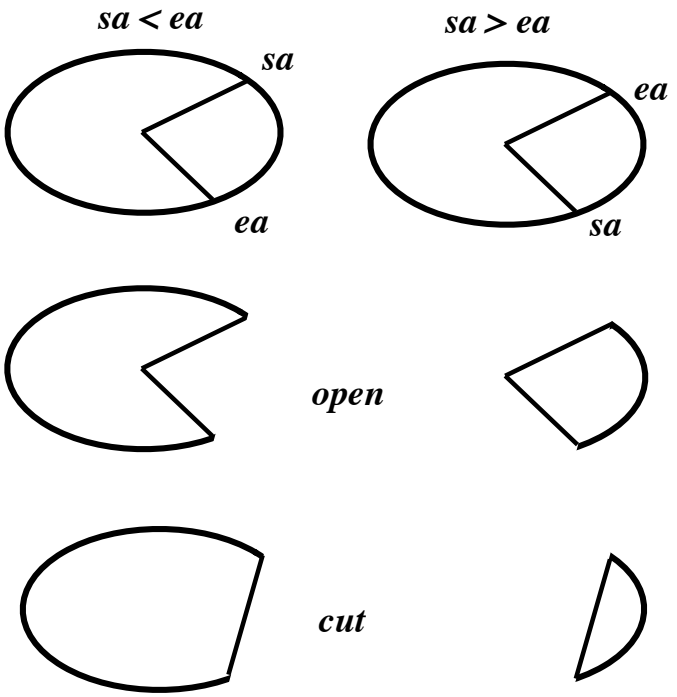
  ♣./drawconfig.sh

- ♣The 'dispconfigbygeomv' and 'disptracebygeomv' commands can be used in a config-existing directory without going to Util/Geomview/ (after executing the drawconfig command)

- Note: The ./usenewvol command is needed only when your config file contains new volume-shape(s) which didn't included in the config file used by the previous ./usenewvol (or usenewvol).

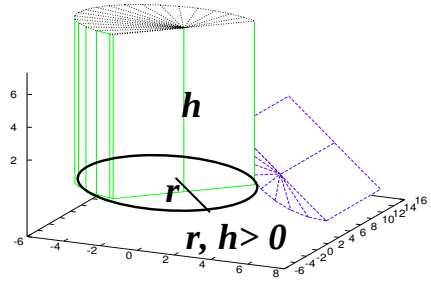
2 Description of each new volume-shape

We shall show volume attributes of a canonical form along with a corresponding figure and (at least) one additional non-canonical form. You will be able to get the same display by specifing the "config file" in *EPICS*/prog/NewVol/Config. The origin of the volume is always needed. For a non-canonical form, the translation will be indicated by a non-canonical origin (normally non-zeros) and for rotation, additional direction cosines of the x and y axes of the cannonical form (6 values) are needed. For the symetrical object around the z-axis, the direction cosine of the z-axis (3 values) may be given. It is important to know that the rotation is first applied to the canonical form and then the translation of the origin is performed.



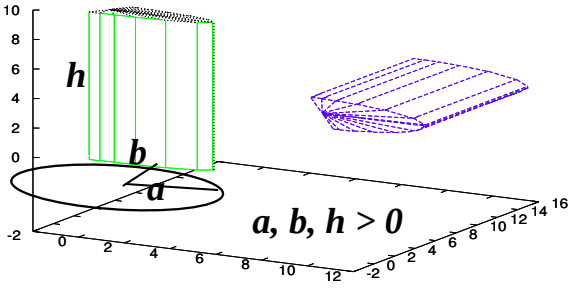
For a cylinder like object, sometimes 'cut' or 'open' is specified. For example, 'ccone' means 'cut elliptic cone' and 'ocyl' is for 'open cylinder'. For such an object, starting angle (sa) and ending angle (ea) must be specified. The cases of sa $<$ ea and ea $<$ sa lead to quite different results as shown left. The meaning of 'open' and 'cut' is also shown left.

ccyl: Cut Cylinder



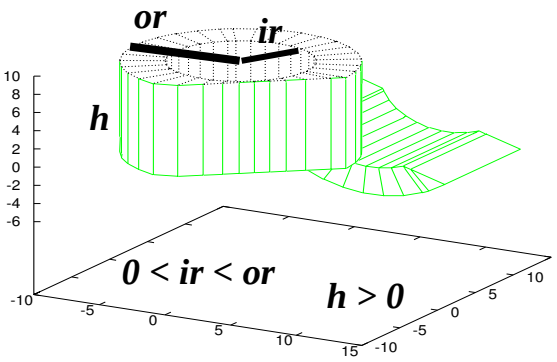
orig r h sa ea [dir]

cecyl: Cut Elliptic Cylinder



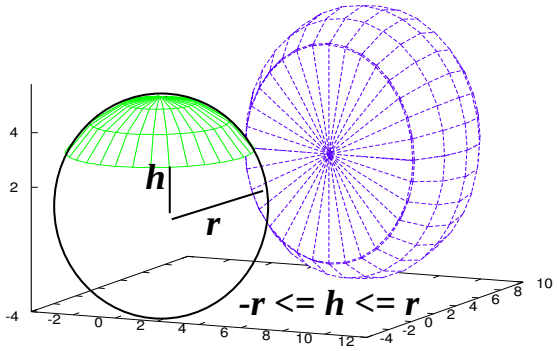
orig a b h sa ea [dir]

cpipe: Cut Pipe



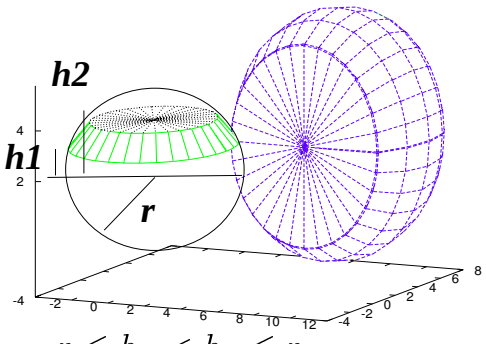
orig ir or h sa ea [dir]

csph: Cut Sphere



orig r h [dir] ***[dir] is for z***

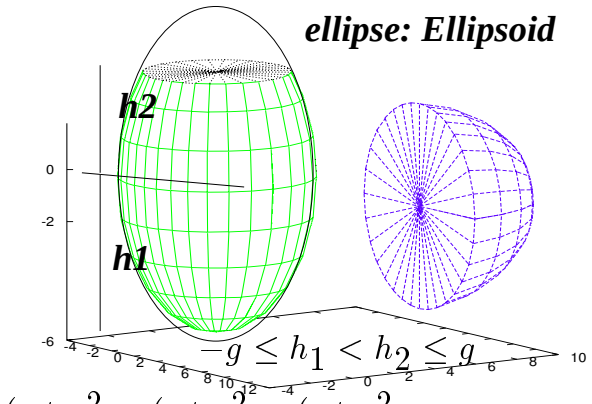
dcsph; Double Cut Sphere



$-r \le h_1 < h_2 \le r$

org r h1 h2 [dir]
[dir] is for z axis

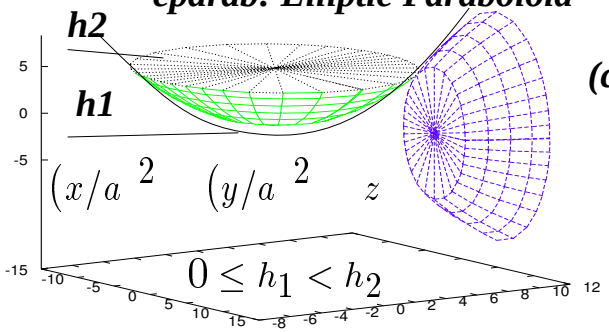
ellipse: Ellipsoid



$(x/a^2 \quad (y/a^2 \quad (z/g^2$

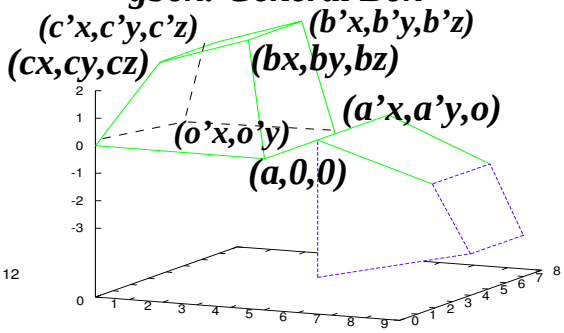
orig a b g h1 h2 [dir]

eparab: Elliptic Paraboloid



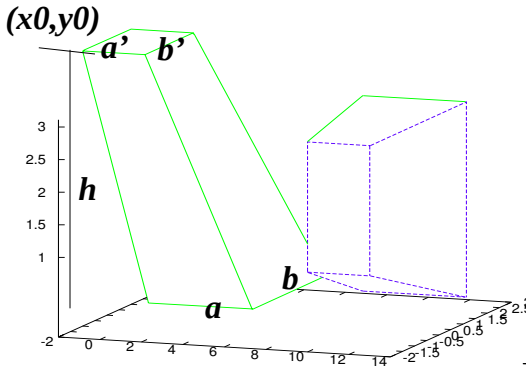
orig a b h1 h2 [dir]

gbox: General Box



orig a bx by bz cx cy cz a'x a'y b'x b'y b'z c'x c'y c'z o'x o'y [dir]

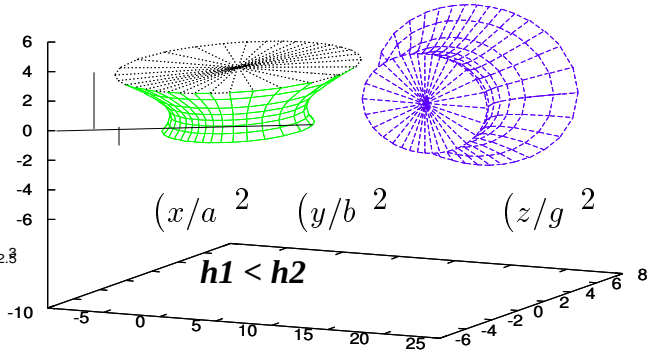
horse: trapezoid



h> 0 a, b, a', b' >=0 but not a=b=a'=b'=0

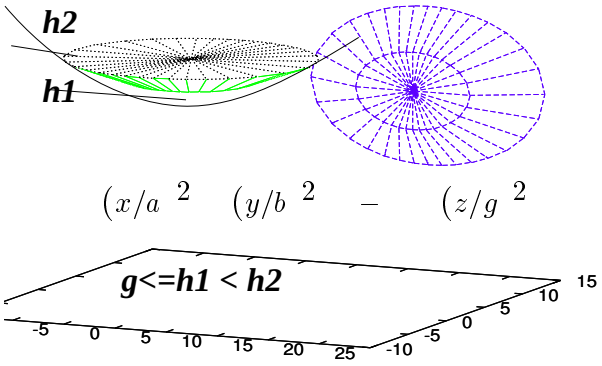
oirg a b h x0 y0 a' b' [dir]

hypb: Hyperboloid of 1st kind



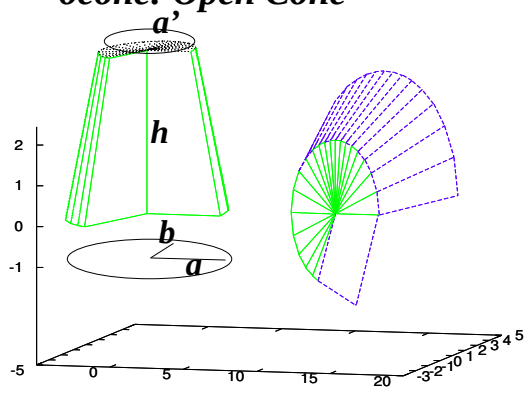
orig a b g h1 h2 [dir]

hypb2: Hyperboloid of 2nd kind



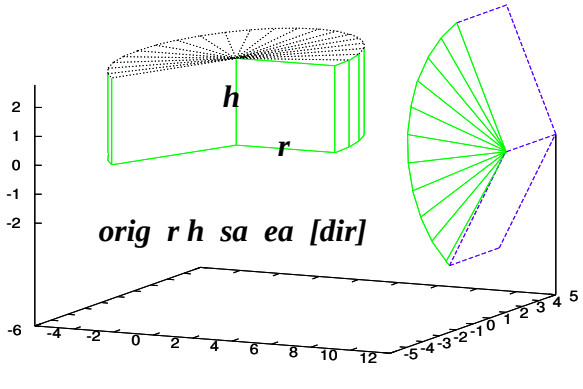
orgi a b g h1 h2 [dir]

ocone: Open Cone

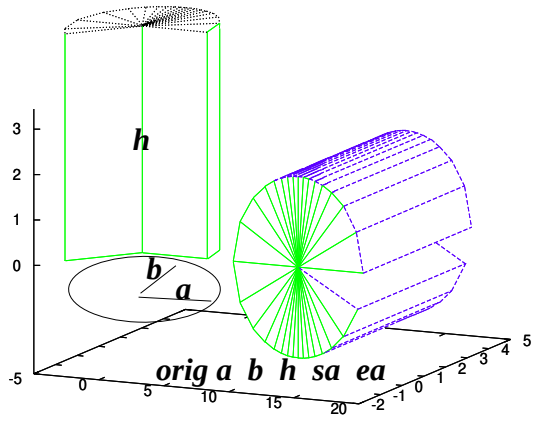


orig a b h a' sa ea

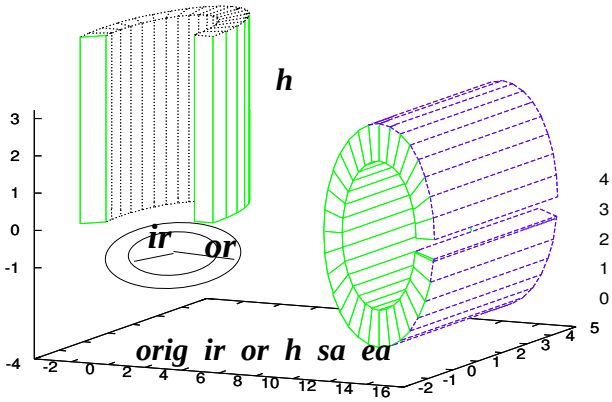
ocyl: Open Cylinder



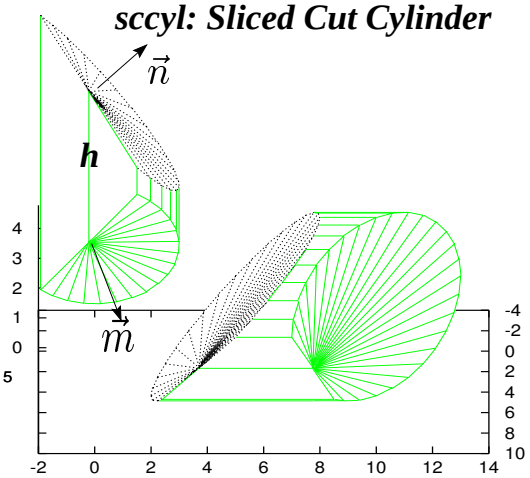
oecyl: Open Elliptic Cylinder



opipe: Open Pipe

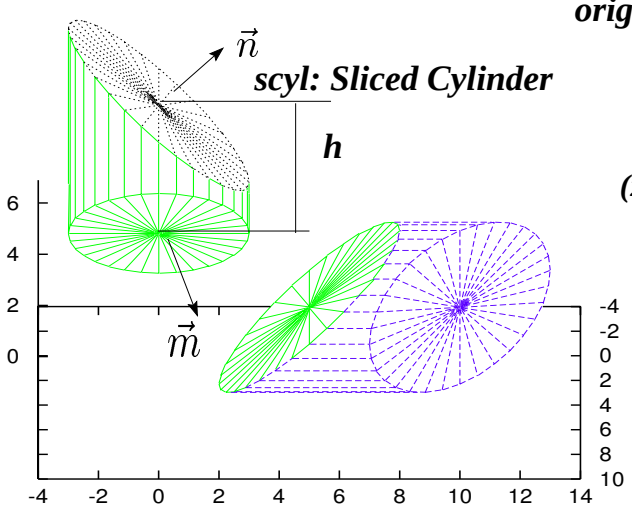


sccyl: Sliced Cut Cylinder



orig r h mx my mz nx ny nz sa ea [dir]

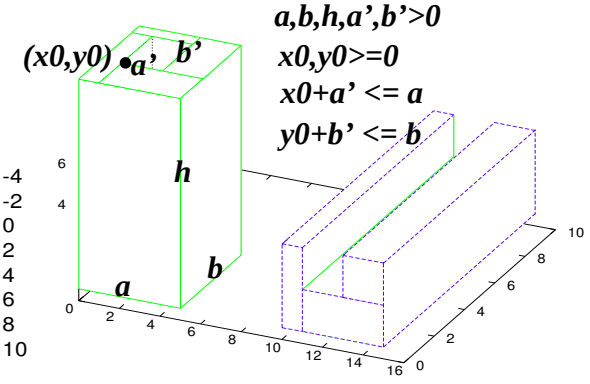
scyl: Sliced Cylinder



orig r h mx my mz nx ny nz [dir]

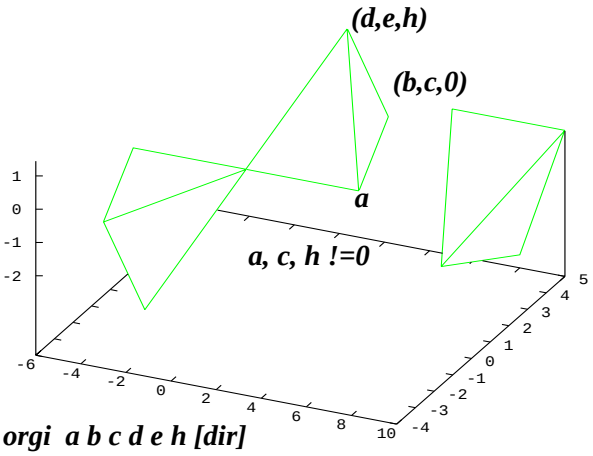
sqpipe: Square Pipe

a,b,h,a',b'>0
x0,y0>=0
x0+a' <= a
y0+b' <= b



orig a b h x0 y0 a' b' [dir]

tripyra: Triangular Pyramid

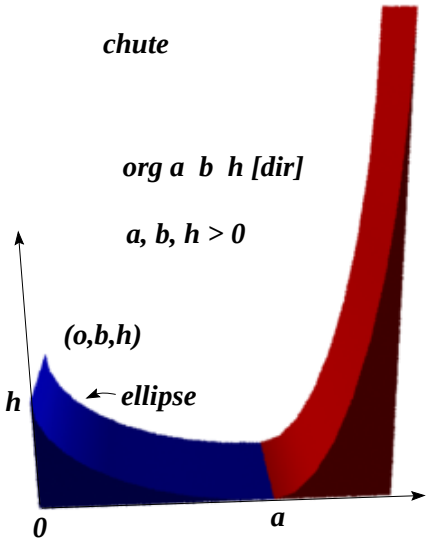


orgi a b c d e h [dir]

chute

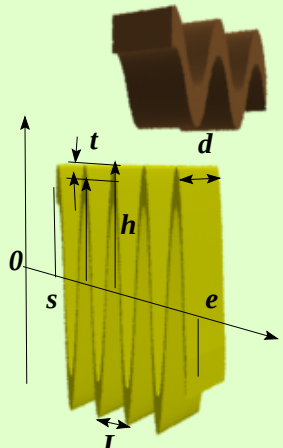
org a b h [dir]

a, b, h > 0



wave

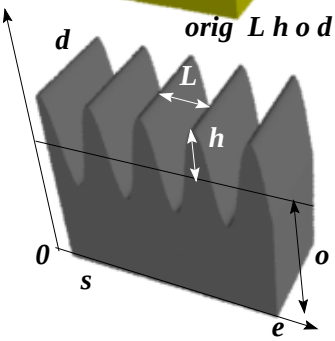
orig L h t d s e [dir]



L, h, t, d, e> 0 s >=0

swave

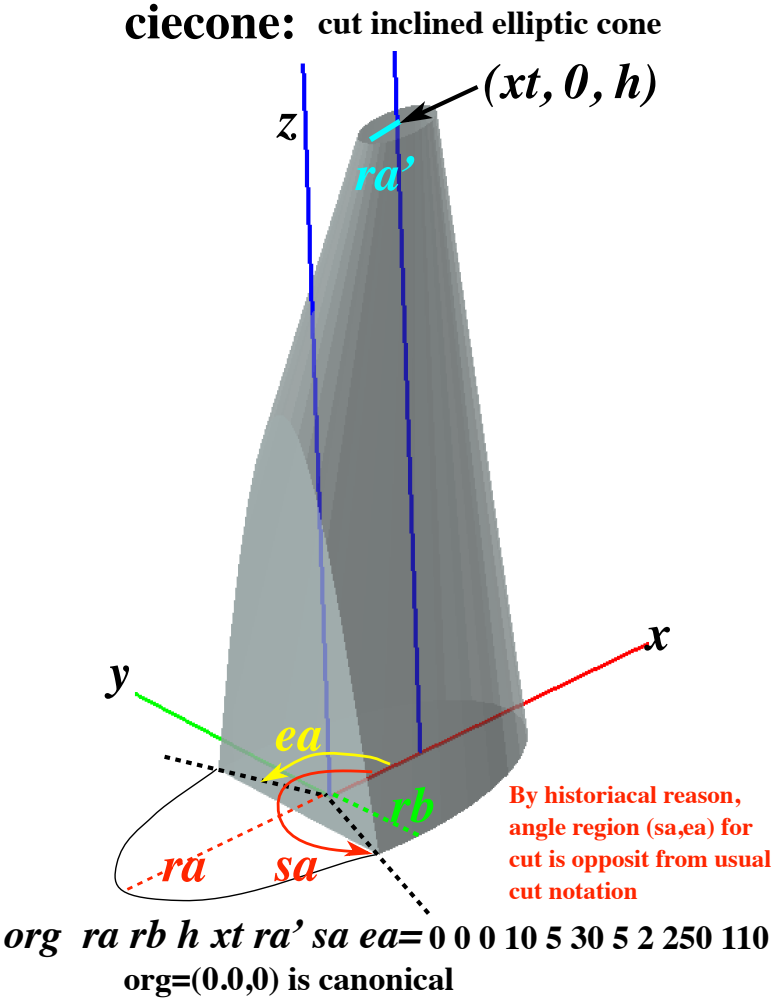
orig L h o d s e [dir]



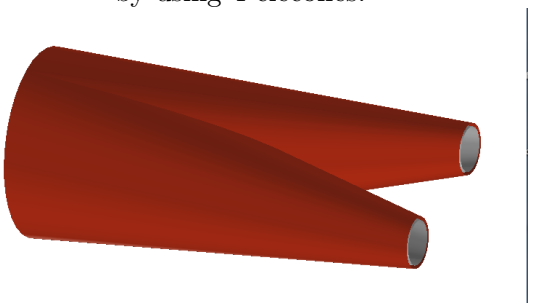
L, h, d, e > 0, s>=0, o>=h

3 ♣Recent ones

3.1 ciecone

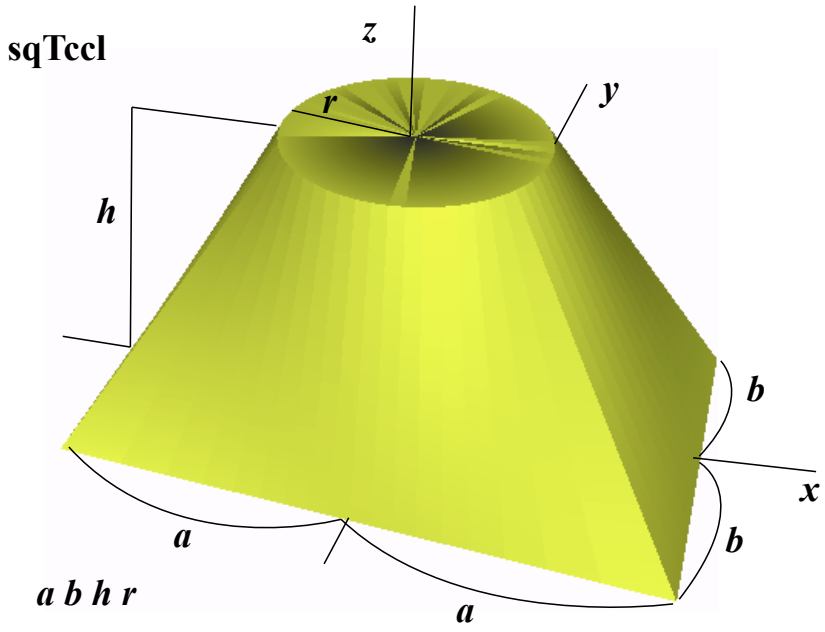


The recombination chamber (so called "pants") for the beam pipe at LHC may be formed by using 4 ciecones.



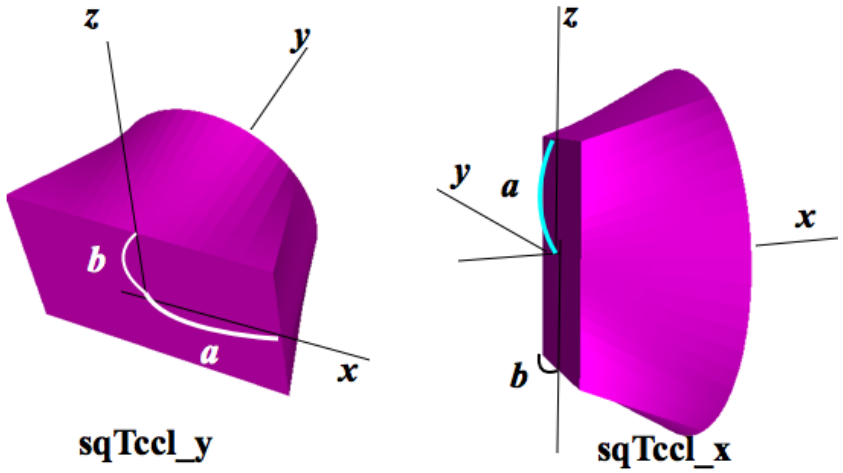
3.2 sqTccl

sqTccl is a component to convert square shape into circle one (or vice versa).



sqTccl_x sqTccl_y are available

Figure 1: Canonical sqTccl. sqTccl_x and sqTccl_y can also be used (Below)

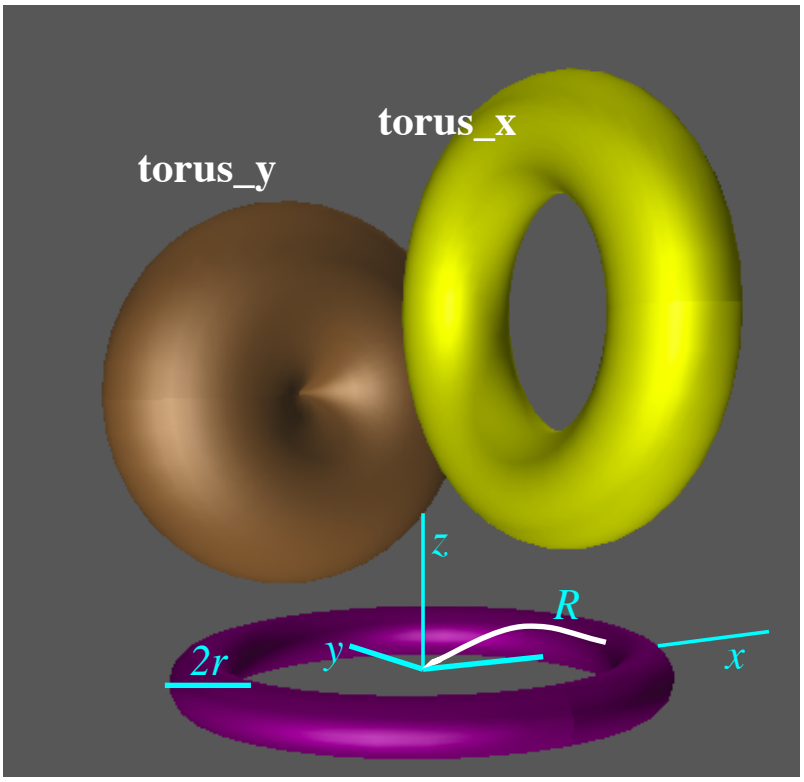


The surface of the side wall part is approximated by a number of triangles, and hence not completely smooth. All attributes must be > 0. (The radius r can be larger than a or b). An example is `#news new-1 sqTccl`

```
1 sqTccl PWO 0 0 0 / 0 0 0 a b h r
```

3.3 torus

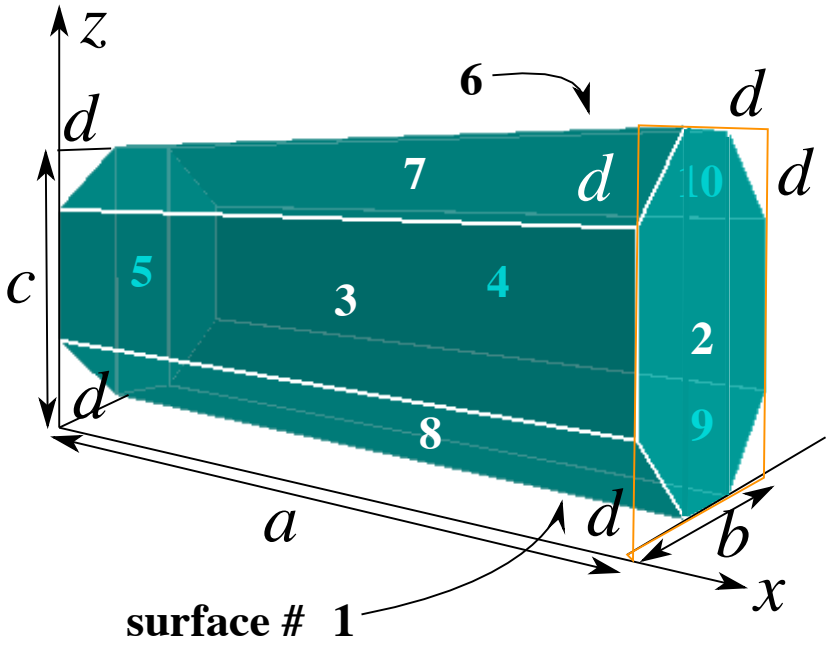
torus



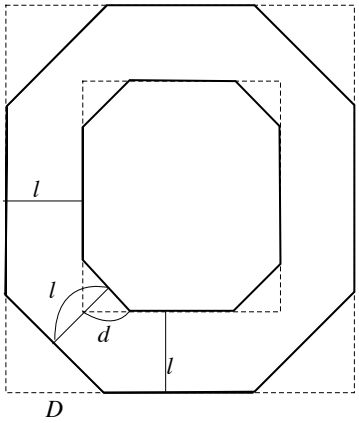
torus_z =torus
canonical one: center is at 0 0 0
config: orig R r

3.4 octagon

Octagon. Canonical config: 0 0 0 a b c d. `octagon_x=octagon` `octagon_y` and `octagon_z` are usable.



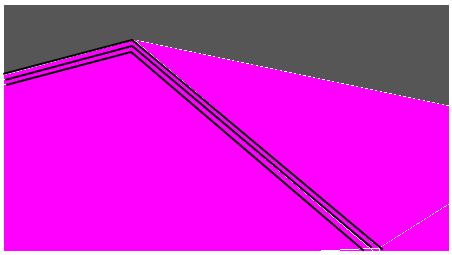
If a clad is to be put, the following may be used.



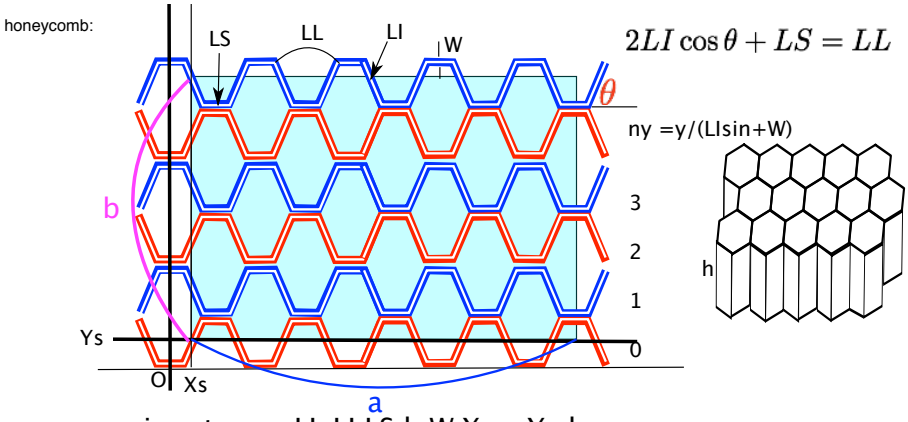
cos = cos(45deg)

(D cos+ l-d*cos)cos=l*

*D = (l/cos+d*cos-l)/cos*



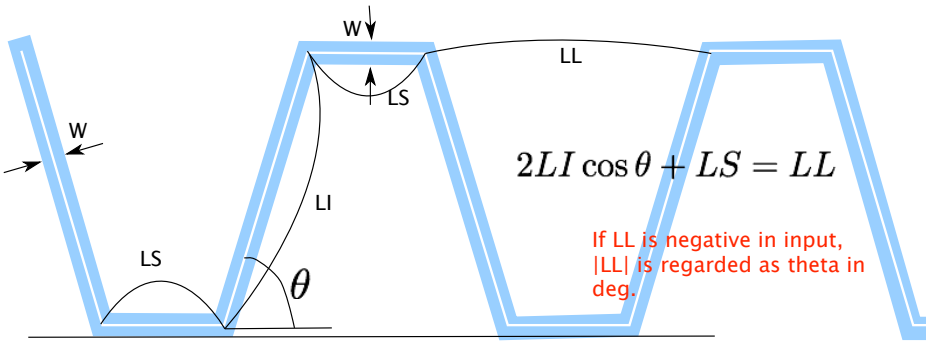
3.5 honeycomb



input : LL LI LS h W Xs a Ys b

if LL is negative: -LL is regarded as theta in deg.
-60 0.5 0.5 ... means regular hexagon type

Effective honyecomb area is the shaded area. Other parts are non existent in the simulation altough drawing may show some of them. In config file, the user must always define a subdetector with a tight world (i.e., box_w sp 0 0 0 / Xs Ys 0 a b h); sp may be air, H2O etc in some case. If Xs=Ys=0, you may simply write box_w sp 0 0 0 /0 0 0. (see next page). ~~Note volume-shape name is~~ honeycm ~~not honeycomb.~~

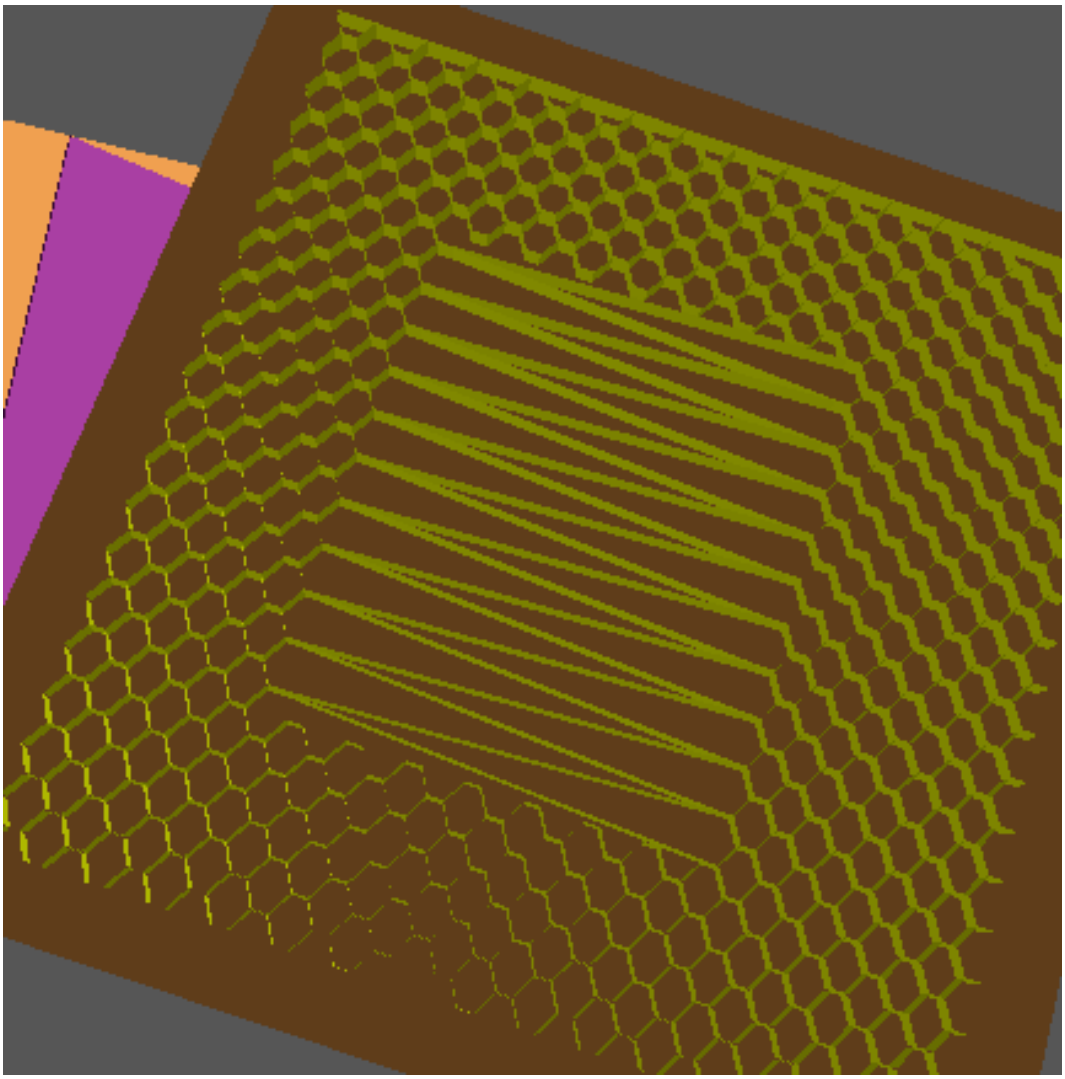


config examle

```
#news  new-1  honeycm
#eq    sq  35
#eq    sq2  40
#subd  honey
1 honeycm Al    3  0  0 / 0   0  0  -60 1 0.6  1  0.05 0 sq  0 sq
2 box_w     sp    0  0  0 /  0 0 0
#end honey
1 box W       0 0 0  /  0 0  0     sq2 sq2 0.5
2 box Carbon  0 0 0  /  0  0  +    =  = 0.1
3 honey  sp   0   0   0 / 2.5  2.5   +
4 box Carbon  0 0 0  /  0  0  +    sq2  sq2 0.1
5 box W       0 0 0  /  0 0  +     =  = 0.5
6 box_w sp      0 0 0  / 0 0 0
```

Honeycom drawing:
Since the unit cell is normally small, and the area covered by the honeycom is large, the number of vertexes becomes order of few 10 thousands. So for such case, default drawiing omits the cetral part of the honecycom (though some abbreviated form may appear). Also, currenly, the wall thickness is not shown. To show all the honeycom elements, you have to give a value > 0.5 to "drawportion" in Util/DrawNewVol/epDraw-honeycm.f.

The wall of a honecomb is drawn neglecting its thickness (only central boundary is drawn). The central part of the square region is abbreviated as shown in the Fig. below. (To change the abbreviation condition, edit Util/Data/honeycomb.dat; currently thickness of the wall is always 0).



3.6 fpolygon

There are 3 canonical types: fpolygon (=fpolygon_xy), fpolygon_yz and fploygon_zx. The canonical form of fpolygon_uv (uv=“ ”, xy, yz or zx) is made as follows. One flat polygon is placed on the uv-plane. The verteces of the polygon are described by (u,v) pairs on the uv-plane. The volume is made by putting another identical polygon at a distnace $h(> 0)$ from the uv-plane, and upper and lower polygons vertices are connected to form a volume. In a config file, it is described as follows, e.g,

```
1 fpolygon_uv Pb 0 0 0 / 0 0 0 np  h  u1 v1 u2 v2 ... \
                                   ui vi ....   unp vnp
```

where ”np“ is the number of verteces, “h” the height and “(ui vi)” a vertex position on the uv plane. \ may be used to indicate that the line is continued to the next line.

```
-------------------------------------------------------------
#news new-1  fpolygon
1  fpolygon Al   0 0 0 / 0  0  0 10 1 \
          1 1   2 0.5  2 1.5   3 1   3 2.5 \
          2 2   1.5 3   1 2    1.5 1.5  0 1.2
-------------------------------------------------------------
```

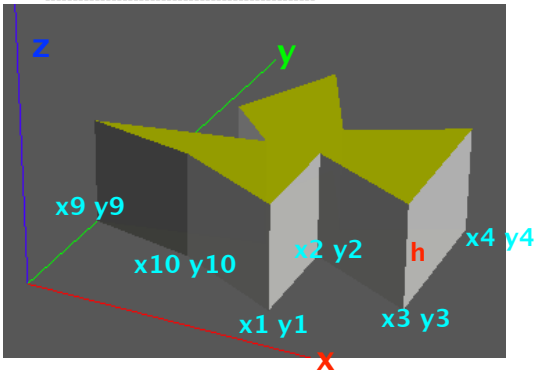


Figure 2: fpolygon =fpolygon_xy

The current maximum value of “np” is 20. It could be expanded by redefining Zep-Maxdef.h. The continuation by \ may be used whereever needed, provided it must not be placed at the top of the line. If it is included in a comment line starting with #, it is a part of the comment.

If a fpolygon is to be rotated, we specify the direction cosine of the new x-axis and new y-axis in the present coordinate (irrespectively of uv). For example,

```
1 fpolygon_zx  Pb 0 0 0 / 0 5 10 np  h  u1 v1 u2 v2 ... \
                        ui vi ....   unp vnp  \
                        wxx wxy wxz wyx wyy wyz
```

where wxp (p=x,y,z) are the direction cosine of the rotated x-axis to the old p-axis. wyp the same for the rotated y-axis. After rotation, the coordinate values are shifted by the values in the origin part. (0 5 10 in the above example).

If it containes other components, the component numbers may be given after “/” as usual.

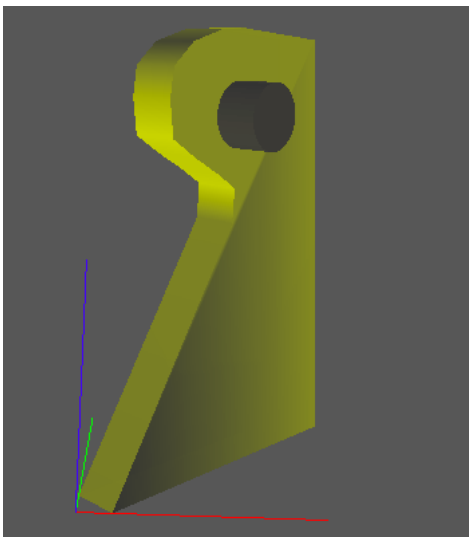


Figure 3: fpolygon_yz contains a cylinder and rotated

```
------------------------------------------------------------------
#news new-1  fpolygon
#subd  demo
1  fpolygon_yz Al  0 0 0 / 0 0 0  13  2 \
  0 0  14 0  11 19  6 19  4.6 18.8  \
  3 18.2  2 17.2  1.9 16  2.2 14.5 3.5 13.8 \
  5 13.1  6.1 12.5  6.1 11.0  / 2
2 cyl_x hollow  0 0 0 / 0 8.5 15.5  1.5 2
3 box_w sp 0 0 0 / 0 0 0
#end  demo
1  demo  sp 0 0 0 / 0. 1.4142  0  0.707106781 -0.707106781 0 \
                               0.707106782  0.707106781 0
2  box_w  sp  0 0 0 / 0 0 0
------------------------------------------------
```

3.6.1 Gemview problem

In displaying the fpolygon by Geomview, a bug (?) is observed. Although the manual says it can draw filled concave polyogns, it fails to draw them correctly as seen in the fig.

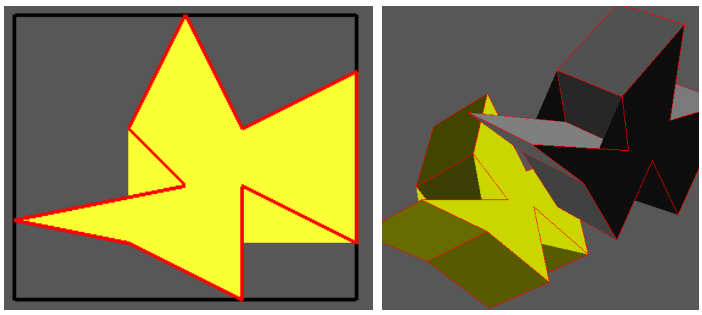


Figure 4: Geomview fails to fill concave polygon surface: red line shows correct boundary

3.6.2 Workaround

In the case of Geomview 1.9.4, if we set α (transparency) not exactly equal to 1, the odd surface will disappear. But this is not perfect as shown below. Geomview ∼1.8.1 has poorer

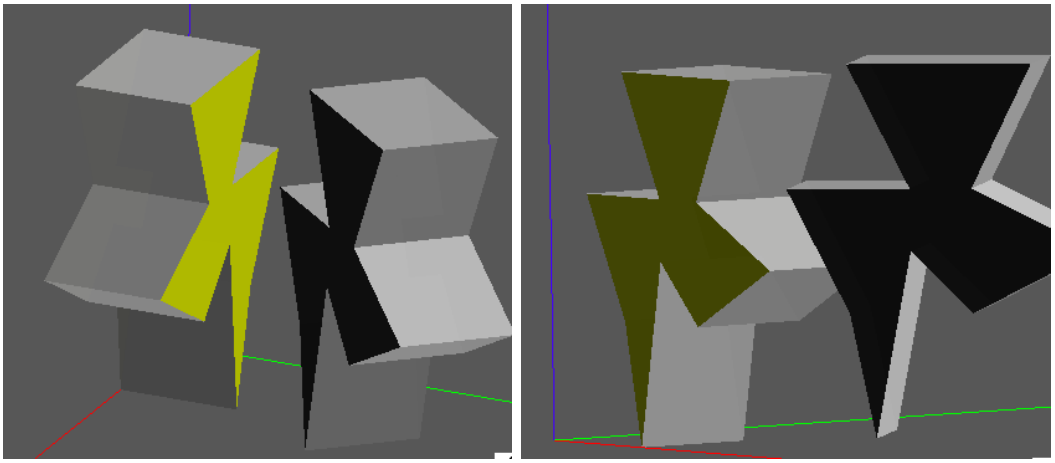


Figure 5: Setting non-1 transparency removes odd surface (left) but sometimes loses perspective (right fig: the right component is actually at far side but looks nearer)

ability.

Anoher workaround (?) is not to draw surface at all. This is acheived by putiing -s when you invoke Geomview. Say,

```
 ./dispconfigbygeomv  -s
or
 ./disptracebygeomv  trace_data det -s
```

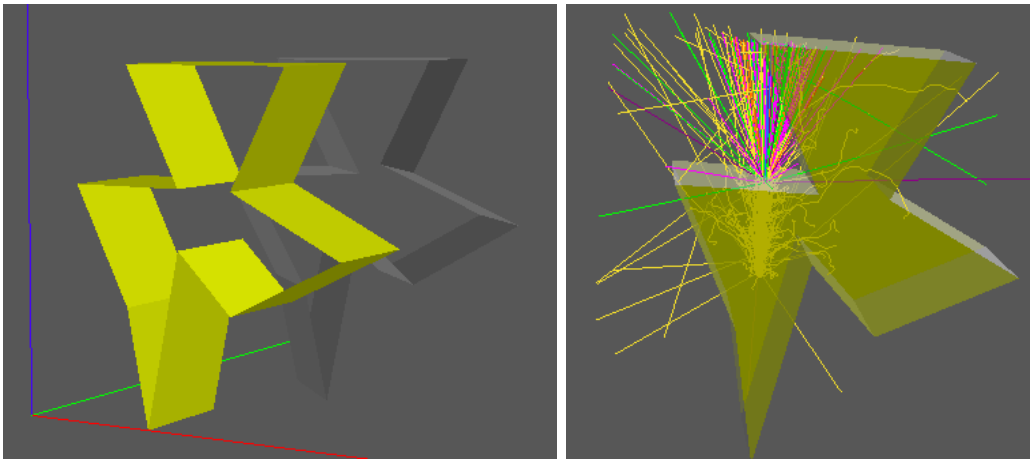


Figure 6: -s option dose not draw surface (left). One example of cascade in a fpolygon(right)