

Les options et les objets ‘cachés’ de pstsolides3d

27 janvier 2008

Résumé

Ce document présente les options et les objets qui n’ont pas pu être traités dans la documentation générale.

- L’option [transform] ;
- les options [inhue] et [inouthue] ;
- l’objet [object=vecteur] ;
- l’objet [object=polygoneregulier] ;
- l’objet [object=objfile].

Table des matières

1	L’option transform	1
1.1	Facteur d’échelle identique appliqué aux trois coordonnées	2
1.2	Facteur d’échelle différent pour les trois coordonnées	3
1.3	Transformation liée à la distance du point à l’origine	5
1.4	Torsion d’une poutre	7
1.5	Torsion d’une sphère	7
1.6	Transformation de Nerfertiti	8
2	Les options [inhue] et [inouthue]	10
3	Les objets vecteur et ligne	11
3.1	Définitions	11
3.2	Normale à une sphère	12
3.3	Normale en un point d’une surface définie par $z = f(x,y)$	13
4	Inclure un objet 3D de type .obj : object=objfile	13
5	Objet : [object=polygoneregulier]	15

1 L’option transform

Pour la démonstration, l’objet qui subira la transformation est un cube. Le cube de référence est en jaune, le cube transformé en vert et le cube en fil de fer représente le cube avant transformation.

1.1 Facteur d'échelle identique appliqué aux trois coordonnées

Le facteur d'échelle est pris égal à 0.5. On l'introduit soit en définissant la variable '/Facteur' :

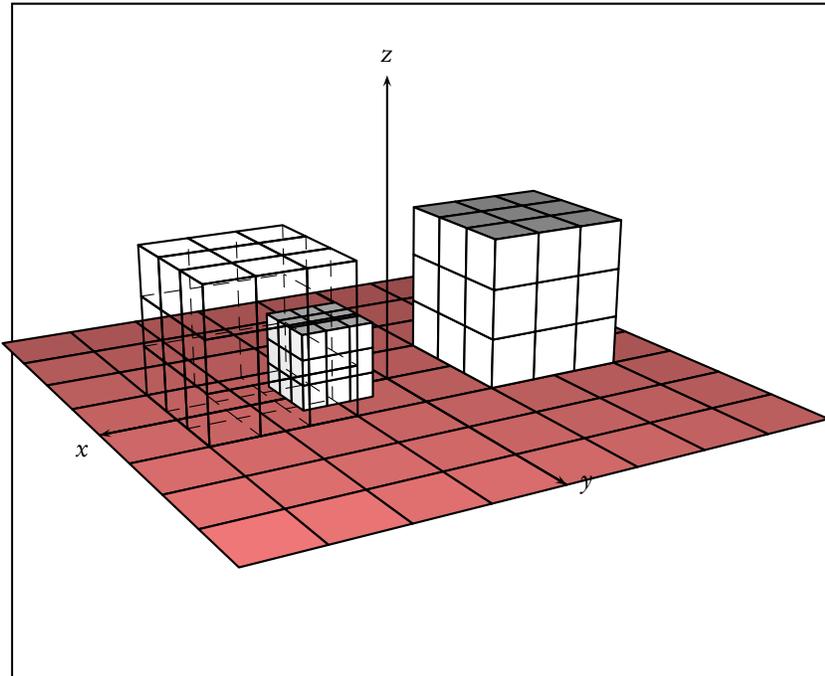
```
\pstVerb{/Facteur {.5 mulv3d} def}%
```

puis en l'introduisant dans l'option 'transform' :

```
\psSolid[object=cube,a=2,ngrid=3,  
transform=Facteur](2,0,1)%
```

soit directement dans le code :

```
\psSolid[object=cube,a=2,ngrid=3,  
transform={.5 mulv3d}](2,0,1)%
```

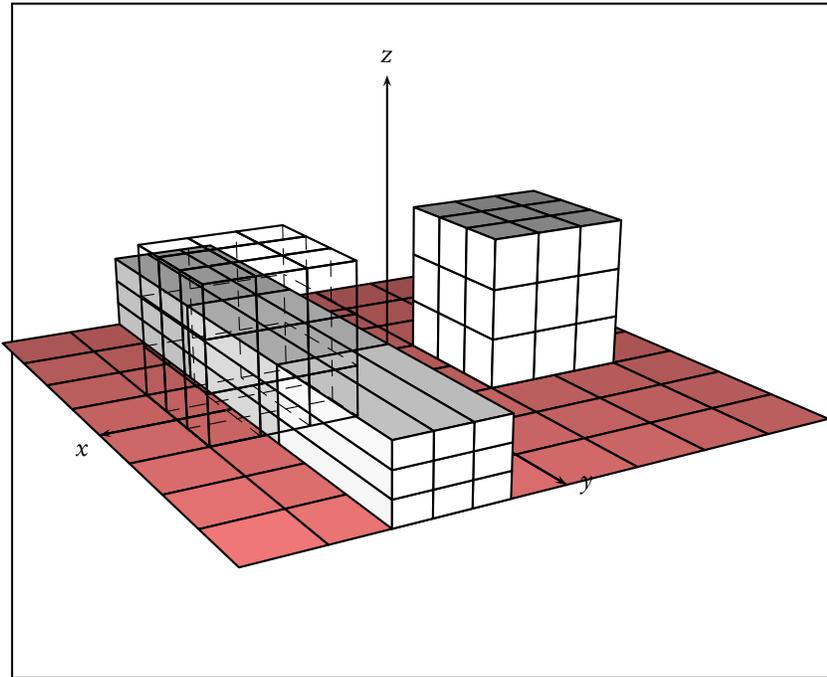


```
1 \psset{SphericalCoord,viewpoint=20 60 20,lightsrc=10 15 7,Decran=20}  
2 \begin{pspicture}(-5,-5)(6,5)  
3 \psframe(-5,-4)(6,5)  
4 \psSolid[object=grille,base=-4 4 -4 4,fillcolor=red!50]%  
5 \axesIIIID(0,0,0)(4,4,4)%  
6 \psSolid[object=cube,  
7 a=2,ngrid=3](-2,0,1)  
8 \psSolid[object=cube,  
9 a=2,transform={.5 mulv3d},  
10 ngrid=3](2,0,1)  
11 \psSolid[object=cube,  
12 action=draw,  
13 a=2,ngrid=3](2,0,1)  
14 \end{pspicture}
```

Le facteur d'échelle s'applique aussi aux coordonnées de la position du centre du cube.

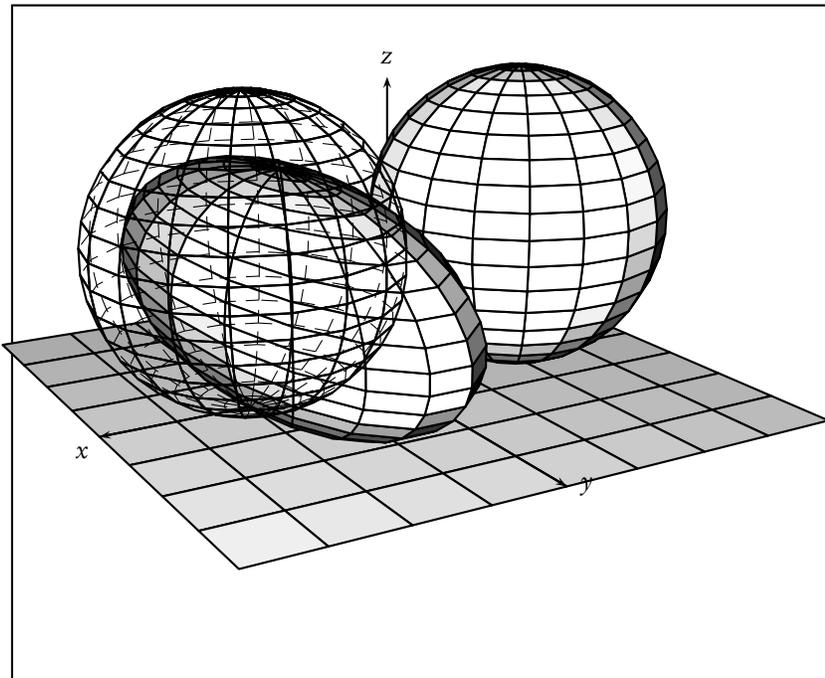
1.2 Facteur d'échelle différent pour les trois coordonnées

Prenons, par exemple, que l'on applique un facteur de 0.75 pour x , 2 pour y et 0.5 pour z , on transforme ainsi un cube en un parallélépipède.



```
1 \psset{SphericalCoord,viewpoint=20 60 20,lightsrc=10 15 7,Decran=20}
2 \begin{pspicture}(-5,-5)(6,5)
3 \psframe(-5,-4)(6,5)
4 \psSolid[object=grille,base=-4 4 -4 4,fillcolor=red!50]%
5 \axesIIIID(0,0,0)(4,4,4)%
6 \pstVerb{
7 /XYZscale {
8   dict begin
9     /M defpoint3d % on récupère les coordonnées
10    /X' M pop pop 0.75 mul def % X' = 0.75*X
11    /Y' M pop 4 mul exch pop def % Y' = 4*Y
12    /Z' M 0.5 mul 3 -2 roll pop pop def % Z' = 0.5*Z
13    X' Y' Z'
14  end
15 } def}%
16 \psSolid[object=cube,
17   a=2,ngrid=3](-2,0,1)
18 \psSolid[object=cube,
19   a=2,transform=XYZscale,
20   ngrid=3](2,0,1)
21 \psSolid[object=cube,
22   action=draw,
23   a=2,ngrid=3](2,0,1)
24 \end{pspicture}
```

Il peut être amusant d'appliquer une telle transformation à une sphère, voici le résultat : la sphère ressemble maintenant à un ballon de rugby !

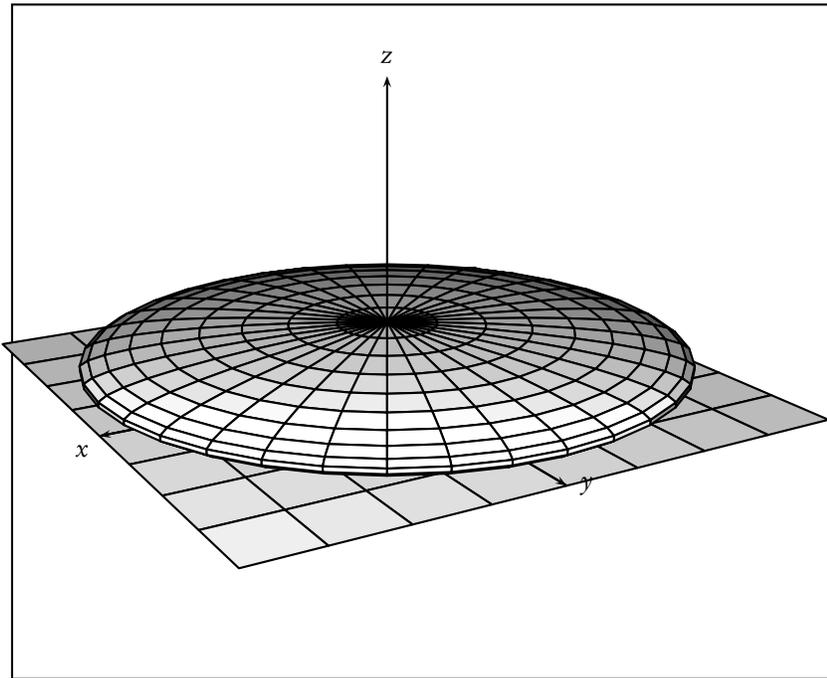


```

1 \psset{SphericalCoor,viewpoint=20 60 20,lightsrc=10 15 7,Decran=20}
2 \begin{pspicture}(-5,-5)(6,5)
3 \psframe(-5,-4)(6,5)
4 \psSolid[object=grille,base=-4 4 -4 4]%
5 \axesIIID(0,0,0)(4,4,4)%
6 \pstVerb{
7 /XYZscale {
8 4 dict begin
9   /M defpoint3d % on récupère les coordonnées
10  /X' M pop pop 0.75 mul def % X' = 0.75*X
11  /Y' M pop 2 mul exch pop def % Y' = 2*Y
12  /Z' M 0.75 mul 3 -2 roll pop pop def % Z' = 0.75*Z
13  X' Y' Z'
14 end} def}%
15 \psSolid[object=sphere,
16   r=2,ngrid=18 18](-2,0,2)
17 \psSolid[object=sphere,
18   r=2,transform=XYZscale,
19   ngrid=18 18](2,0,2)
20 \psSolid[object=sphere,
21   action=draw,
22   r=2,ngrid=18 18](2,0,2)
23 \end{pspicture}

```

Un autre (mal)traitement pour en faire une crêpe ou bien une soucoupe volante !



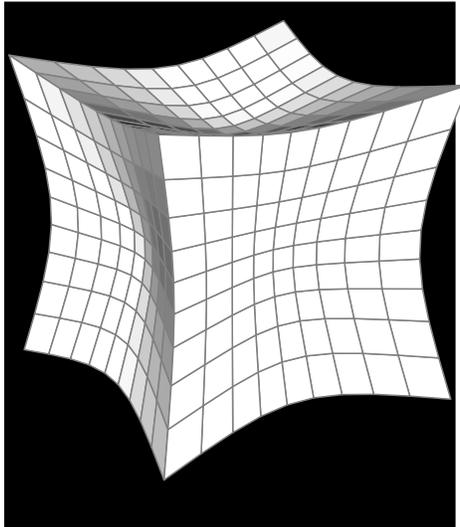
```

1 \pset{SphericalCoor,viewpoint=20 60 20,lightsrc=10 15 7,Decran=20}
2 \begin{pspicture}(-5,-5)(6,5)
3 \psframe(-5,-4)(6,5)
4 \psSolid[object=grille,base=-4 4 -4 4]%
5 \axesIIIID(0,0,0)(4,4,4)%
6 \pstVerb{
7 /XYZscale {
8 dict begin
9 /M defpoint3d % on récupère les coordonnées
10 /X' M pop pop 2 mul def % X' = 02*X
11 /Y' M pop 2 mul exch pop def % Y' = 2*Y
12 /Z' M 0.2 mul 3 -2 roll pop pop def % Z' = 0.2*Z
13 X' Y' Z'
14 end} def}%
15 \psSolid[object=sphere,
16 r=2,transform=XYZscale,
17 ngrid=18 36](0,0,2)
18 \end{pspicture}

```

1.3 Transformation liée à la distance du point à l'origine

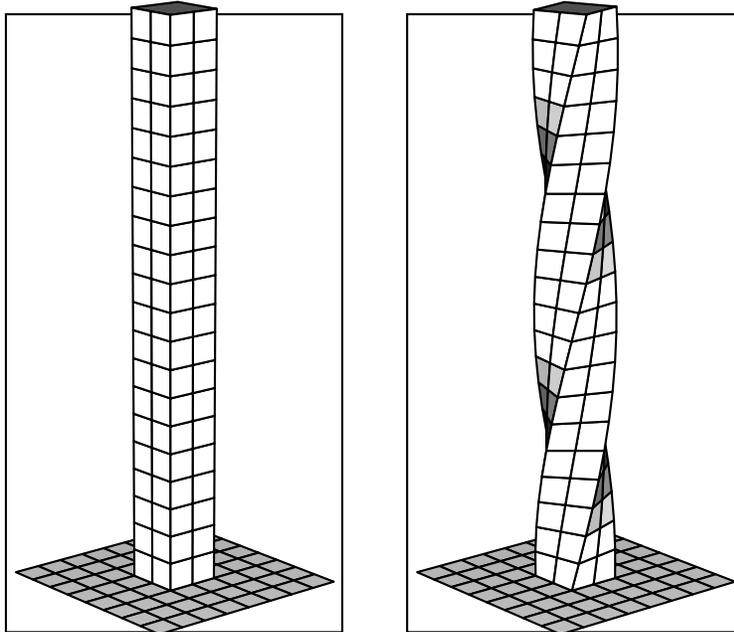
$$\begin{cases} x' = (0.5\sqrt{x^2 + y^2 + z^2} + 1 - 0.5\sqrt{3})x \\ y' = (0.5\sqrt{x^2 + y^2 + z^2} + 1 - 0.5\sqrt{3})y \\ z' = (0.5\sqrt{x^2 + y^2 + z^2} + 1 - 0.5\sqrt{3})z \end{cases}$$



```
1 \begin{pspicture}(-3,-4)(3,3)
2 \psset{SphericalCoor,viewpoint=20 60 20,lightsrc=10 15 7,Decran=20}
3 \pstVerb{
4 /gro {
5   dict begin
6     /M defpoint3d
7     /a .5 def
8     /b 1 a 3 sqrt mul sub def
9     /k M norme3d a mul b add def
10    M k mulv3d
11  end
12 } def}%
13 \psframe*(-3,-4)(3,3)
14 \psset{linewidth=.02,linecolor=gray}
15 \psSolid[object=cube,a=3,ngrid=9,
16         transform=gro]%
17 \end{pspicture}
```

1.4 Torsion d'une poutre

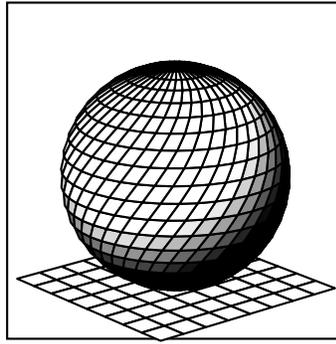
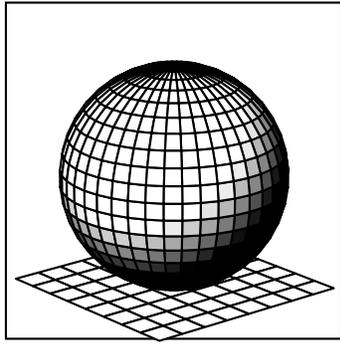
Le solide de départ est un prisme de hauteur 10 cm de 20 étages ($ngrid=20$ 2). À chaque étage, on applique une rotation supplémentaire d'axe Oz et de valeur 10° par exemple. Comme les niveaux sont espacés de 0,5 cm, on multiplie $z \times 20$.



```
1 \psset{SphericalCoor,viewpoint=50 50 20,lightsrc=25 37 17,Decran=50,unit=0.75}
2 \begin{pspicture}(-3,-1)(3.5,10)
3 \psframe(-3,-1)(3,10)
4 \psSolid[object=grille,base=-2 2 -2 2,ngrid=8]%
5 \psSolid[object=prisme,h=10,ngrid=20 2,
6     base=0.5 0 0.5 0.5 0 0.5 -0.5 0.5 -0.5 0 -0.5 -0.5 0 -0.5 0.5 -0.5]%
7 \end{pspicture}
8 \begin{pspicture}(-3.5,-1)(3,10)
9 \psframe(-3,-1)(3,10)
10 \psSolid[object=grille,base=-2 2 -2 2,ngrid=8]%
11 \pstVerb{
12 /torsion {
13 2 dict begin
14 /M defpoint3d % on récupère les coordonnées
15 /AngleTorsion M 20 mul 3 -2 roll pop pop def
16 % on tourne de 10 degrés à chaque niveau
17 M 0 0 AngleTorsion rotate0point3d
18 end} def}%
19 \psSolid[object=prisme,h=10,ngrid=20 2,
20     base=0.5 0 0.5 0.5 0 0.5 -0.5 0.5 -0.5 0 -0.5 -0.5 0 -0.5 0.5 -0.5,
21     transform=torsion]%
22 \end{pspicture}
```

1.5 Torsion d'une sphère

On peut appliquer cette torsion au maillage d'une sphère :



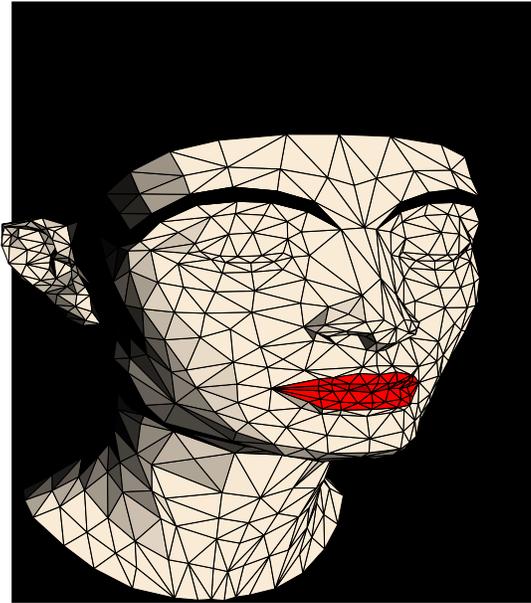
```

1 \psset{SphericalCoor,viewpoint=50 50 20,lightsrc=25 10 30,Decran=50,unit=0.75}
2 \begin{pspicture}(-3,-1)(3.5,5)
3 \psframe(-3,-1)(3,5)
4 \psSolid[object=grille,base=-2 2 -2 2,ngrid=8]%
5 \psSolid[object=sphere,r=2,ngrid=20 40](0,0,2)%
6 \end{pspicture}
7 \begin{pspicture}(-3.5,-1)(3,5)
8 \psframe(-3,-1)(3,5)
9 \psSolid[object=grille,base=-2 2 -2 2,ngrid=8]%
10 \pstVerb{
11 /torsion {
12 2 dict begin
13   /M defpoint3d % on récupère les coordonnées
14   /AngleTorsion M 20 mul 3 -2 roll pop pop def
15   % on tourne de 10 degrés à chaque niveau
16   M 0 0 AngleTorsion rotate0point3d
17 end} def}%
18 \psSolid[object=sphere,r=2,ngrid=20 40,transform=torsion](0,0,2)
19 \end{pspicture}

```

1.6 Transformation de Nefertiti

Une autre vision de Nefertiti, l'aimerez-vous mieux ainsi ?



```

1 \psset{unit=0.5}
2 \definecolor{AntiqueWhite}{rgb}{0.98,0.92,0.84}
3 \begin{pspicture}(-7,-7)(7,9)
4 \pstVerb{
5 /XYZscale {
6 4 dict begin
7   /M defpoint3d % on récupère les coordonnées
8   /X' M pop pop 2.5 mul def % X' = 2.5*X
9   /Y' M pop 2 mul exch pop def % Y' = 2*Y
10  /Z' M 1.5 mul 3 -2 roll pop pop def % Z' = 1.5*Z
11   X' Y' Z'
12 end} def}%
13 \psset{lightsrc=30 -40 10}
14 \psframe*(-7,-7)(7,9)
15 \psset{SphericalCoor,viewpoint=50 -80 10,Decran=30,linewidth=0.5\pslinewidth}
16 \psset{RotX=90,RotZ=30,sommets= (sommets_nefer.dat) run}
17 \psSolid[object=new,fillcolor=AntiqueWhite,
18   faces={(faces_nefer.dat) run},transform=XYZscale]%
19 \psSolid[object=new,fillcolor=red,
20   faces={(faces_nefer_levres.dat) run},transform=XYZscale]%
21 \psSolid[object=new,fillcolor=black,
22   faces={(faces_nefer_sourcils.dat) run},transform=XYZscale]%
23 \end{pspicture}

```

2 Les options [inhue] et [inouthue]

Pour colorier les faces d'un solide nous disposons de deux options pour l'extérieur :

- [fillcolor]
- [hue]

et d'une seule pour l'intérieur :

- [incolor]

Les détails de ces options sont détaillées dans la documentation générale du package.

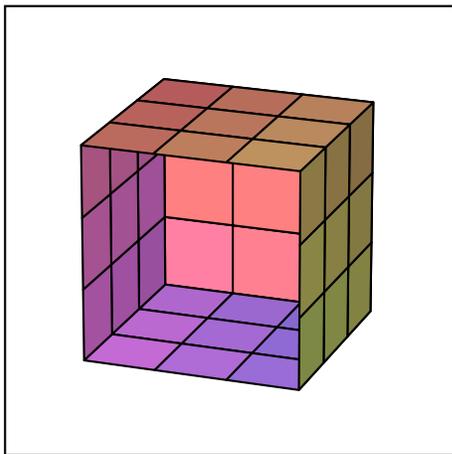
Nous avons, à présent, deux nouvelles options :

- [inhue]

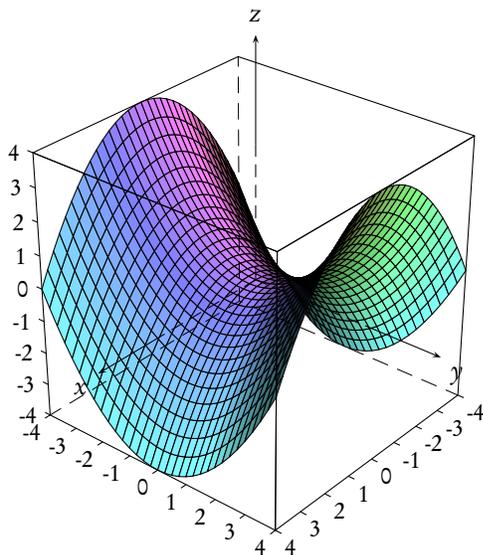
qui permet de colorier les faces intérieures avec les mêmes paramètres que [hue] et,

- [inouthue]

qui peindra dans la continuité faces intérieures et extérieures avec les paramètres de [hue]. On rappelle que pour voir les faces intérieures il faut activer l'option **hollow**, comme dans les deux exemples suivants.

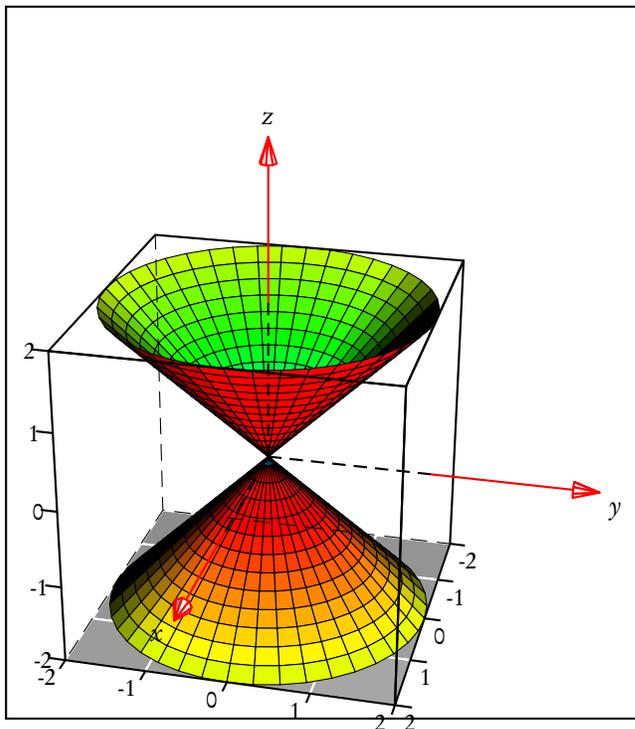


```
1 \psset{lightsrc=45 15 20,SphericalCoor=true,
2   viewpoint=50 20 20,Decran=50}
3 \begin{pspicture}(-3,-3)(3,3)
4 \psframe(-3,-3)(3,3)
5 \psSolid[object=cube,
6   a=3,ngrid=3,
7   hollow,
8   inouthue=0 1 0.5 1,
9   rm=36 37 38 39 40 41 42 43 44]{}
10 \end{pspicture}
```



```
1 \psset{unit=0.5}
2 \psset{lightsrc=30 30 25}
3 \psset{SphericalCoor,viewpoint=50 40 30,Decran=50}
4 \begin{pspicture}(-6,-8)(7,8)
5 \psSurface[ngrid=.25 .25,inouthue=1 0 0.5 1,
6   linewidth=0.5\pslinewidth,axesboxed,
7   algebraic](-4,-4)(4,4){%
8   ((y^2)-(x^2))/4 }
9 \end{pspicture}
```

Pour colorier avec les paramètres de [hue] les faces intérieures et extérieures on utilisera l'option [hue] pour l'extérieur et l'option [inhue] pour l'intérieur, comme dans l'exemple suivant :



```

\psset{unit=0.5}
\begin{pspicture}(-7,-7)(10,12)
\psframe(-7,-7)(10,12)
\psset[pst-solides3d]{viewpoint=20 5
10,
Decran=50,lightsrc=20 10 5}
\psSolid[object=grille,base=-2 2 -2
2,
linecolor=white](0,0,-2)
% Parametric Surfaces
\defFunction{cone}(u,v){u v Cos mul
}{u v Sin mul}{u}
\psSolid[object=surfaceparametree,
base=-2 2 0 2 pi mul,
%
% inouthue=0 1,
% inhue=0.8 0.2,hue=0.8 0.2, %
% problème !
function=cone,linewidth=0.5\
pslinewidth,ngrid=25 40]%
\psgridIID[Zmin=-2,Zmax=2](-2,2)
(-2,2)
\end{pspicture}

```

3 Les objets vecteur et ligne

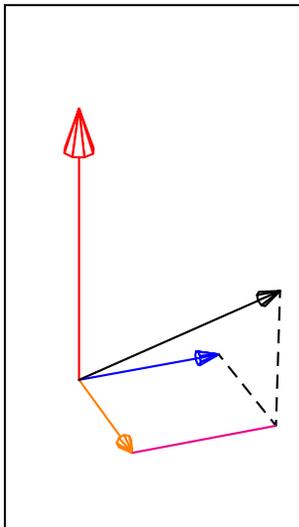
3.1 Définitions

L'objet **vecteur** dessine un vecteur en 3 dimensions dont la pointe est un cône. Les coordonnées du sommet sont contenues dans le paramètre [args=x y z] qui dessinera un vecteur d'origine O et d'extrémité le point de coordonnées (x,y,z).

Ce vecteur peut subir une translation déterminée par les coordonnées de la position du solide. Ce vecteur étant interprété comme un solide pourra aussi subir des rotations avec les paramètres RotX= etc..

La taille du vecteur est modifiable avec l'option transform, par exemple [transform=2 mulv3d] multiplie la taille de la flèche par 2.

L'objet **ligne** dessine une ligne brisée dont les coordonnées des points sont donnés dans l'ordre dans le paramètre [base=x0 y0 z 0 x1 y1 z1 ...xn yn zn] .



```

1 \psset{SphericalCoor=true,viewpoint=50 -20 30,Decran=50}
2 \begin{pspicture}(-1,-2)(3,5)
3 \psframe(-1,-2)(3,5)
4 \psSolid[object=vecteur,
5   args=0 0 2,
6   transform={2 mulv3d},
7   linecolor=red]%
8 \psSolid[object=vecteur,
9   args=2 0 0,linecolor=orange]%
10 \psSolid[object=vecteur,
11  args=0 2 0,linecolor=blue]%
12 \psSolid[object=vecteur,
13  args=2 2 2]%
14 \psSolid[object=ligne,linestyle=dashed,
15  base=2 2 2
16  2 2 0
17  0 2 0]%
18 \psSolid[object=ligne,linecolor=magenta,
19  base=2 2 0
20  2 0 0]%
21 \end{pspicture}

```

Les applications sont variées, on pourra par exemple faire un quadrillage 3D avec les axes représentés par ces vecteurs : voir la macro `\psgridIIID` utilisée dans l'exemple sur les options `[inhue]` et `[hue]` et qui est définie dans le préambule du document ou bien tracer la normale en différents points d'une surface comme dans les deux exemples suivants.

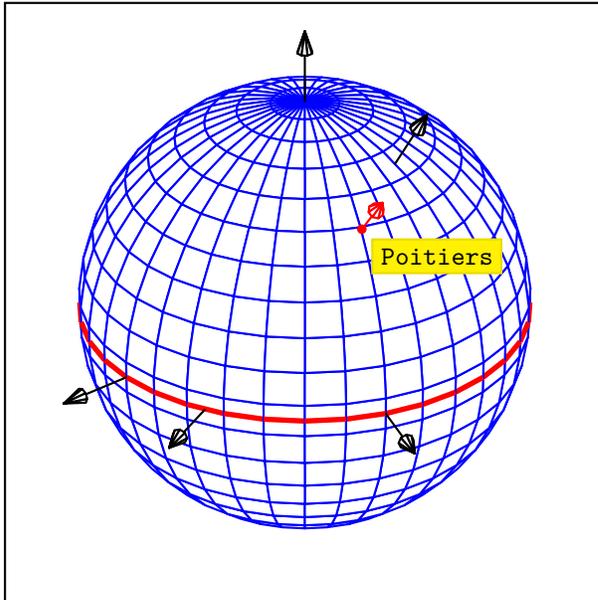
3.2 Normale à une sphère

La normale à la surface de la sphère est prédéfinie dans une commande :

```

\def\NormalSphere#1#2#3{%
% #1 rayon
% #2 longitude
% #3 latitude
\pstVerb{/latitude #3 def
        /longitude #2 def
        /Rayon #1 def
        /rotY latitude neg def
        /rotZ longitude def
        /xP Rayon longitude cos latitude cos mul mul def
        /yP Rayon longitude sin latitude cos mul mul def
        /zP Rayon latitude sin mul def}%
\psSolid[object=vecteur,RotZ=rotZ,RotY=rotY,
        args=1 0 0](xP,yP,zP)}

```



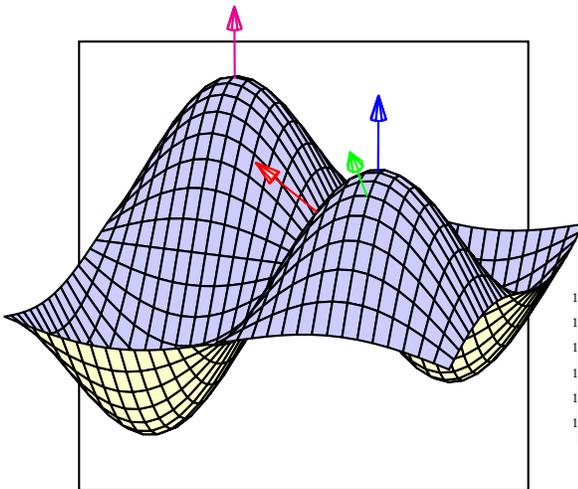
```

\psset{SphericalCoor=true,viewpoint=50
-20 30,Decran=50}
\begin{pspicture}(-4,-4)(4,4)
\psframe(-4,-4)(4,4)
\psSolid[object=sphere,r=3,ngrid=20 36,
linecolor=blue,
tracelignedeniveau=true,
hauteurlignedeniveau=0,
linewidthlignedeniveau=2,
couleurlignedeniveau=red]%
\NormalSphere{3}{30}{60}%
\NormalSphere{3}{0}{90}%
\NormalSphere{3}{0}{0}%
\NormalSphere{3}{-45}{0}%
\NormalSphere{3}{-70}{0}%
{\psset{linecolor=red}\NormalSphere
{3}{0.33}{46.6}}
% \psSolid[object=ligne,
%         base=0 0 0 xP yP zP,
%         linecolor=red,linestyle=dashed
% ]%
\psPoint(xP,yP,zP){P}
\psdot[linecolor=red](P)
\uput[dr](P){\Cadre{Poitiers}}
\end{pspicture}

```

3.3 Normale en un point d'une surface définie par $z = f(x, y)$

On prend la surface d'équation $z = 2 \sin x \sin y$.



```

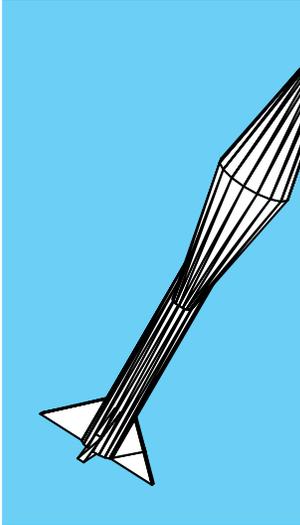
\psset{SphericalCoor=true,viewpoint=50 20 20,Decran
=50}
\begin{pspicture}(-3,-3)(3,4)
\psframe(-3,-3)(3,3)
\psSurface[algebraic,
ngrid=.2 .2,
incolor=yellow!20,fillcolor=blue
!20](-3,-3)(3,3){2*sin(x)*sin(y)}%
\psset{linecolor=red}%
\NormalSin{1.57}{0.75}%
\psset{linecolor=blue}%
\NormalSin{1.57}{1.57}%
\psset{linecolor=green}%
\NormalSin{2}{1.57}%
\psset{linecolor=magenta}%
\NormalSin{-1.57}{-1.57}%
\end{pspicture}

```

4 Inclure un objet 3D de type .obj : object=objfile

Ce sont des fichiers très utilisés dans le domaine de la 3D et qu'on trouve en abondance sur internet. On veillera à supprimer tous les commentaires # du fichier original et ne conserver que les sommets : ce sont les lignes qui commencent par v et les faces, lignes commençant par f.

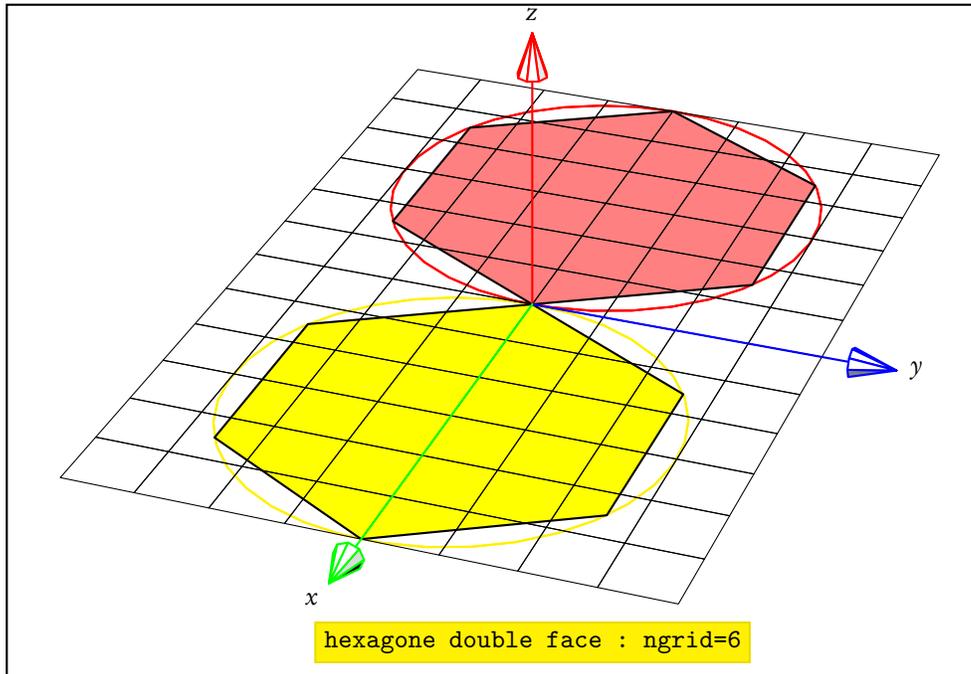
Les fichiers trop volumineux ne seront pas pris en compte car le nombre maximal d'éléments pour un tableau `postscript` est 65535. Donc il faut que le nombre de sommets soit inférieur à 21845.



```
1 \psset{SphericalCoor,viewpoint=20 15 10,Decran=20}  
2 \begin{pspicture}(-3,-4)(3,3)  
3 \psframe*[linecolor=cyan!50](-3,-4)(1,3)  
4 \psSolid[object=objfile,  
5     unit=20,RotX=60,  
6     file=rocket.obj] %  
7 \end{pspicture}
```

5 Objet : [object=polygoneregulier]

[object=polygoneregulier] dessine un polygone régulier double face, dont le nombre de côtés est fixé par le paramètre [ngrid=n], inscrit dans un cercle de rayon déterminé par [r=valeur]. Comme tous les solides, cet objet peut subir les rotations [RotX=valeur, RotY=...] et les translations.



```

1 \begin{pspicture}(-7,-5)(6,4)
2 \psframe(-7,-5)(6,4)
3 \defFunction[algebraic]{f1}(x){3*cos(x)+3}{3*sin(x)}{
4 \psProjection[object=courbeR2,
5   range=-3.14 3.14,linecolor=yellow,
6   normal=0 0 1,function=f1}%
7 \defFunction[algebraic]{f2}(x){3*cos(x)-3}{3*sin(x)}{
8 \psProjection[object=courbeR2,
9   range=-3.14 3.14,linecolor=red,
10  normal=0 0 1,function=f2}%
11 \psSolid[object=polygoneregulier,fillcolor=red!50,incolor=yellow,
12   r=3,ngrid=6](-3,0,0)
13 \psSolid[object=polygoneregulier,fillcolor=red!50,incolor=yellow,
14   r=3,ngrid=6,RotX=180](3,0,0)
15 \psSolid[object=grille,base=-6 6 -4 4,action=draw,linewidth=0.5\pslinewidth]%
16 \psSolid[object=vecteur,args=0 0 2,
17   transform={2 mulv3d},linecolor=red}%
18 \psSolid[object=vecteur,args=0 2.5 0,
19   transform={2 mulv3d},linecolor=blue}%
20 \psSolid[object=vecteur,args=3.5 0 0,
21   transform={2 mulv3d},linecolor=green}%
22 \psPoint(0,0,4){Z}\uput[u](Z){Z}
23 \psPoint(7,0,0){X}\uput[dl](X){X}
24 \psPoint(0,5,0){Y}\uput[r](Y){Y}
25 \rput(0,-4.5){\Cadre{hexagone double face : ngrid=6}}
26 \end{pspicture}

```