

# ‘pst-solides3d’ : section d’un solide par un plan avec `\codejps`

16 juin 2008

## Résumé

Un solide peut être coupé partagé par un plan et les deux parties obtenues utilisées séparément. Les illustrations de ce document ont été essentiellement obtenues en employant le code `jps`, un deuxième document traite du même sujet, mais par usage des commandes `PStricks`.

## Table des matières

<b>1</b>	<b>Coupe d’un octaèdre par un plan parallèle à l’une des faces</b>	<b>2</b>
1.1	Usage avec <code>PStricks</code>	2
1.2	Usage avec <code>\codejps</code>	2
1.2.1	Voir l’intérieur	3
1.2.2	On considère le solide comme plein	3
1.2.3	Les deux parties du solide découpé	4
<b>2</b>	<b>Coupes d’un cube</b>	<b>5</b>
2.1	Usage avec <code>PStricks</code>	5
2.2	Usage avec <code>\codejps</code>	6
<b>3</b>	<b>Coupes dans un parallélépipède</b>	<b>7</b>
3.1	Exemple 1	7
3.2	Exemple 2	8
<b>4</b>	<b>Coupes d’un prisme octogonal</b>	<b>9</b>
4.1	Troncature des sommets d’un prisme octogonal	9
4.2	Chanfreinage des arêtes des bases d’un prisme octogonal	10
<b>5</b>	<b>Coupes d’un cylindre</b>	<b>11</b>
5.1	Coupe du solide plein	11
5.2	Coupe du solide creux	11
5.3	Coupe en faisant varier la distance du plan	13
5.4	Coupe par un plan parallèle à l’axe	14
<b>6</b>	<b>Coupe d’un cône par un plan parallèle à l’axe</b>	<b>14</b>
<b>7</b>	<b>Autres exemples</b>	<b>16</b>

# 1 Coupe d'un octaèdre par un plan parallèle à l'une des faces

## 1.1 Usage avec PStricks

L'équation du plan s'écrit :

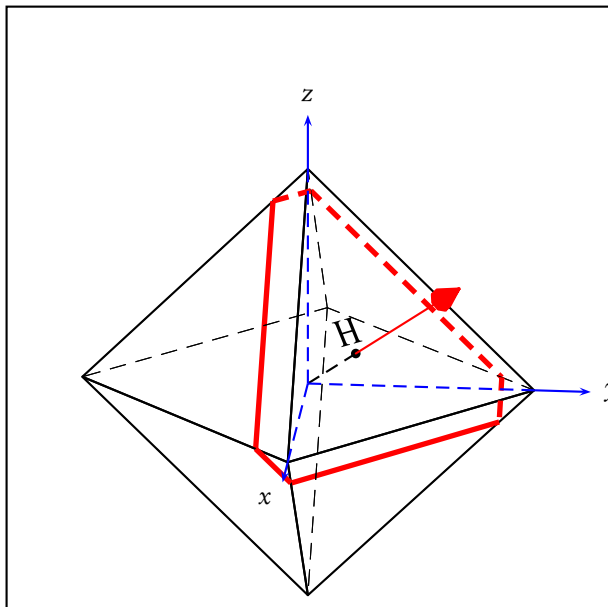
$$ax + by + cz = d$$

Rappelons que  $a, b, c$  sont les coordonnées du vecteur unitaire normal au plan et  $d$  la distance de l'origine  $O$  au plan. L'équation du plan se passe dans un tableau  $[a \ b \ c \ -d]$  dans le paramètre

`[intersectionplan={a b c -d}]`.

Une valeur positive ou nulle de `[intersectiontype=0]` dessine la trace de l'intersection. L'épaisseur en points et la couleur sont indiquées dans les options :

- `[intersectionlinewidth=2]`
- `[intersectioncolor=(rouge)]`.



```
\psset{SphericalCoord=true,viewpoint=100
5 20,Decran=150}
\begin{pspicture}(-4,-3)(4,5)
\pstVerb{/distance 0.4 2 mul def
/n_x 45 cos 35.2644 cos mul def
/n_y 45 sin 35.2644 cos mul def
/n_z 35.2644 sin def
/xH distance n_x mul def
/yH distance n_y mul def
/zH distance n_z mul def
}%
\psframe(-4,-3)(4,5)
\psSolid[object=octahedron,a=2,action=
draw,intersectiontype=0,
intersectionplan={
[ n_x % a
n_y % b
n_z % c
distance neg]],
intersectionlinewidth=2,
intersectioncolor=(rouge)]
\psPoint(xH,yH,zH){H}
\psPoint(0,0,0){O}
\psline[linestyle=dashed](O)(H)
\psdot(H)
\psProjection[object=texte,
fontsize=10,pos=dc,text=H,
normal=n_x n_y n_z -30](xH,yH,zH)
\Normale[linecolor=red](distance
,45,35.2644)
\axesIIID[linecolor=blue](2,2,2)
(2.5,2.5,2.5)
\end{pspicture}
```

## 1.2 Usage avec \codejps

Dans ce cas les paramètres du plan `[a b c -d]` sont passés en argument à la macro `solidplansepare`.

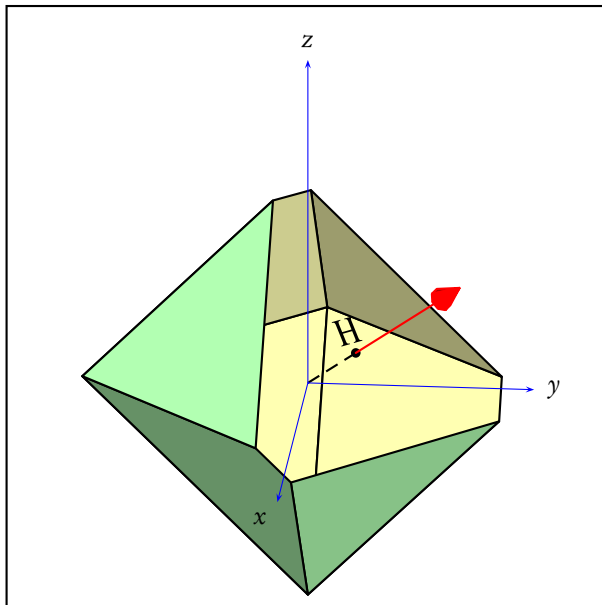
La syntaxe sera la suivante :

```
solide
[a b c -d]
solidplansepare
```

Cette commande dépose sur la pile les deux parties du solide découpé, qu'on peut donc traiter séparément. La partie qui se trouve sur le dessus de la pile est celle qui est physiquement située au dessous de la normale en  $H$ . Dans les deux exemples suivants on ne s'occupe que de cette partie.

### 1.2.1 Voir l'intérieur

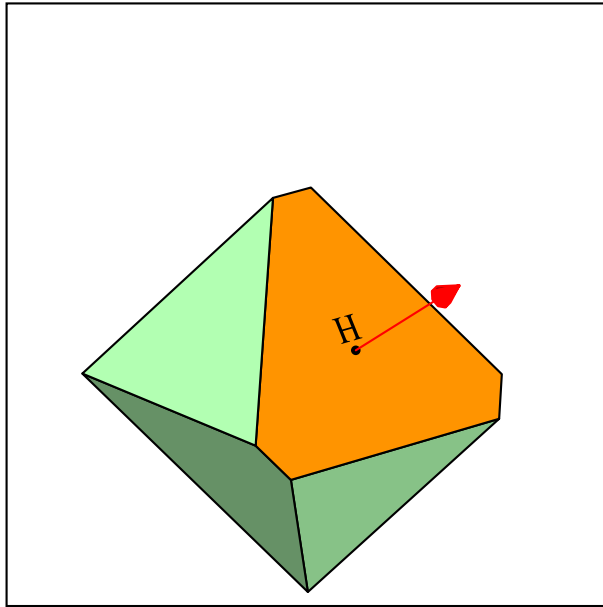
Dans le premier exemple on a supprimé la face découpée pour voir l'intérieur.



```
\psset{SphericalCoor=true,viewpoint=100
5 20,Decran=150,a=2}
\begin{pspicture}(-4,-3)(4,5)
\pstVerb{/distance 0.4 2 mul def
/n_x 45 cos 35.2644 cos mul def
/n_y 45 sin 35.2644 cos mul def
/n_z 35.2644 sin def
/xH distance n_x mul def
/yH distance n_y mul def
/zH distance n_z mul def
}%
\psframe(-4,-3)(4,5)
\psset{lightsrc=93.6 8.2 34.2}
\codejps{
a newoctaedre
% plan : ax+by+cz+d=0
[ n_x % a
n_y % b
n_z % c
distance neg % distance variable
] solidplansepare
dup [0] solidrmfaces
dup videsolid
dup (1 1 0.7 setrgbcolor) (0.7 1 0.7
setrgbcolor) inoutoutputcolors
solidlightOn
drawsolid**}
\psPoint(xH,yH,zH){H}
\psPoint(0,0,0){O}
\psline[linestyle=dashed](O)(H)
\psdot(H)
\psProjection[object=texte,
fontsize=10,pos=dc,text=H,
normal=n_x n_y n_z -30](xH,yH,zH)
\Normale[linecolor=red](distance
,45,35.2644)
\axesIIID[linecolor=blue,linewidth=0.4
pt](0,0,0)(3,2,3)
\end{pspicture}
```

### 1.2.2 On considère le solide comme plein

Dans ce deuxième exemple, l'octaèdre est découpé comme un solide plein.

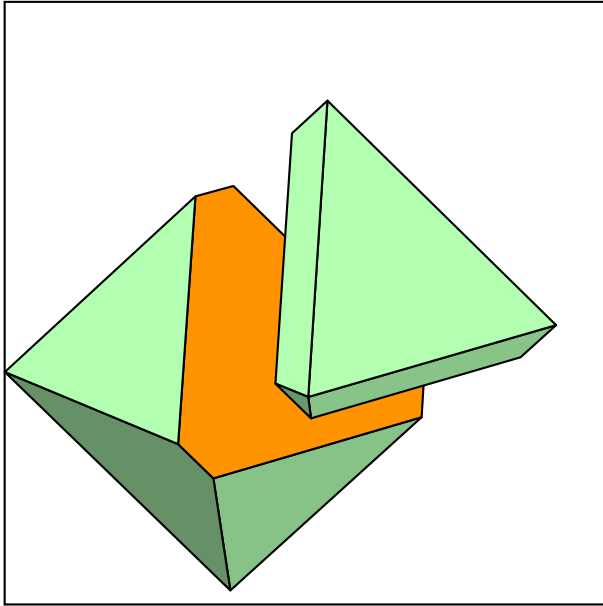


```

1 \psset{SphericalCoor=true,viewpoint=100
2   5 20,Decran=150,a=2}
3 \begin{pspicture}(-4,-3)(4,5)
4 \pstVerb{/distance 0.4 2 mul def
5           /n_x 45 cos 35.2644 cos mul def
6           /n_y 45 sin 35.2644 cos mul def
7           /n_z 35.2644 sin def
8           /xH distance n_x mul def
9           /yH distance n_y mul def
10          /zH distance n_z mul def
11          }%
12 \psframe(-4,-3)(4,5)
13 \psset{lightsrc=93.6 8.2 34.2}
14 \codejps{
15 a newoctaedre
16 % plan : ax+by+cz+d=0
17 [ n_x % a
18   n_y % b
19   n_z % c
20   distance neg % distance variable
21 ] solidplansepare
22 dup (0.7 1 0.7 setrgbcolor)
23   outputcolors
24   dup 0 (YellowOrange) solidputfcolor
25   solidlightOn
26   drawsolid**}
27 \psPoint(xH,yH,zH){H}
28 \psdot(H)
29 \psProjection[object=texte,
30   fontsize=10,pos=dc,text=H,
31   normal=n_x n_y n_z -30](xH,yH,zH)
32 \Normale[linecolor=red](distance
33   ,45,35.2644)
34 \end{pspicture}

```

### 1.2.3 Les deux parties du solide découpé



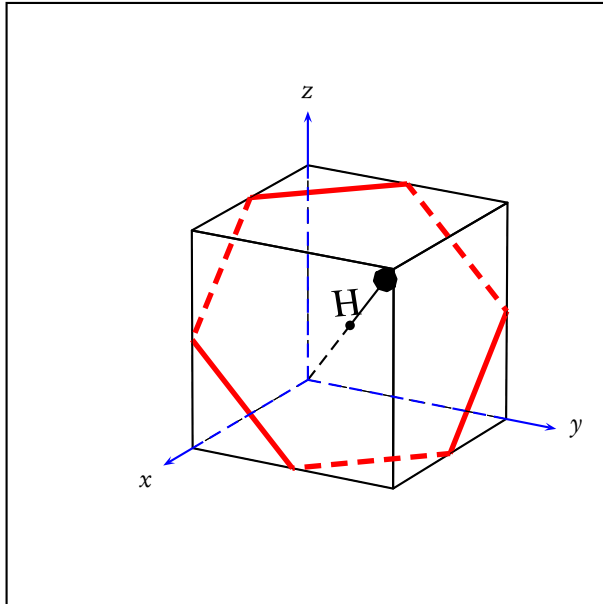
```

1 \psset{SphericalCoor=true,viewpoint=100
  5 20,Decran=150,a=2}
2 \begin{pspicture}(-3,-3)(5,5)
3 \pstVerb{/distance 0.4 2 mul def
4         /n_x 45 cos 35.2644 cos mul def
5         /n_y 45 sin 35.2644 cos mul def
6         /n_z 35.2644 sin def
7         /xH distance n_x mul def
8         /yH distance n_y mul def
9         /zH distance n_z mul def
10        }%
11 \psframe(-3,-3)(5,5)
12 \psset{lightsrc=93.6 8.2 34.2}
13 \codejps{
14   solidlight0n
15   a newoctaetre
16   dup (0.7 1 0.7 setrgbcolor)
17   outputcolors
18   % plan : ax+by+cz+d=0
19   [ n_x % a
20     n_y % b
21     n_z % c
22     distance neg % distance variable
23   ] solidplansepare
24   dup 0 (YellowOrange) solidputfcolor
25   drawsolid**
26   {2 xH mul 2 yH mul 2 zH mul
27     translatepoint3d} solidtransform
28   drawsolid**}
29 \end{pspicture}

```

## 2 Coupes d'un cube

### 2.1 Usage avec PStricks

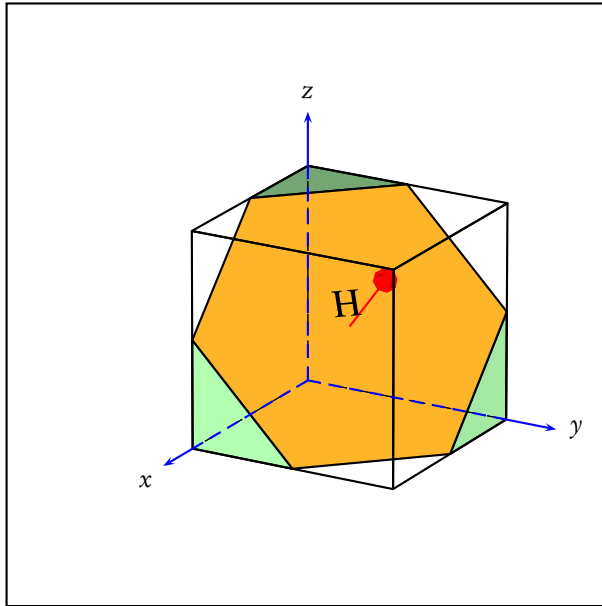


```

1 \psset{SphericalCoor=true,viewpoint=100
2   30 20,Decran=150}
3 \begin{pspicture}(-4,-3)(4,5)
4 \pstVerb{/distance 1.732 def
5   /n_x 0.57735 def
6   /n_y 0.57735 def
7   /n_z 0.57735 def
8   /xH distance n_x mul def
9   /yH distance n_y mul def
10  /zH distance n_z mul def
11  }%
12 \psframe(-4,-3)(4,5)
13 \psProjection[object=teixe,
14   fontsize=10,pos=dc,text=H,
15   normal=n_x n_y n_z -30](xH,yH,zH)
16 \Normale[fillcolor=red](distance
17   ,45,35.2644)
18 \psSolid[object=cube,a=2,action=draw,
19   intersectiontype=0,
20   intersectionplan={
21     [ n_x % a
22     n_y % b
23     n_z % c
24     distance neg]],
25   intersectionlinewidth=2,
26   intersectioncolor=(rouge)](1,1,1)
27 \psPoint(xH,yH,zH){H}
28 \psPoint(0,0,0){O}
29 \psline[linestyle=dashed](O)(H)
30 \psdot(H)
31 \axesIIIID[linecolor=blue](2,2,2)
32   (2.5,2.5,2.5)
33 \end{pspicture}

```

## 2.2 Usage avec \codejps



```

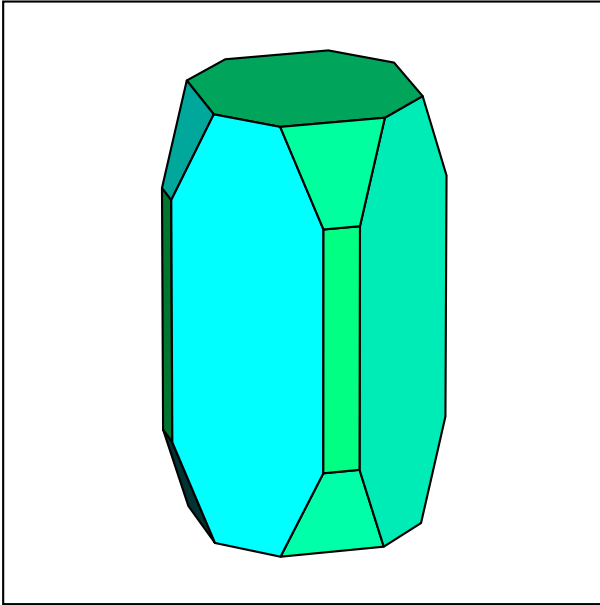
1 \psset{SphericalCoor=true,viewpoint=100
2   30 20,Decran=150,a=2}
3 \begin{pspicture}(-4,-3)(4,5)
4 \psset{lightsrc=81.4 47 34.2}
5 \pstVerb{/distance 1.732 def
6   /n_x 0.57735 def
7   /n_y 0.57735 def
8   /n_z 0.57735 def
9   /xH distance n_x mul def
10  /yH distance n_y mul def
11  /zH distance n_z mul def
12  }%
13 \psframe{(-4,-3)(4,5)
14 \codejps{
15 a newcube
16 {a 2 div a 2 div a 2 div translatepoint
17 3d} solidtransform
18 % plan : ax+by+cz+d=0
19 [ n_x % a
20   n_y % b
21   n_z % c
22   distance neg % distance variable
23 ] solidplansepare
24 dup (0.7 1 0.7 setrgbcolor)
25   outputcolors
26   dup 0 (Dandelion) solidputfcolor
27 % dup solidnumfaces
28 solidlightOn
29 drawsolid**}
30 \Normale[linecolor=red](distance
31   ,45,35.2644)
32 \psSolid[object=cube,a=2,action=draw
33   ](1,1,1)
34 \psProjection[object=texte,
35   fontsize=10,pos=dc,text=H,
36   normal=n_x n_y n_z -30](xH,yH,zH)
37 \axesIIID[linecolor=blue](2,2,2)
38   (2.5,2.5,2.5)
39 \end{pspicture}

```

### 3 Coupes dans un parallélépipède

#### 3.1 Exemple 1

Le solide de départ est un parallélépipède. Les sommets sont tronqués `.333 solidtronque` et les arêtes verticales chanfreinées en faisant tourner un plan de coupe vertical autour de l'axe.



```

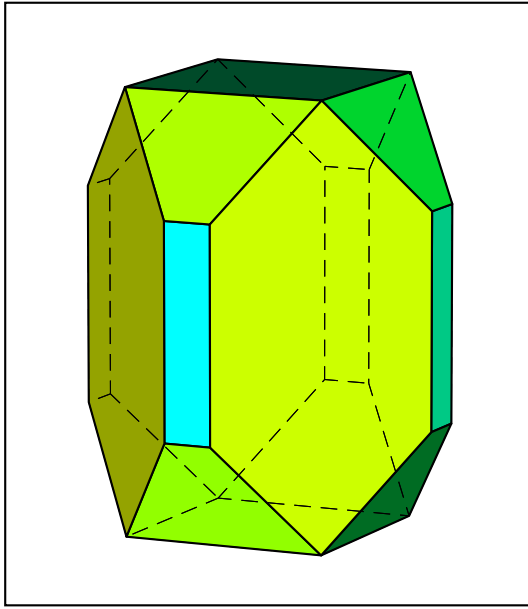
1 \psset{SphericalCoor=true,viewpoint=100
   30 20,Decran=150,a=2}
2 \psset{lightsrc=81.4 47 34.2}
3 \begin{pspicture}(-4,-4)(4,4)
4 \psframe(-4,-4)(4,4)
5 \codejps{
6 2 2 4 newparallelepiped
7 .333 solidtronque
8 45 90 360 {
9 /iAngle exch def
10 [ iAngle cos % a
11   iAngle sin % b
12   0          % c
13   -1.25      % -d
14 ] solidplansepare
15 } for
16 dup [.4 .5] solidputhuecolors
17 solidlightOn
18 drawsolid**}
19 \end{pspicture}

```

### 3.2 Exemple 2

Le principe est identique à l'exemple précédent mais troncatures et chanfreinages sont réalisées avec des plans de coupe.





```

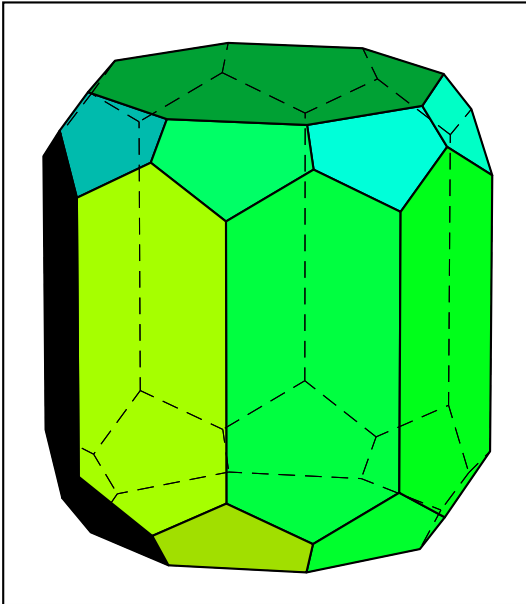
\psset{SphericalCoor=true,viewpoint=100 -20
10,Decran=100}
\psset{lightsrc=92.5 -33.7 17.4}
\begin{pspicture}(-3.5,-4)(3.5,4)
\psframe(-3.5,-4)(3.5,4)
\codejps{
4 4 6 newparallelepiped
45 90 360 {
/iAngle exch def
/n_x iAngle cos 35.2644 cos mul def
/n_y iAngle sin 35.2644 cos mul def
/n_z 35.2644 sin def
/distance 2 3 add 3 sqrt div neg def
[n_x n_y n_z distance]
solidplansepare
} for
45 90 360 {
/iAngle exch def
/n_x iAngle cos 35.2644 cos mul def
/n_y iAngle sin 35.2644 cos mul def
/n_z 35.2644 sin neg def
/distance 2 3 add 3 sqrt div neg def
[n_x n_y n_z distance]
solidplansepare
} for
45 90 360 {
/iAngle exch def
% plan : ax+by+cz-d=0
[iAngle cos % a
iAngle sin % b
0 % c
-2.5 % -d]
solidplansepare
} for
dup [.5 .2] solidputhuecolors
solidlightOn
drawsolid*}
\end{pspicture}

```

## 4 Coupes d'un prisme octogonal

Suivant le plan de coupe et/ou le point de vue on pourra noter quelques “bogues” : faces coupées absentes ou bien qui débordent sur les voisines.

### 4.1 Troncature des sommets d'un prisme octogonal

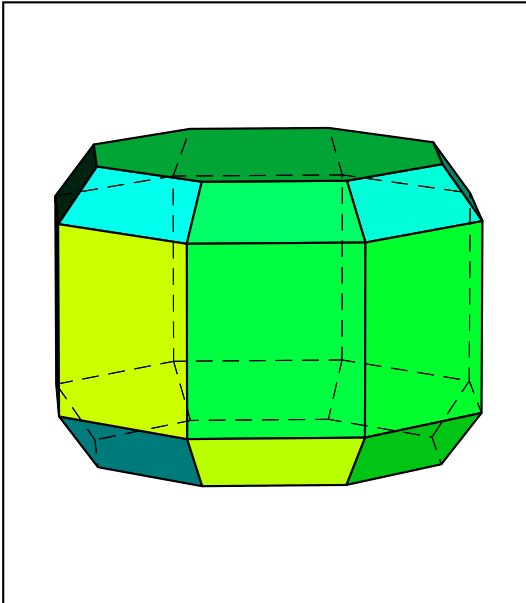


```

\psset{SphericalCoor=true,viewpoint=100 10 15,
  Decran=100}
\psset{lightsrc=81.4 47 34.2}
\begin{pspicture}(-3.5,-4)(3.5,4)
\psframe(-3.5,-4)(3.5,4)
\codejps{
-3 3 3 [1 8] newcylindre
0 45 315 {
/iAngle exch def
/n_x iAngle cos 0.816497 mul def
/n_y iAngle sin 0.816497 mul def
/n_z 0.57735 def
/distance -3.535534 def
[n_x n_y n_z distance]
solidplansepare
} for
0 45 315 {
/iAngle exch def
/n_x iAngle cos 0.816497 mul def
/n_y iAngle sin 0.816497 mul def
/n_z 0.57735 neg def
/distance -3.535534 def
[n_x n_y n_z distance]
solidplansepare
} for
dup [.5 .2] solidputhuecolors
solidlightOn
drawsolid*}
\end{pspicture}

```

## 4.2 Chanfreinage des arêtes des bases d'un prisme octogonal



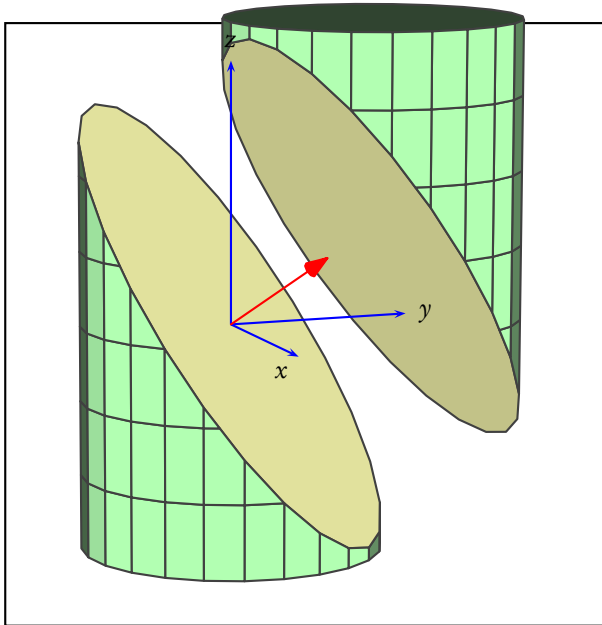
```

\psset{SphericalCoor=true,viewpoint=100 20 10,
  Decran=100}
\psset{lightsrc=81.4 47 34.2}
\begin{pspicture}(-3.5,-4)(3.5,4)
\psframe(-3.5,-4)(3.5,4)
\codejps{
solidlightOn
-2 3 2 [1 8] newcylindre
22.5 45 337.5 {
/iAngle exch def
/n_x iAngle cos 0.816497 mul def
/n_y iAngle sin 0.816497 mul def
/n_z 0.57735 def
/distance -3 def
[n_x n_y n_z distance]
solidplansepare
} for
22.5 45 337.5 {
/iAngle exch def
/n_x iAngle cos 0.816497 mul def
/n_y iAngle sin 0.816497 mul def
/n_z 0.57735 neg def
/distance -3 def
[n_x n_y n_z distance]
solidplansepare
} for
dup [.5 .2] solidputhuecolors
drawsolid*}
\end{pspicture}

```

## 5 Coupes d'un cylindre

### 5.1 Coupe du solide plein

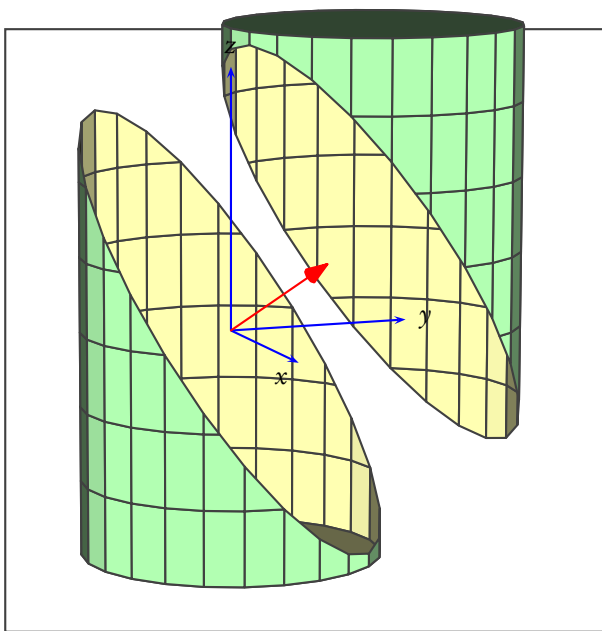


```

1 \begin{pspicture}(-3,-4)(5,4)
2 \psframe(-3,-4)(5,4)
3 \psset{SphericalCoor=true,viewpoint=50
4   -20 10,Decran=50,linecolor=darkgray}
5 \psset{lightsrc=92.5 -33.7 17.4}
6 \codejps{
7 solidlightOn
8 -3 2 3 [6 24] newcylindre
9 [1 1 1 -0.0001] solidplansepare
10 exch
11 {0 0 60 rotate0point3d} solidtransform
12 {0 2 1 translatepoint3d} solidtransform
13 dup (0.7 1 0.7 setrgbcolor)
14   outputcolors
15   dup 0 (1 1 0.7 setrgbcolor)
16   solidputfcolor
17 % dup
18 drawsolid**
19 % solidnumfaces
20 dup (0.7 1 0.7 setrgbcolor)
21   outputcolors
22   dup 0 (1 1 0.7 setrgbcolor)
23   solidputfcolor
24 % dup
25 drawsolid**
26 % solidnumfaces
27 }
28 \axesIIID[linecolor=blue](0,0,0)
29   (2.5,2.5,3.5)
30 \Normale[linecolor=red,fillcolor=red]
31   (0,45,35.2644)
32 \end{pspicture}

```

### 5.2 Coupe du solide creux

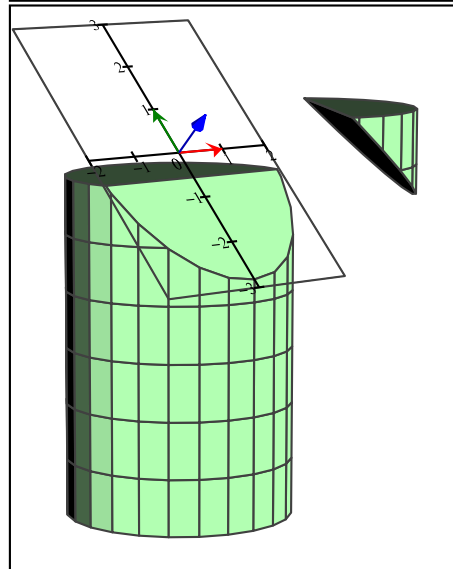
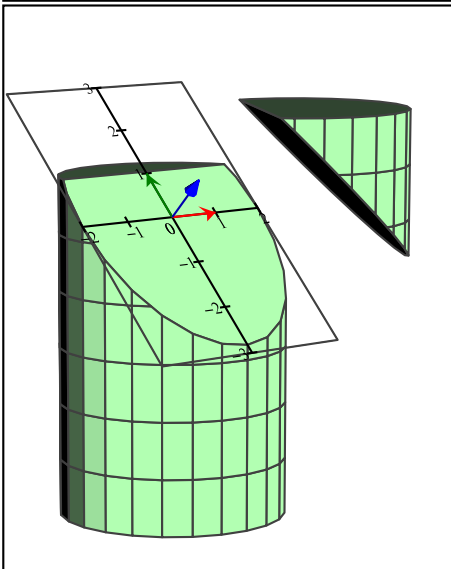
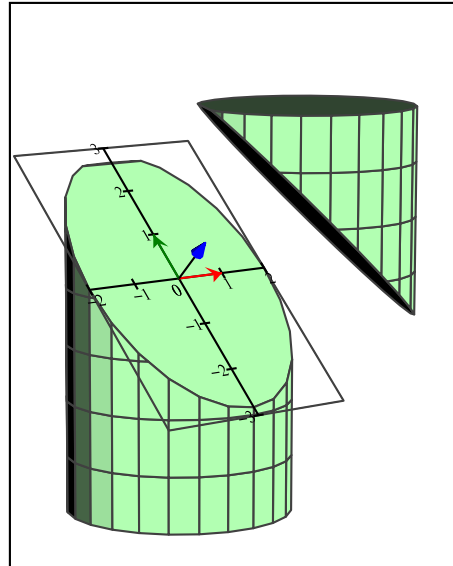
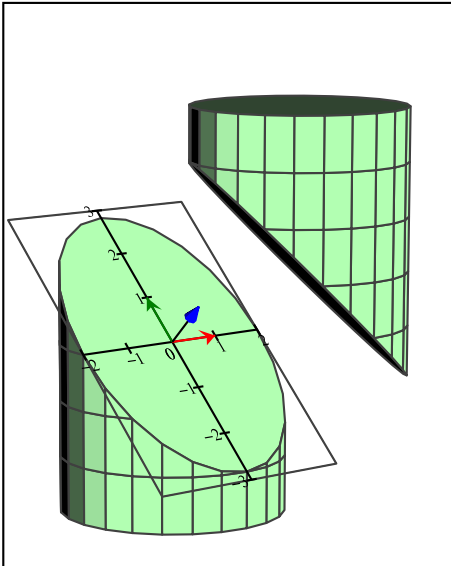


```

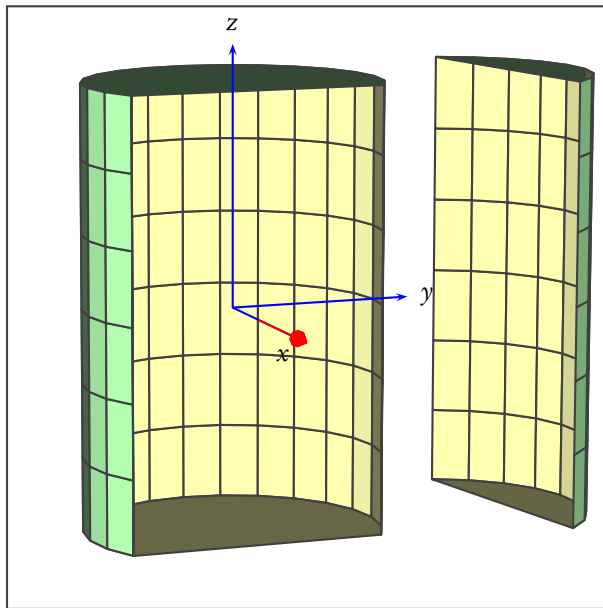
1 \begin{pspicture}(-3,-4)(5,4)
2 \psset{SphericalCoor=true,viewpoint=50
3       -20 10,Decran=50,linecolor=darkgray}
4 \psset{lightsrc=92.5 -33.7 17.4}
5 \psframe(-3,-4)(5,4)
6 \codejps{
7   solidlight0n
8   -3 2 3 [6 24] newcylindre
9   [1 1 1 0] solidplansepare
10  exch
11  dup [0] solidrmfaces
12  dup videsolid
13  {0 0 60 rotate0point3d} solidtransform
14  {0 2 1 translatepoint3d} solidtransform
15  dup (1 1 0.7 setrgbcolor) (0.7 1 0.7
16    setrgbcolor) inoutputcolors
17  drawsolid**
18  dup [0] solidrmfaces
19  dup videsolid
20  dup (1 1 0.7 setrgbcolor) (0.7 1 0.7
21    setrgbcolor) inoutputcolors
22  drawsolid**
23 }
24 \axesIIID[linecolor=blue](0,0,0)
25   (2.5,2.5,3.5)
26 \Normale[linecolor=red](0,45,35.2644)
27 \end{pspicture}

```

### 5.3 Coupe en faisant varier la distance du plan



## 5.4 Coupe par un plan parallèle à l'axe



```

1 \psset{SphericalCoor=true,
   viewpoint=50 -20 10,Decran=50,
   linecolor=darkgray}
2 \psset{lightsrc=92.5 -33.7 17.4}
3 \begin{pspicture}(-3,-4)(5,4)
4 \psframe(-3,-4)(5,4)
5 \codejps{
6   solidlightOn
7   -3 2 3 [6 24] newcylindre
8   [1 0 0 -1.001] solidplansepare
9   exch
10  dup [0] solidrmfaces
11  dup videsolid
12  {0 0 100 rotate0point3d}
   solidtransform
13  {0 3 0 translatepoint3d}
   solidtransform
14  dup (1 1 0.7 setrgbcolor) (0.7 1
   0.7 setrgbcolor)
   inoutputcolors
15  %dup
16  drawsolid**
17  %solidnumfaces
18  dup [0] solidrmfaces
19  dup videsolid
20  dup (1 1 0.7 setrgbcolor) (0.7 1
   0.7 setrgbcolor)
   inoutputcolors
21  drawsolid**
22 }
23 \axesIIID[linecolor=blue](0,0,0)
   (2.5,2.5,3.5)
24 \Normale[linecolor=red,fillcolor
   =red](1,0,0)
25 \end{pspicture}

```

## 6 Coupe d'un cône par un plan parallèle à l'axe

Dans une première étape, on fabrique le bi-cône en fusionnant deux cônes de sommet commun :

```

% fabrique du bi-cone
\codejps{
-5 3 0 [18 60] newcone
 5 3 0 [18 60] newcone
dup solidfacesreverse
solidfuz
(bicone) writesolidfile
}

```

Ces lignes pourront être supprimées par la suite. Dans une deuxième étape, le bi-cône est partagé par le plan choisi et les paramètres des deux parties sont conservés dans des fichiers de données :

```

(bicone) readsolidfile
[1 0 0 -1] solidplansepare
exch
{0 0 90 rotate0point3d} solidtransform
{0 5 0 translatepoint3d} solidtransform
(biconepart0) writesolidfile
(biconepart1) writesolidfile

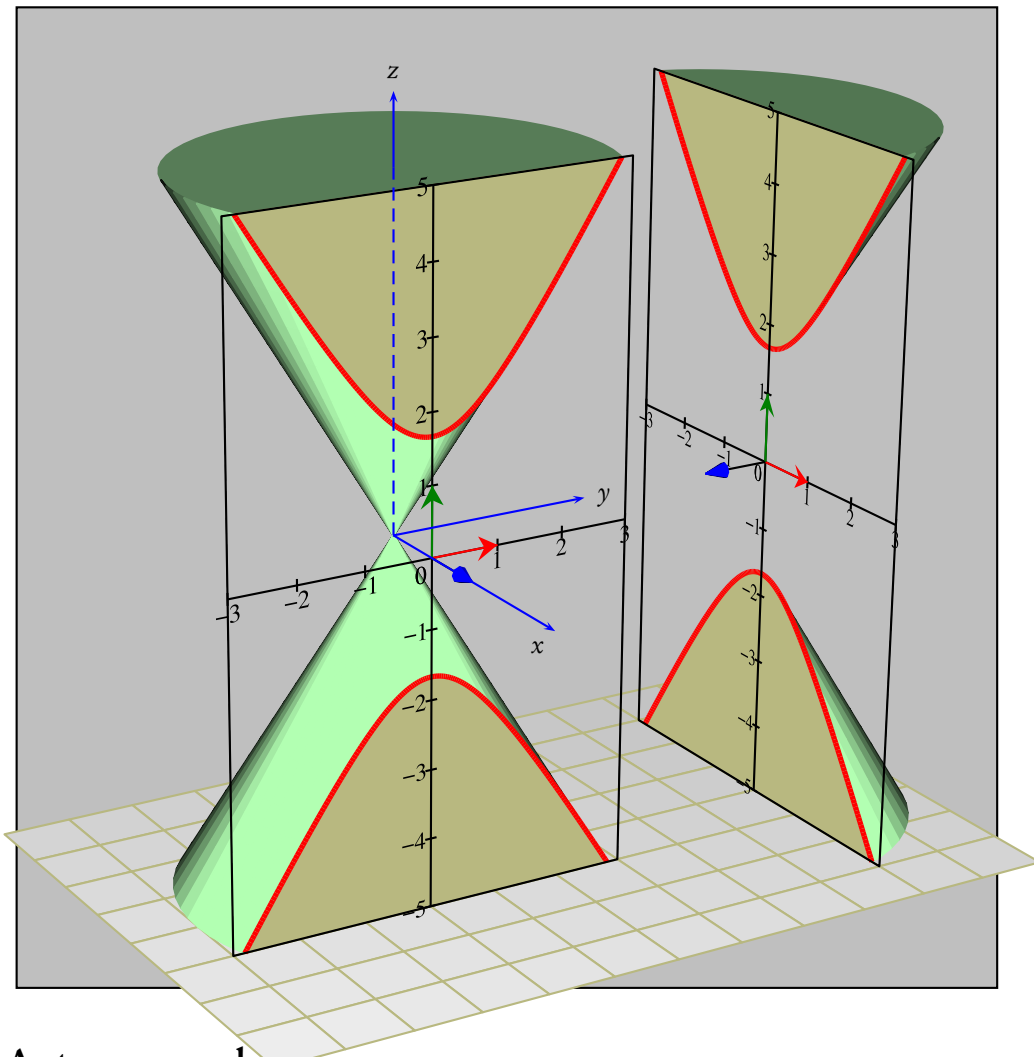
```

Chacune des parties est ensuite lue et traitée, on notera, en particulier, que la section comporte deux faces de découpe **0** et **1** qui sont coloriées différemment. Toute cette partie d'écriture des données sur la disque n'est effectuée qu'une fois et sera effacée par la suite.

```

(bicone-part0) readsolidfile
dup (0.7 1 0.7 setrgbcolor) outputcolors
dup 0 (0.72 0.72 0.5 setrgbcolor) solidputfcolor
dup 1 (0.72 0.72 0.5 setrgbcolor) solidputfcolor
drawsolid**
(bicone-part1) readsolidfile
dup (0.7 1 0.7 setrgbcolor) outputcolors
dup 0 (0.72 0.72 0.5 setrgbcolor) solidputfcolor
dup 1 (0.72 0.72 0.5 setrgbcolor) solidputfcolor
drawsolid**

```



## 7 Autres exemples

1. Vous trouverez une version codée de ce document en **PStricks** dans le lien suivant :  
<http://melusine.eu.org/syracuse/mluque/solides3d2007/sections/doc-sections>
2. Une étude des sections coniques sur :  
<http://melusine.eu.org/syracuse/mluque/solides3d2007/sections/sections-cone>
3. Une étude des sections cylindriques sur :  
<http://melusine.eu.org/syracuse/mluque/solides3d2007/sections/section-cylindre>
4. Une étude sur les sections du tore :  
<http://melusine.eu.org/syracuse/mluque/solides3d2007/sections/section-tore>