PSTricks

Images on a spherical mirror

A PSTricks package for drawing 3D objects on a sphere, v1.02

May 17, 2024

Package author(s):
Manuel Luque
Herbert Voß

Contents

Thanks for feedback and contributions to:
Bruno Guegan; Jean-Paul Vignault

1 Objectifs

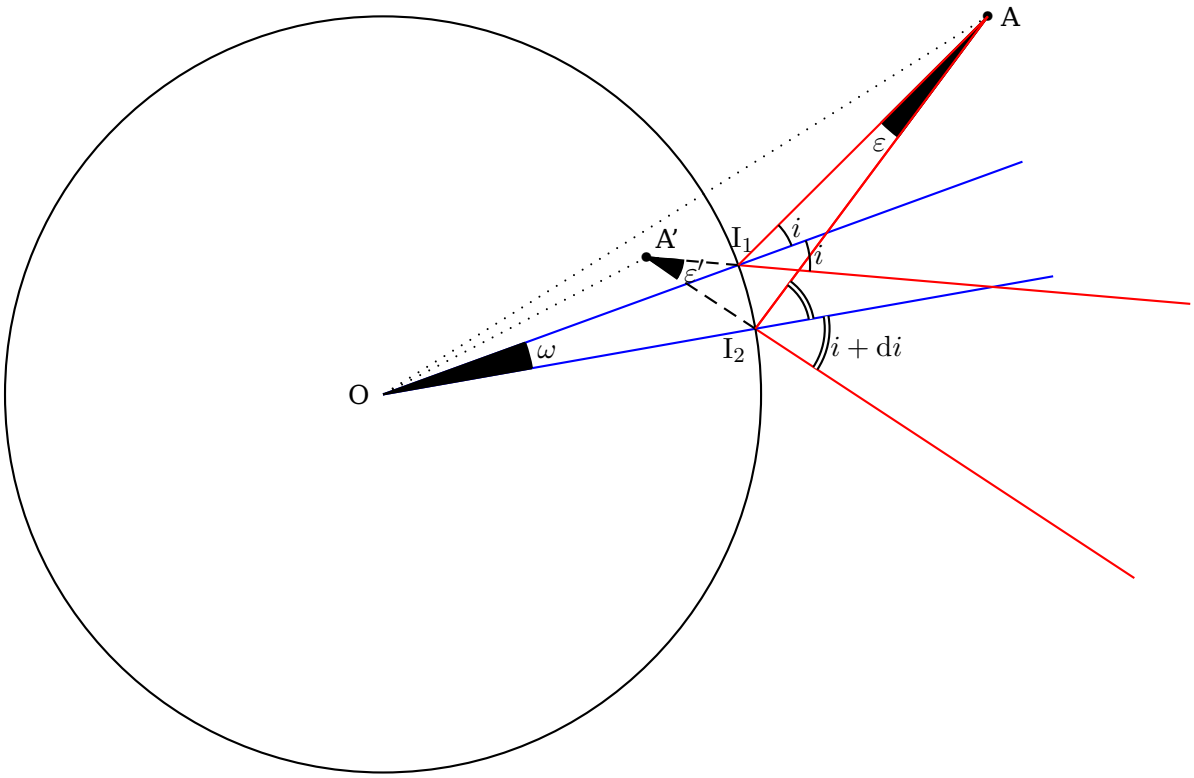
On voit souvent dans les villes, lorsque la visibilité à la sortie d'un garage ou à un carrefour est insuffisante, ou bien dans certains magasins pour des motifs de surveillance, des miroirs bombés donnant de l'environnement un panorama trés large. On peut prendre aussi une boule argentée ou dorée que l'on suspend aux branches des sapins de Noél, pour voir le décor se réfléchir dans la Sphere avec une distorsion bien particuliére.

L'objectif est de définir des éléments de décor en 3D, *des briques*, pour reconstruire ce qu'un observateur placé é une certaine distance d'une boule miroir, é la méme hauteur que son centre verrait.

Les *briques* é notre disposition sont :

- Un parallélépipéde rectangle donné par ses trois dimensions `A,B,C` : il peut donc se transformer en cube et méme en dé.
- Un quadrillage plan défini par son origine, la normale au plan et ses limites.
- Une ligne.
- Un rectangle défini comme le quadrillage.
- Un polygone.
- Un cercle défini par la normale é son plan, son origine et son rayon, un arc de cercle défini comme le cercle avec ses deux angles limites.
- Un point de l'espace.
- Un tétraédre donné par les coordonnées du centre de sa base et le rayon du cercle inscrivant chaque face, que l'on peut faire tourner.
- Une pyramide é base carrée donnée par le demi-cété de sa base et sa hauteur que l'on peut faire tourner et placer oé on veut.
- Une sphére donnée par les coordonnées du centre `CX=...,CY=...,CZ=...` et le rayon, que l'on peut faire tourner avec les paramétres `RotX=...,RotY=...,RotZ=...`.
- Un cylindre vertical défini par son rayon, sa hauteur, que l'on peut faire tourner avec les paramétres `RotX=...,RotY=...,RotZ=...` et dont le centre de la base se placera avec les paramétres `CX=...,CY=...,CZ=...`
- Un céne et tronc de céne définis par le rayon de la base, la hauteur et la fraction de hauteur qu'on souhaite retenir (pour le tronc ce céne). On peut les placer ou l'on veut et les faire tourner comme les autres éléments.
- Image d'un texte écrit sur un plan quelconque.

2 Présentation du probléme théorique



Il s'agit d'étudier la réflexion d'un mince pinceau conique émis par A se réfléchissant sur le miroir sphérique et qui pénétrera dans l'œil de l'observateur. C'est un probléme trés complexe qui a été abordé dans sa généralité par Henri Bouasse dans son livre : *Optique supérieure*, paru en 1917 aux éditions Delagrave.

Le point lumineux A, quel qu'il soit, est dans le plan du grand cercle de la sphére défini par $\mathrm{OI_1A}$. Je donne le début de cette étude théorique, qui est une adaptation des pages 108 et suivantes de son livre, sans la développer jusqu'é son terme, car comme l'indique Henri Bouasse lui-méme « *Le probléme est généralement inextricable, sauf dans le cas oé la symétrie du systéme indique quels rayons il faut regrouper aprés réflexion* ». Et bien que la fin de cette remarque s'applique au miroir sphérique, encore faudrait-il que les calculs soient facilement programmables et relativement rapides et j'avoue avoir été découragé par leur complexité. Heureusement, nous verrons que dans certaines conditions, de judicieuses approximations permettent d'obtenir une solution réaliste.

Afin d'apprécier la difficulté du probléme, on considére d'abord les limites du pinceau dans ce plan, c'est é dire sur l'arc $\widehat{\mathrm{I_1I_2}}$.

Les angles sont exagérément grands pour des raisons de clarté.

Dans le triangle $\mathrm{OI_1A}$,

$$i = \widehat{\mathrm{I_1OA}} + \widehat{\mathrm{OAI_1}}$$

Dans le triangle $\mathrm{OI_2A}$,

$$\begin{aligned} i + \mathrm{d}i &= \widehat{\mathrm{I_2OA}} + \widehat{\mathrm{OAI_2}} \\ &= \widehat{\mathrm{I_2OI_1}} + \widehat{\mathrm{I_1OA}} + \widehat{\mathrm{OAI_1}} + \widehat{\mathrm{I_1AI_2}} \\ &= \omega + \widehat{\mathrm{I_1OA}} + \widehat{\mathrm{OAI_1}} + \varepsilon \end{aligned}$$

On en déduit :

$$\mathrm{d}i = \omega + \varepsilon \tag{1}$$

Dans le triangle $\mathrm{OI_1A'}$

$$i = \pi - (\widehat{\mathrm{I_1OA'}} + \widehat{\mathrm{OA'I_1}})$$

Dans le triangle $\mathrm{OI_2A'}$

$$\begin{aligned} i + \mathrm{d}i &= \pi - (\widehat{\mathrm{I_2OA'}} + \widehat{\mathrm{OA'I_2}}) \\ &= \pi - \Big((\widehat{\mathrm{I_2OI_1}} + \widehat{\mathrm{I_1OA'}}) + (\widehat{\mathrm{OAI_1'}} - \widehat{\mathrm{I_1A'I_2}})\Big) \\ &= \pi - \Big((\omega + \widehat{\mathrm{I_1OA'}}) + (\widehat{\mathrm{OA'I_1}} - \varepsilon')\Big) \end{aligned}$$

On en déduit :

$$\mathrm{d}i = -\omega + \varepsilon' \tag{2}$$

En rapprochant (1) et (2) :

$$\omega + \varepsilon = -\omega + \varepsilon' \Longleftrightarrow 2\omega = \varepsilon' - \varepsilon \tag{3}$$

En se rappelant qu'en réalité ε, ε' et ω sont de petits angles, on établit les relations suivantes, en posant :

$$t = \mathrm{AI_1} \text{ et } t' = \mathrm{A'I_1}$$

$$\mathrm{d}s = \widehat{\mathrm{I_1I_2}} = R\omega = \frac{t\varepsilon}{\cos i} = \frac{t'\varepsilon'}{\cos i}$$

$$\varepsilon = \frac{\mathrm{R}\omega\cos i}{t} \qquad \varepsilon' = \frac{\mathrm{R}\omega\cos i}{t'}$$

Compte-tenu de la relation (3), on en déduit la formule fondamentale :

$$\frac{1}{t'} - \frac{1}{t} = \frac{2}{\mathrm{R}\cos i}$$

S'il n'y avait que le faisceau de rayons balayant depuis A l'arc $\widehat{\mathrm{I_1I_2}}$, l'image virtuelle que verrait l'œil serait $\mathrm{A'}$, qui est dans ce plan la trace de la *focale tangentielle*.

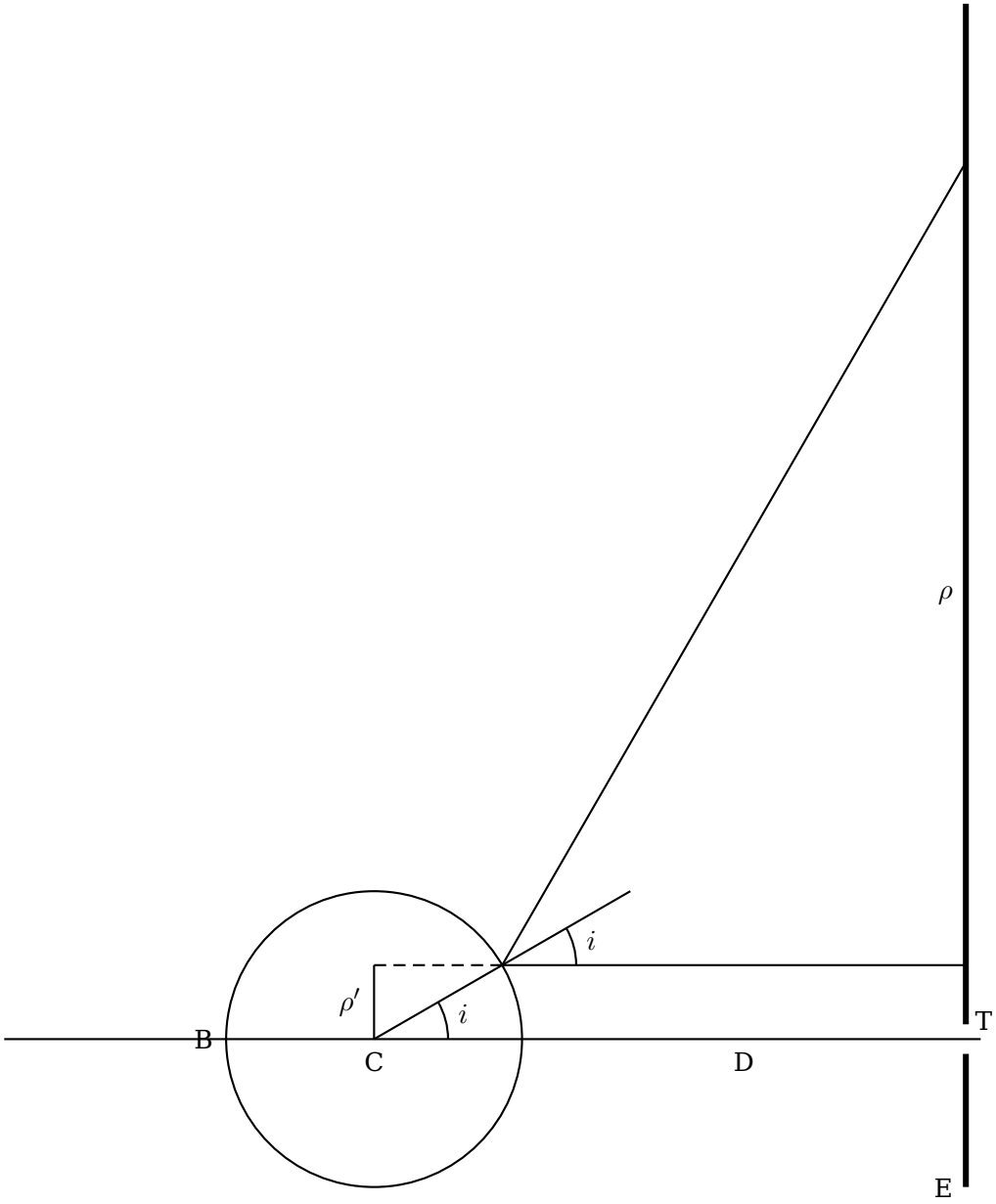
Il faut maintenant envisager toutes les autres génératrices du céne. On en prend deux de particuliéres, celles qui sont sur les bouts de l'arc de cercle de la sphére perpendiculaire au grand cercle précédent. Nous ne pouvons plus nous placer maintenant dans un plan pour faire le schéma. Ces deux génératrices permettent de déterminer la *focale radiale* $\mathrm{A''}$. Les deux focales $\mathrm{A'}$ et $\mathrm{A''}$ sont alignées avec le rayon réfléchi correspondant au rayon moyen du pinceau incident.

En conclusion, é la page 134, Henri Bouasse écrit : « *l'œil n'utilise qu'un pinceau extrémement étroit ; il s'accommode sur l'une des focales ou sur le cercle de diffusion.* ». Nous avons un peu avancé, mais quel(s) point(s) calculer?

Les milieux des focales tangentielles, radiales et du cercle de diffusion se trouvent alignés avec l'œil et le point d'incidence du rayon moyen du faisceau (celui qui aprés réflexion passe par l'œil). L'image obtenue d'un objet quelconque n'est pas plane. On pourrait contourner la difficulté en imaginant un plan fictif perpendiculaire à l'axe œil-centre de la boule et passant par une valeur moyenne des cercles de diffusion, sur lequel on calculerait les intersections des rayons. Il faut bien reconnaétre que tout cela est passablement compliqué.

Henri Bouasse propose, dans certaines conditions, une méthode approximative qui donne de bons résultats et qui a l'avantage d'étre simple. C'est l'intégralité de cette solution qu'il donne dans les pages 133, 134 et 135 de son livre, qui est développée ci-aprés.

3 Vision dans une boule de jardin



1° — Pour effectuer les expériences suivantes, on se sert d'une boule de jardin achetée dans un bazar, ou d'un petit ballon de verre de 5 cm de diamétre environ dont on argente la surface extérieure. On peut argenter extérieurement une bille d'ivoire.

Sur une trés grande feuille de carton blanc E, on trace un quadrillage en traits noirs épais. On regarde l'image de ce quadrillage par un petit trou percé au centre du carton.

La distance $\mathrm{D} = \mathrm{TB}$ sera prise d'une trentaine de centimétres.

Le quadrillage apparaét déformé *en barillet*, mais *parfaitement net*.

2° — Pour avoir une idée de la loi de déformation supposons l'œil assez loin pour que les rayons réfléchis qui y parviennent, soient quasi-paralléles. Appelons ρ' la distance d'un rayon é la droite TB.

Soit R le rayon de la boule. l'angle d'incidence est :

$$i = \arcsin(\rho' : \mathrm{R})$$

Le point oé le rayon conjugué coupe l'écran E, est é une distance ρ de TB donnée par la formule :

$$\rho = \mathrm{D} \tan 2i$$

D'oé la loi approchée :

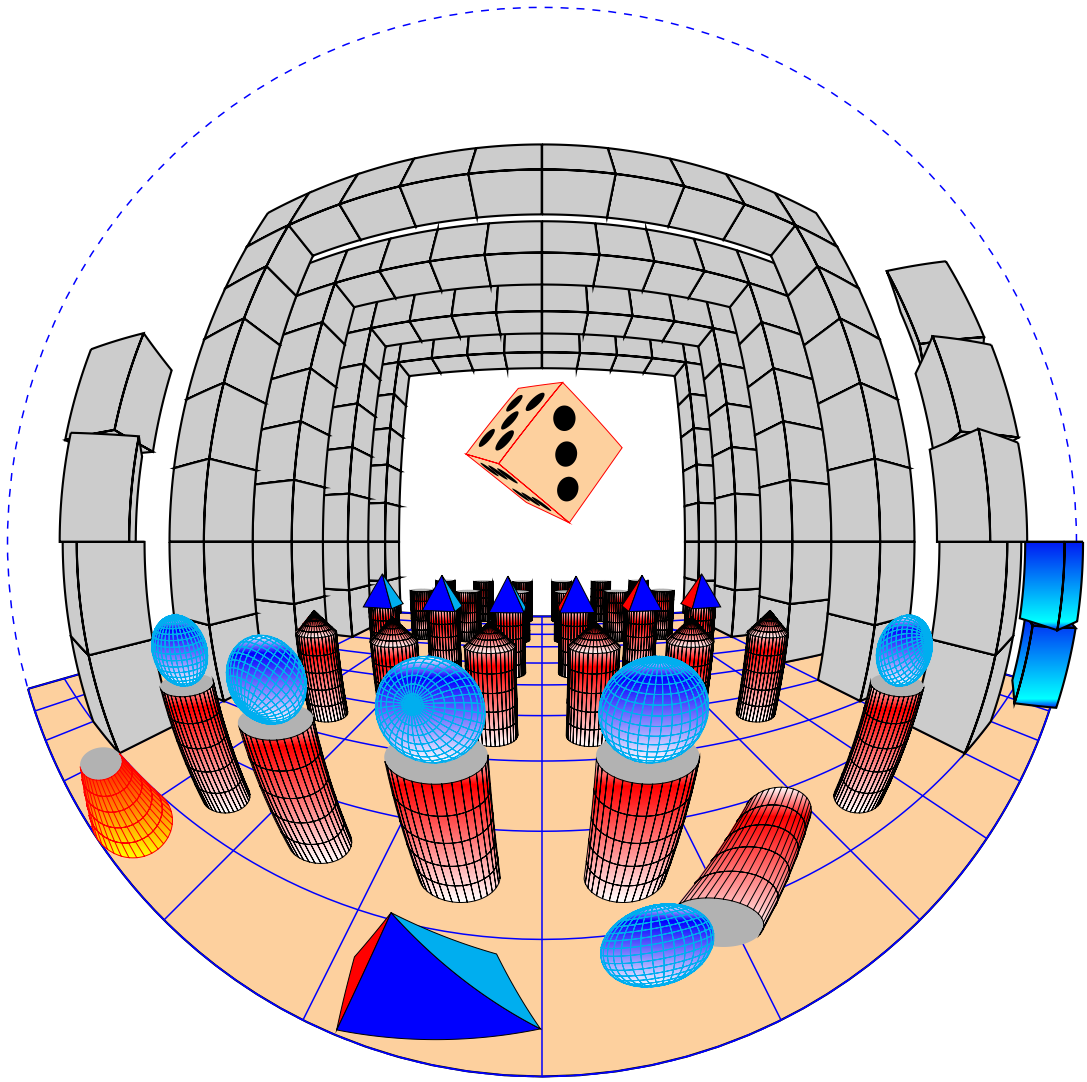
$$\arctan \frac{\rho}{\mathrm{D}} = 2 \arcsin \frac{\rho'}{\mathrm{R}}$$

ρ devient infini pour :

$$\arcsin(\rho' : \mathrm{R}) = 45^{\mathrm{o}} \qquad \rho' = 0.7\mathrm{R}$$

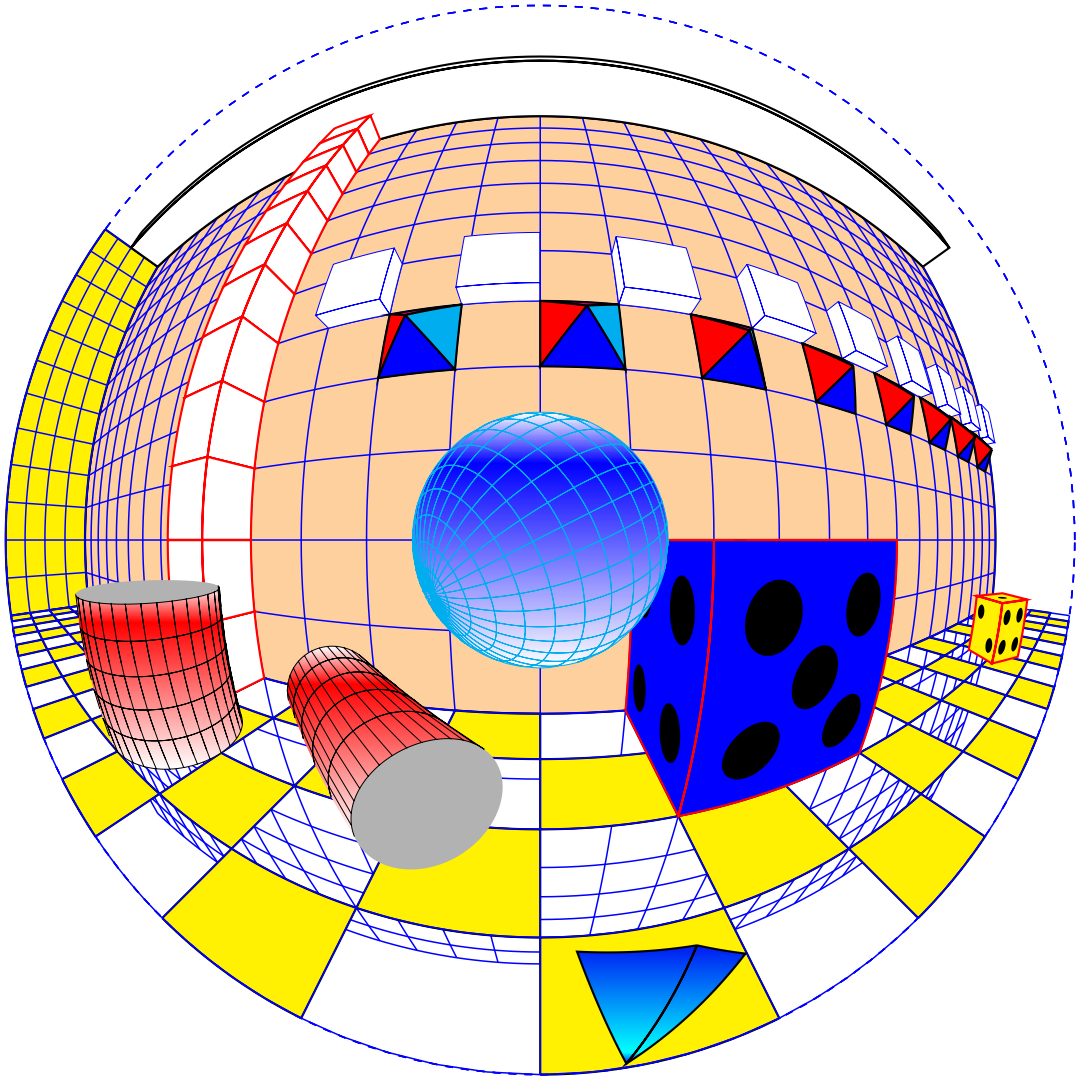
En conséquence, si grand que soit le carton, son image occupera moins des sept dixiémes du diamétre.

4 Exemples



```
\begin{pspicture}(-8,-8)(8,8)
\pstSphereFrame[normale=0 90,Xorigine=50,Yorigine=0,Zorigine=-20,fillstyle=solid,
```

```
  fillcolor=OrangePale](-70,-50)(70,50)
\pstSphereGrid[normale=0 90,linecolor=blue,linewidth=0.2mm,grille=10,
  Ymin=-50,Ymax=50,Xmax=70,Xmin=-70](50,0,-20)
{\psset{fillstyle=solid,fillstyle=solid,fillcolor=GrisClair,A=5,B=A,C=A}
\multido{\iX=95+-20}{4}{\multido{\iZ=55+-10}{6}{\psset{CX=\iX,CZ=\iZ,CY=45}\pstSphereCube}
                  \multido{\iZ=-15+10}{2}{\psset{CX=\iX,CZ=\iZ,CY=45}\pstSphereCube}
                  \multido{\iZ=55+-10}{6}{\psset{CX=\iX,CZ=\iZ,CY=-45}\pstSphereCube}
                  \multido{\iZ=-15+10}{2}{\psset{CX=\iX,CZ=\iZ,CY=-45}\pstSphereCube}
                  \multido{\iY=35+-10}{4}{\psset{CX=\iX,CZ=55,CY=\iY}\pstSphereCube}
                  \multido{\iY=-35+10}{4}{\psset{CX=\iX,CZ=55,CY=\iY}\pstSphereCube}}}
\multido{\iZ=-15+10,\iRotZ=20+45}{2}{%
  \psset{fillstyle=solid,fillstyle=gradient,fillcolor=GrisClair,A=5,B=A,C=A,
       CX=5,CY=-65,CZ=\iZ,RotZ=\iRotZ}\pstSphereCube}
\multido{\iZ=-15+10}{2}{%
  \psset{fillstyle=solid,fillstyle=solid,fillcolor=GrisClair,A=5,B=A,C=A,CX=15,CY=45,CZ=\
     iZ}\pstSphereCube
  \psset{CY=-45}\pstSphereCube}
\multido{\iZ=15+-10,\iRotZ=30+30}{2}{%
  \psset{fillstyle=solid,fillstyle=solid,fillcolor=GrisClair,A=5,B=A,C=A,
    CX=15,CY=45,CZ=\iZ,RotZ=\iRotZ}\pstSphereCube}
\multido{\iZ=25+-10,\iRotZ=30+20}{3}{%
  \psset{fillstyle=solid,fillstyle=solid,fillcolor=GrisClair,A=5,B=A,C=A,
    CX=15,CY=-45,CZ=\iZ,RotZ=\iRotZ}\pstSphereCube}
\multido{\iX=95+-20}{5}{\multido{\iY=-25+10}{6}{%
  \psset{fillstyle=gradient,gradbegin=white,gradend=red,gradmidpoint=0.2,linewidth=0.05mm,
    CY=\iY,CZ=-20,CX=\iX}
  \ifnum\iX=15
    \ifnum\iY=-15 {\psset{RotY=90,RotZ=-45,CZ=-17.5}\pstSphereCylinder{2.5}{-10}}
    \else \pstSphereCylinder{2.5}{10}\fi
  \else\pstSphereCylinder{2.5}{10}\fi}}
\multido{\iY=25+-10,\iRotY=0+60}{6}{%
  \psset{fillstyle=solid,linewidth=0.05mm,CX=55,CY=\iY,CZ=-10,A=2.5,Hpyramide=5}\
     pstSpherePyramide}
\multido{\iY=25+-10,\iRotY=0+60}{6}{%
  \psset{fillstyle=gradient,gradbegin=white,gradend=red,gradmidpoint=0.2,
    linewidth=0.05mm,CX=35,CY=\iY,CZ=-10}\pstSphereCone{2.5}{2.5}}
\multido{\iY=25+-10,\iRotY=0+60}{6}{%
\psset{fillstyle=gradient,gradbegin=white,gradend=blue,gradmidpoint=0.2,linecolor=cyan,
    linewidth=0.1mm}
\ifnum\iY=-15 {\psset{CX=7.5,CY=-5,CZ=-17.5,RotY=\iRotY} \pstMirrorSphere{2.5}}
\else {\psset{CX=15,CY=\iY,CZ=-7.5,RotY=\iRotY}\pstMirrorSphere{2.5}}\fi}
\psset{CX=2.5,CZ=-10,CY=15,fracHcone=0.5,linecolor=red,fillstyle=gradient,gradbegin=yellow
    ,
  gradend=red,gradmidpoint=0,linewidth=0.05mm}
\pstSphereCone{2.5}{5}
{\psset{RotX=0,RotZ=20,RotY=30,CX=2.5,CZ=-10,CY=2.5,fillstyle=solid,linecolor=black,A=2,
    Hpyramide=3}
\pstSpherePyramide}
{\psset{linecolor=red,CZ=5,A=2.5,B=A,C=A,RotZ=-50,RotX=20,RotY=30,CY=0,CX=20,%
  fillstyle=solid,fillcolor=OrangePale}
\pstSphereDie}
\end{pspicture}
```



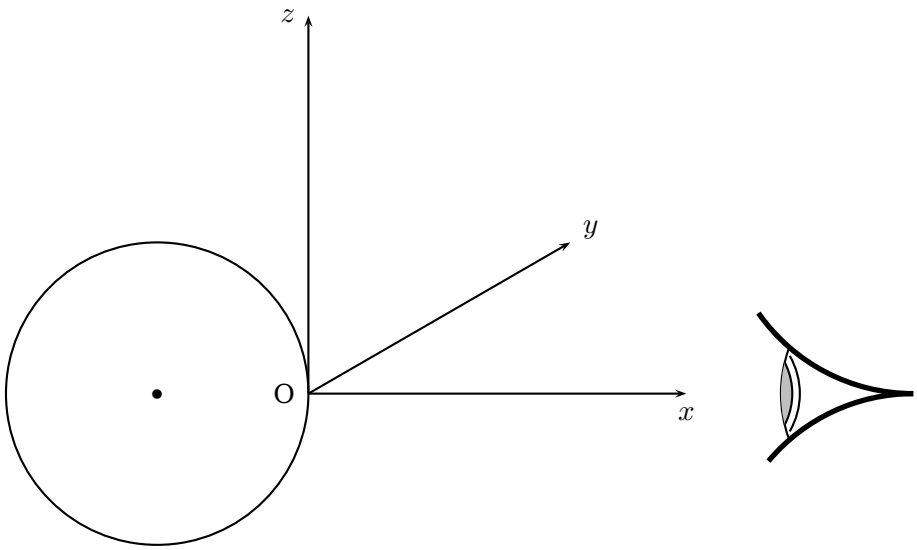
```
\begin{pspicture}(-8,-8)(8,8)
%\psset{scale=2}
\pstSphereFrame[Xorigine=20,Yorigine=0,Zorigine=0,fillstyle=solid,fillcolor=OrangePale
    ](-70,-50)(70,10)
\pstSphereGrid[linecolor=blue,linewidth=0.2mm,grille=5,Ymin=-50,Ymax=50,Xmax=70,Xmin
    =-70](20,0,0)%
%
\pstSphereFrame[normale=90 0,Xorigine=10,Yorigine=70,Zorigine=0,fillstyle=solid,fillcolor=
    yellow](-10,-50)(10,10)
\multido{\nX=17.5+-10.0}{2}{%
  \multido{\nY=-67.5+10.0}{14}{%
    \pstSphereFrame[normale=0 90,Xorigine=\nX,Yorigine=\nY,Zorigine=-10,fillstyle=solid,
     fillcolor=yellow](-2.5,-2.5)(2.5,2.5)}}
\multido{\nX=12.5+-10.0}{2}{%
  \multido{\nY=-62.5+10.0}{14}{%
    \pstSphereFrame[normale=0 90,Xorigine=\nX,Yorigine=\nY,Zorigine=-10,fillstyle=solid,
     fillcolor=yellow](-2.5,-2.5)(2.5,2.5)}}
\pstSphereGrid[normale=90 0,linecolor=blue,linewidth=0.2mm,grille=5,
```

```
  Xmin=-10,Xmax=10,Ymax=10,Ymin=-50](10,70,0)%
\pstSphereGrid[normale=0 90,linecolor=blue,linewidth=0.2mm,grille=5,
  Xmin=-70,Xmax=70,Ymax=10,Ymin=-10](10,0,-10)%
{\psset{linecolor=black,CZ=52,A=5,B=70,C=2,CY=0,CX=15,fillstyle=solid}\pstSphereCube}
{\psset{linecolor=red,CX=15,A=5,B=A,C=A,CY=-10,CZ=-5,RotZ=0,fillstyle=solid,fillcolor=blue
    }\pstSphereDie}%
\multido{\nZ=47.5+-5.0}{10}{%
  \psset{linecolor=red,CY=22.5,A=2.5,B=A,C=A,CZ=\nZ,CX=17.5,fillstyle=solid}\pstSphereCube
     }
\multido{\nZ=-7.5+5.0}{2}{%
  \psset{linecolor=red,CY=22.5,A=2.5,B=A,C=A,CZ=\nZ,CX=17.5,fillstyle=solid}\pstSphereCube
     }
{\psset{fillstyle=gradient,gradbegin=white,gradend=blue,gradmidpoint=0.2,linecolor=cyan,
    linewidth=0.1mm,%
  CX=15,RotX=45,RotY=60}
\pstMirrorSphere{5}}
{\psset{fillstyle=gradient,gradbegin=white,gradend=red,gradmidpoint=0.2,linewidth=0.1mm,CX
    =10,CY=20,CZ=-10}
\pstSphereCylinder{5}{7.5}}
{\psset{fillstyle=gradient,gradbegin=white,gradend=red,gradmidpoint=0.2,linewidth=0.1mm,%
   CX=15,CY=10,CZ=-7.5,RotY=90,RotX=-45}\pstSphereCylinder{2.5}{-10}}
{\psset{linecolor=red,CX=10,A=2.5,B=A,C=A,CY=-40,CZ=-7.5,RotZ=60,fillstyle=solid,fillcolor
    =yellow}\pstSphereDie}%
{\psset{CZ=-10,CY=-2.5,CX=2.5,RotZ=-20,linecolor=black,fillstyle=gradient,Rtetraedre=2.5}\
    pstSphereTetraedre}
\multido{\nY=-67.5+10.0}{9}{%
  {\psset{linecolor=blue,CZ=17.5,A=1,B=2.5,C=2.5,CY=\nY,CX=19,fillstyle=solid,linewidth
     =0.1mm}\pstSphereCube}}
\multido{\nY=-62.5+10.0}{8}{\psset{RotX=0,RotZ=0,RotY=-90,CX=20,CZ=12.5,CY=\nY,
  fillstyle=solid,linecolor=black,A=2.5,Hpyramide=3}\pstSpherePyramide}
\end{pspicture}
```

5 Utilisation des briques

5.1 Le repére

l'origine est placée contre la boule, sur l'horizontale joignant le centre C de la boule é l'œil de l'observateur. Oz est vertical, Oxy forme le plan horizontal. Tous les objets seront donc placés au-delé de O avec $x > 0$.



5.2 Placer un quadrillage

Le plan du quadrillage est défini par une origine (Xorigine,Yorigine,Zorigine), le vecteur unitaire normal au plan donné par sa longitude et sa latitude (θ, ϕ) lesquelles sont paramétrées, avec des valeurs en degrés avec :

- normale =θ ϕ

Un vecteur unitaire quelconque a pour coordonnées, en fonction de θ et ϕ :

$$\vec{u} = \begin{pmatrix} \cos\theta\cos\phi \\ \sin\theta\cos\phi \\ \sin\phi \end{pmatrix}$$

$\vec{i}, \vec{j}, \vec{k}$ sont les vecteurs unitaires dirigés respectivement suivant Ox, Oy et Oz.

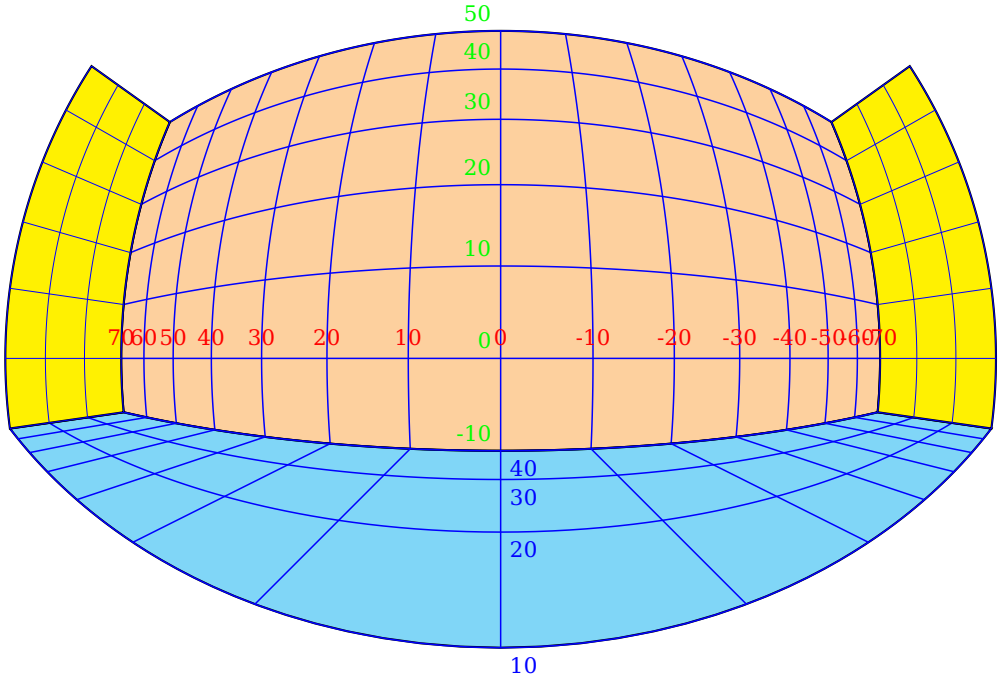
Le vecteur unitaire normal é un plan de front, c'est-é-dire paralléle é Oyz, est le vecteur $\vec{i}$ ou $-\vec{i}$, $\vec{u}(\theta = 0$ ou $\pi, \phi = 0)$.

Un plan horizontal est paralléle é Oxy, un vecteur normal est soit $\vec{k}$, soit $-\vec{k}$, il est défini par $\vec{u}(\theta =$ valeur indifférente$(0), \phi = \pi/2$ ou $-\pi/2)$.

Un plan vertical paralléle é Oxz a pour normale $\vec{j}$ ou $-\vec{j}$, donc $\theta = \pi/2$ ou $-\pi/2$ et $\phi = 0$.

Une remarque importante : le vecteur unitaire choisi normal au plan $\vec{u}$, détermine l'orientation des vecteurs $\vec{I}$ et $\vec{J}$ du plan, avec les régles habituelles d'orientation : le triédre $(\vec{I}, \vec{J}, \vec{u})$ est direct, **mais le vecteur $\vec{I}$ a étépris dans le plan de référence Oxy.**

Pour que le quadrillage soit parfaitement déterminé, on fixe les valeurs de deux sommets opposé avec : Ymin=-50,Ymax=10,Xmax=70,Xmin=-70. Ainsi que le pas de la grille avec grille=10, par exemple.




```
\begin{pspicture}(-5,-4)(5,5)
  \decor% les graduations, see preamble
  \multido{\iY=-70+10}{15}{\pstSpherePoint(40,\iY,0){A}\uput[90](A){\footnotesize\red\iY}}
  \multido{\iX=10+10}{4}{\pstSpherePoint(\iX,0,-10){A} \uput[-45](A){\footnotesize\blue\iX
      }}
  \multido{\iZ=-10+10}{7}{\pstSpherePoint(40,0,\iZ){A} \uput[135](A){\footnotesize\green\
      iZ}}
\end{pspicture}
```

En bleu graduations de l'axe Ox, en rouge celles de Ox et en vert celles de Oz.

Le quadrillage du plan de front est défini par :

```
\pstSphereGrid[normale=0 0,% default
  linecolor=blue,linewidth=0.2mm,grille=10,Ymin=-50,Ymax=10,Xmax=70,Xmin=-70](40,0,0)%
```

Le quadrillage du plan horizontal par :

```
\pstSphereGrid[normale=0 90,linecolor=blue,linewidth=0.2mm,grille=10,
   Ymin=-30,Ymax=0,Xmax=70,Xmin=-70](40,0,-10)%
```

Les quadrillages des plans latéraux par :

```
\pstSphereGrid[normale=90 0,linecolor=blue,linewidth=0.05mm,grille=10,
   Ymin=-60,Ymax=0,Xmin=-30,Xmax=0](40,70,-10)
\pstSphereGrid[normale=90 0,linecolor=blue,linewidth=0.05mm,grille=10,%
   Ymin=-60,Ymax=0,Xmin=-30,Xmax=0](40,-70,-10)
```

On peut se rendre compte de ce qui se produit en changeant l'orientation du vecteur normal. J'inverse l'orientation de la normale pour les quatre plans.

On obtient la méme chose que précédemment en faisant :

```
\begin{pspicture}(-5,-4)(5,4)
% plan de front
\psset{Sphere=false}
{\psset{normale=180 0,Xorigine=40,Yorigine=0,Zorigine=0}
```

```
\pstSphereFrame[fillstyle=solid,fillcolor=OrangePale](-70,-50)(70,10)
\pstSphereGrid[linecolor=blue,linewidth=0.2mm,grille=10,
              Ymin=-50,Ymax=10,Xmax=70,Xmin=-70](40,0,0)}
% plan horizontal
{\psset{normale=0 -90,Xorigine=40,Yorigine=0,Zorigine=-10}
\pstSphereFrame[fillstyle=solid,fillcolor=cyan](70,30)(-70,0)
\pstSphereGrid[linecolor=blue,linewidth=0.2mm,grille=10,
  Ymin=0,Ymax=30,Xmax=70,Xmin=-70](40,0,-10)}%
% plan vertical de gauche
{\psset{normale=-90 0,Xorigine=40,Yorigine=70,Zorigine=-10}
\pstSphereFrame[fillstyle=solid,fillcolor=yellow](0,0)(30,-60)
\pstSphereGrid[linecolor=blue,linewidth=0.05mm,grille=10,
  Ymin=-60,Ymax=0,Xmin=0,Xmax=30](40,70,-10)}
% plan vertical a droite
{\psset{normale=-90 0,Xorigine=40,Yorigine=-70,Zorigine=-10}
\pstSphereFrame[fillstyle=solid,fillcolor=yellow](0,0)(30,-60)
\pstSphereGrid[linecolor=blue,linewidth=0.05mm,grille=10,Ymin=-60,
  Ymax=0,Xmin=0,Xmax=30](40,-70,-10)}
\end{pspicture}
```

5.3 Dessiner un rectangle

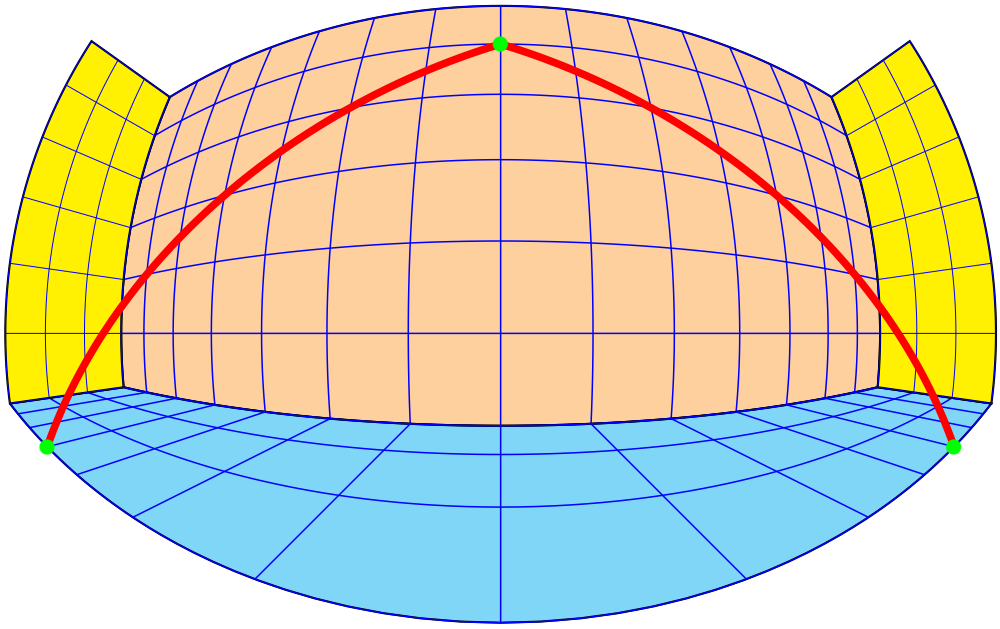
Dans l'exemple précédent, j'ai dessiné les rectangles correspondants aux quadrillages. La commande est identique é celle du quadrillage avec les mémes paramétres pour l'origine et le vecteur normal au plan. On donne les coordonnées de deux sommets opposés comme dans la commande `\psframe` de PSTricks, dans le plan.

5.4 Tracer une ligne, marquer un point

Un segment de droite est déterminé par les coordonnées de ses extrémités. Un point par ses coordonnées. On peut tracer une ligne brisée en plaéant les points é la suite, comme dans la commande de PStricks `\psline`.

```
\pstSpherePoint(40,0,40){A}
\pstSpherePoint(10,-40,-10){B}
\pstSpherePoint(10,40,-10){C}
\psdots[dotsize=1mm,linecolor=green](A)(B)(C)
\pstSphereLine[linecolor=red](10,40,-10)(40,0,40)(10,-40,-10)
```

Vous remarquerez que les coordonnées de y sont inversées. Je rappelle que le plan couleur cyan est é la cote $z = -10$.



5.5 Dessiner un polygone

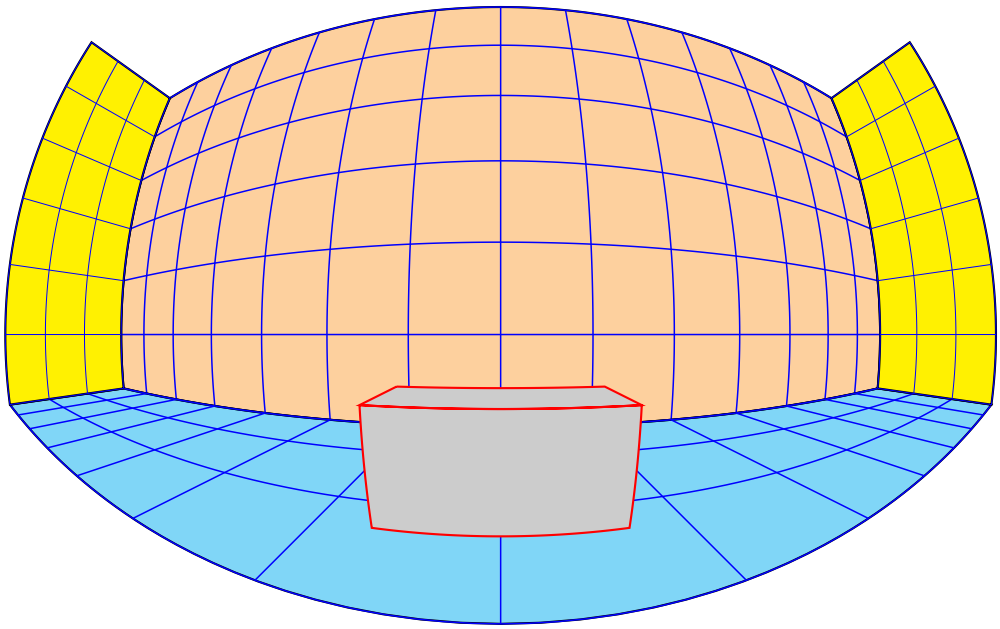
Commande identique é celle de PStricks.

```
\pstSpherePolygon[options](x0,y0,z0)(x1,y1,z1)...(xn,yn,zn)
```

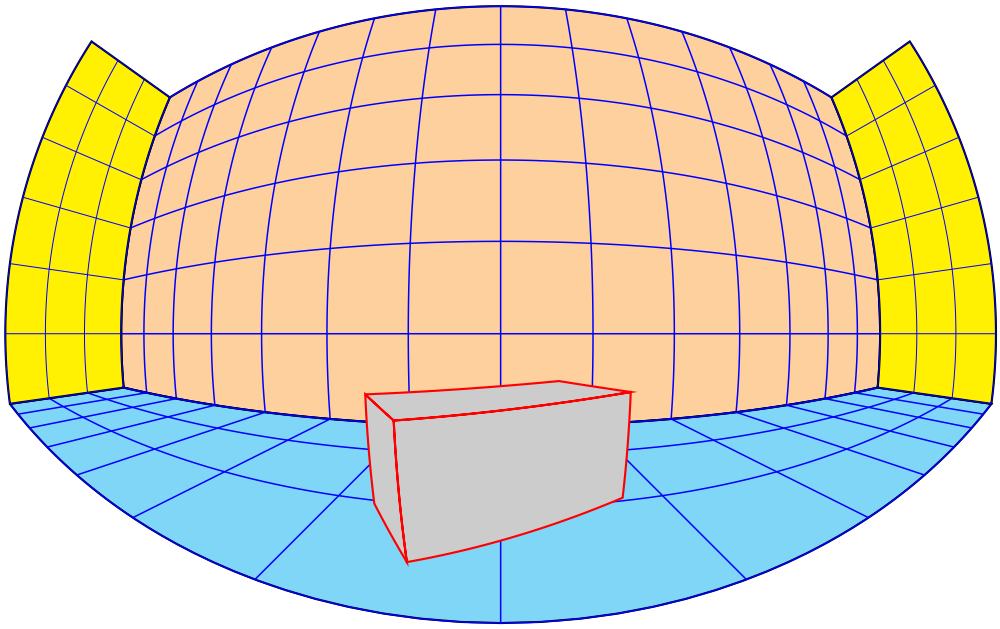
5.6 Parallélépipéde

Le parallélépipéde est défini par ses trois dimensions A,B,C. On peut le déplacer en fixant les coordonnées de son centre avec : CX,CY,CZ et le faire tourner avec autour des axes avec : RotX,RotY,RotZ dont les valeurs sont par défauts, positionnées é 0.

```
\psset{A=2.5,B=5,C=A,fillstyle=solid,fillcolor=GrisClair,linecolor=red}
\psset{CX=10,CZ=-5,CY=0}
\pstSphereCube
```



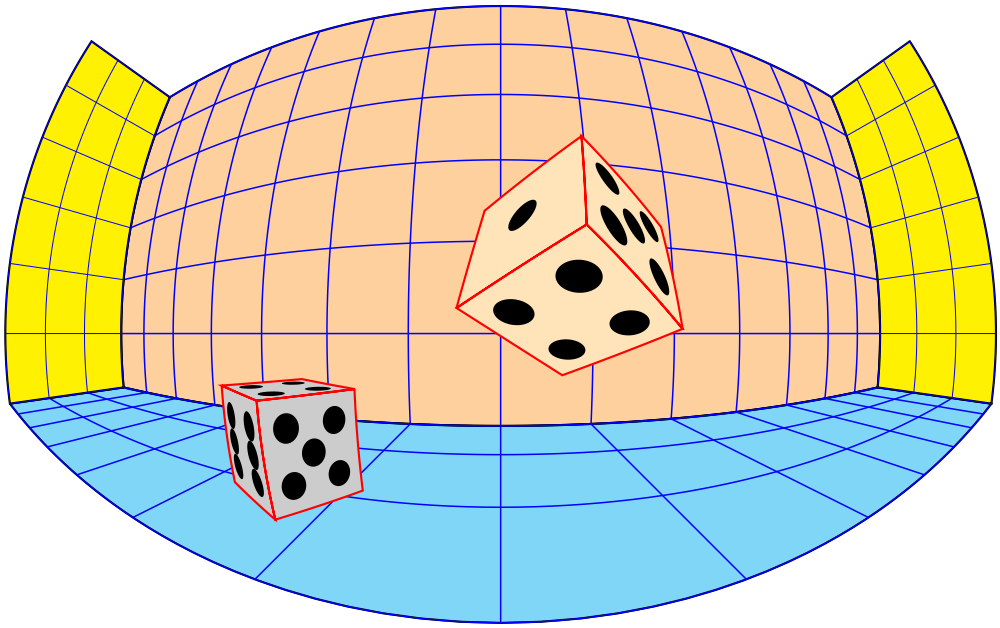
On le fait tourner autour de Oz avec `RotZ=30` etc.



```
\psset{CX=15,CZ=-5,CY=0,RotZ=30}
```

5.7 Le dé

Le dé posséde une commande propre \pstSphereDie, qui en interne utilise celle du \pstSphereCube en affichant les points. Voici quelques dés.



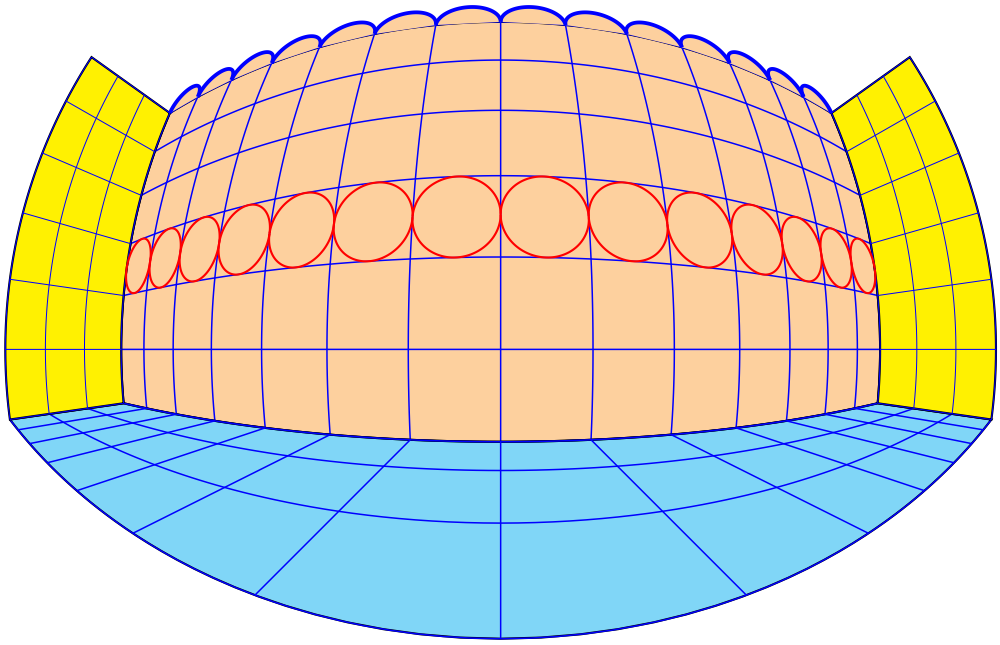
```
\psset{A=2.5,B=A,C=A,fillstyle=solid,fillcolor=GrisClair,linecolor=red}
\psset{CX=15,CZ=-5,CY=10,RotZ=60,RotX=90}
\pstSphereDie
\psset{A=5,B=A,C=A,fillstyle=solid,fillcolor=OrangeTresPale,linecolor=red}
\psset{CX=25,CZ=5,CY=-5,RotZ=45,RotY=20,RotX=-60}
\pstSphereDie
```

5.8 Tracer un cercle ou un arc de cercle

Le cercle et l'arc de cercle seront définis par le plan auquel ils appartiennent. Comme précédemment le plan sera défini par une origine et le vecteur unitaire normal é ce plan. Ces données étant posées, il n'y plus que le rayon du cercle é fixer et les deux angles limites s'il s'agit de l'arc de cercle.

```
\pstSphereCircle [Options] {radius}
\pstSphereArc [Options] {radius}{angle_min}{angle_max}
```

\pstSphereArc : respectez l'ordre des angles limites sinon l'interpréteur PostScript signalera une erreur.

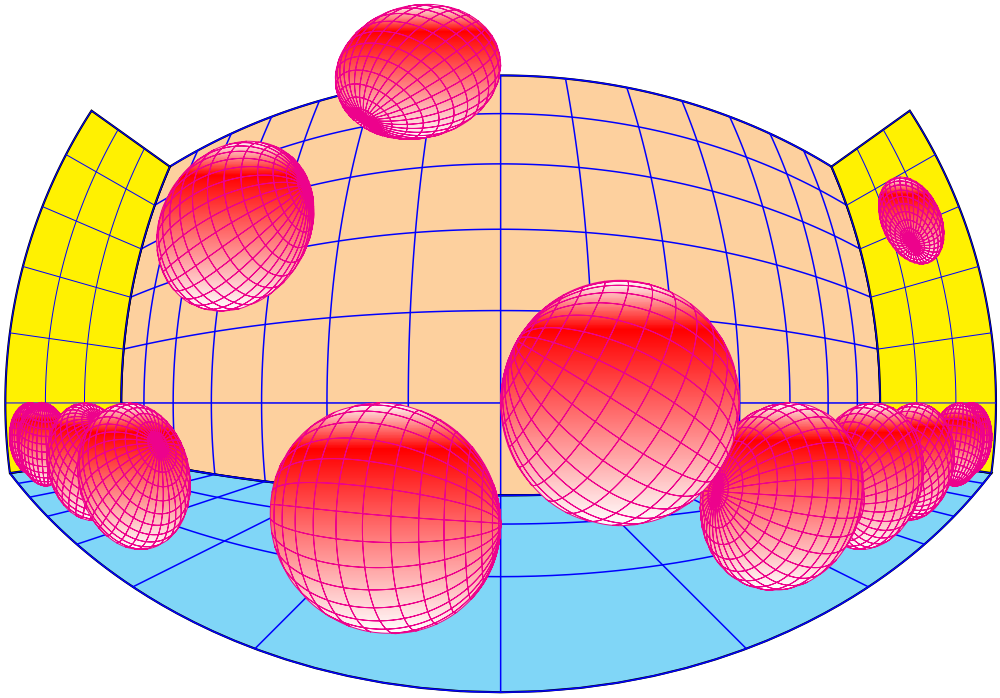


Des cercles et des arcs de cercle sur le plan de front :

```
\psset{Xorigine=40}
\multido{\iYorigine=-65+10}{14}{%
  \psset{Yorigine=\iYorigine,Zorigine=15}\pstSphereCircle[linecolor=red]{5}}
\multido{\iYorigine=-65+10}{14}{\psset{Yorigine=\iYorigine,Zorigine=50}
  \pstSphereArc[linecolor=blue,linewidth=0.5mm,fillstyle=solid,fillcolor=OrangePale
     ]{5}{180}{360}}
```

5.9 La sphére

Elle est définie par son rayon, son centre se place avec les paramétres CX, CY, CZ, on la fait tourner autour de son centre avec RotX, RotY, RotZ.

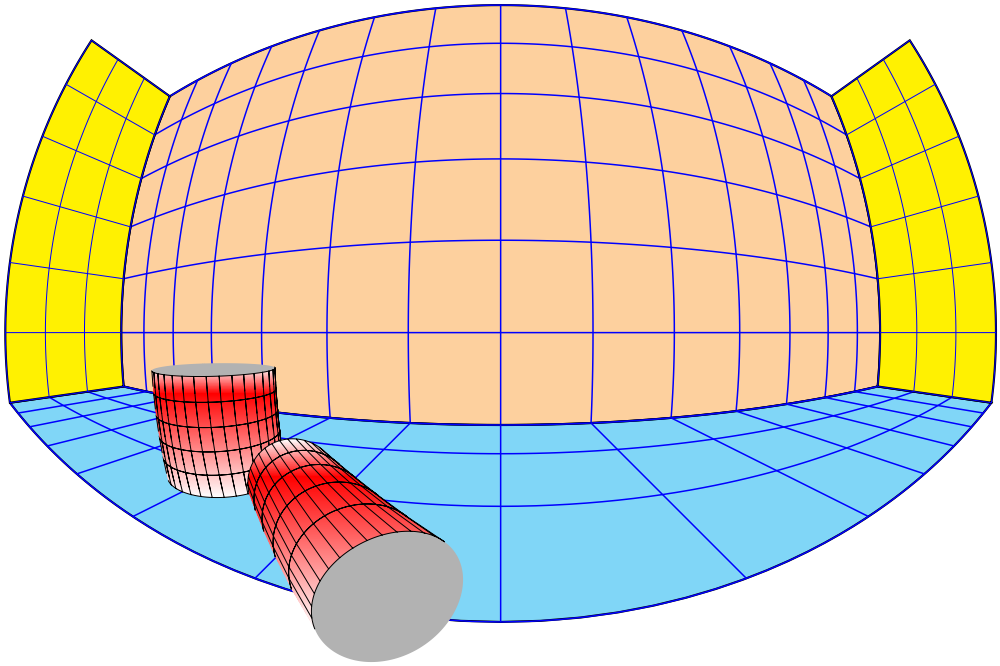


```
\multido{\iCY=-65+10,\iRotX=0+20}{7}{%
  \psset{CY=\iCY,CZ=-5,CX=15,RotX=\iRotX,fillstyle=gradient,gradbegin=white,gradend=red,%
      gradmidpoint=0.2,linecolor=magenta,linewidth=0.05mm}\pstMirrorSphere{5}}
```

5.10 Dessiner un cylindre

Le cylindre sera donné par le rayon de sa base et sa hauteur. Le centre de la base sera positionné avec CX, CY, CZ, on le fait tourner autour autour des axes avec RotX, RotY, RotZ.

```
\pstSphereCylinder [Options] {radius}{hauteur}
```



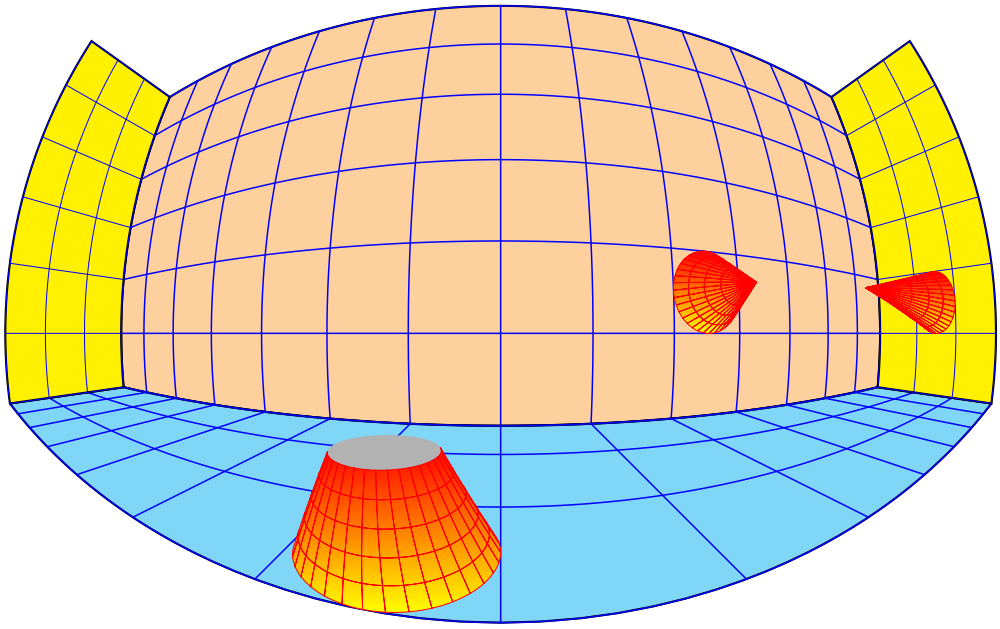
```
\psset{fillstyle=gradient,gradbegin=white,gradend=red,%
     gradmidpoint=0.2,linewidth=0.1mm}
{\psset{CX=20,CY=20,CZ=-10}
\pstSphereCylinder{5}{7.5}}
{\psset{CX=15,CY=10,CZ=-7.5,RotY=90,RotX=-45}
\pstSphereCylinder{2.5}{-10}}
```

5.11 Dessiner un céne

Le céne se définit comme le cylindre, rayon de la base, hauteur et en option la fraction de hauteur de céne si l'on souhaite avoir un tronc de céne.

`\pstSphereCone[Options]{radius}{hauteur}`

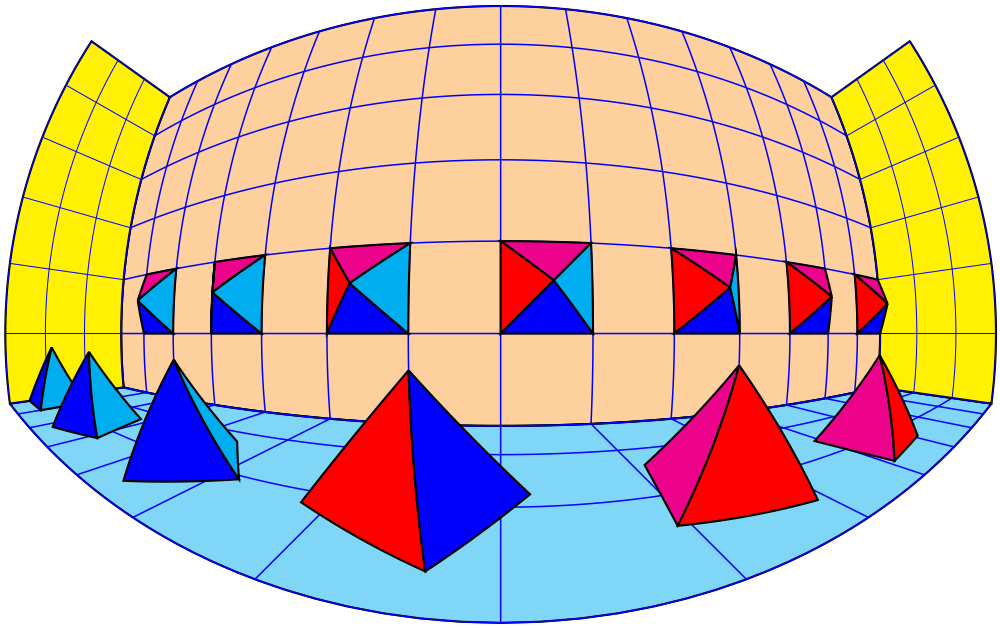
`fracHcone` est obligatoirement un nombre compris entre 0 et 1.



```
{\psset{CX=15,CZ=-10,CY=5,fracHcone=0.5}\pstSphereCone{5}{10}}
{\psset{CX=40,CZ=5,CY=-25,RotY=-90} \pstSphereCone{5}{10}}
{\psset{CX=25,CZ=5,CY=-70,RotX=-90} \pstSphereCone{5}{30}}
```

5.12 Tracer une pyramide

La pyramide é base carrée est donnée par le demi-cété (A)de sa base et sa hauteur (Hpyramide), que l'on peut faire tourner et placer oé on veut, comme les objets précédents.



```
\psset{fillstyle=solid,linecolor=black,A=5}
\multido{\nY=-65+20.0}{4}{%
  \psset{RotX=0,RotZ=0,RotY=-90,CX=40,CZ=5,CY=\nY,Hpyramide=5}\pstSpherePyramide}
\multido{\nY=65+-20.0,\iRotZ=0+20}{6}{%
  \psset{RotX=0,RotZ=\iRotZ,RotY=0,CX=20,CZ=-10,CY=\nY,Hpyramide=8}\pstSpherePyramide}
```

6 Image d'un texte

C'est la commande \pstSphereText [Options] [*x,y,z*]{*texte*} qui permet d'afficher le texte souhaité avec les paramétres suivants :

- Le plan qui contient le texte est défini par :
  - sa normale normale=θ ϕ ;
  - et son origine (x,y,z).
- La taille des caractéres est fixée par fontscale=1 : 1 cm par défaut.
- Le type de fonte par fontname=Times-Roman par défaut.
- Le booléen isolatin qui est positionné é true par défaut, permet l'écriture des caractéres accentués. Si l'on souhaite utiliser la police Symbol, permettant d'écrire les caractéres grecs, il faut positionner ce booléen é false.


```
\begin{pspicture}(-8,-10)(8,10)
%\psset{Rayon=8}
\newpsstyle{GradWhiteYellow}{fillstyle=gradient,%
   gradbegin=yellow,gradend=yellow!20,linecolor=yellow!50,GradientCircle=true,gradmidpoint
       =0,GradientPos={(1,1)}}%
\pscircle[style=GradWhiteYellow]{7.07}
\pstSphereGrid[linecolor=red,grille=10,Ymin=-50,Ymax=50,Xmax=80,Xmin=-80,normale=0
    0](20,0,0)
```

```
\pstSphereGrid[linecolor=blue,grille=10,Ymin=-40,Ymax=-20,Xmax=80,Xmin=-80,normale=0
    90](40,0,-10)%
\pstSphereText[fillstyle=solid,fillcolor=red,normale=0 0,fontscale=30,
  fontname=Bookman-Demi,yO=0](20,0,10){Tangente}
\pstSphereText[fillstyle=solid,fillcolor=red,normale=0 0,fontscale=30,PSfont=Bookman-Demi,
    yO=0](20,0,10){Tangente}
\pstSphereText[fillstyle=solid,fillcolor=blue,normale=0 90,fontscale=10,
  fontname=Helvetica,yO=2.5](10,0,-10){l'aventure math\'ematique}%
\end{pspicture}
```

6.1 Les images

L'image doit être au format eps et ell doit être « *normalisée* » pour que le calcul de son image dans le miroir puisse s'effectuer. La procédure de calcul est identique à celle utilisée dans le package pst-anamorphosis. Cette image doit être dans un plan défini comme pour le texte.

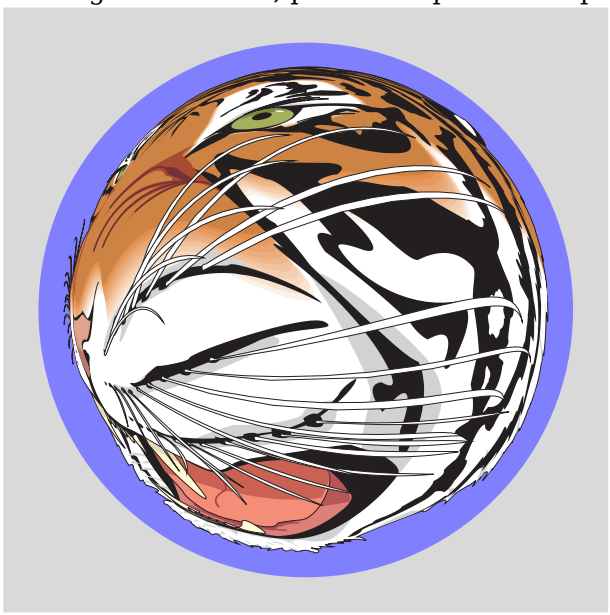
```
\pstSphereImage [Options] (x,y,z){filename.eps}
\parametricplotSphere [Options] {x0}{x1} [PS code] {function}
```

```
\begin{pspicture}(-7,-7)(7,7)
```

```
{\psset{normale=0 0,Xorigine=15,Yorigine=0,Zorigine=0}
\pstSphereFrame[fillstyle=solid,fillcolor=black](-70,-50)(70,50)
\pstSphereGrid[linecolor=white,linewidth=0.2mm,grille=10,Ymin=-50,Ymax=50,Xmax=70,Xmin
    =-70](15,0,0)}%
\pstSphereImage[normale=0 0,scale=0.75](15,0,0){images/donaldNB.eps}
\pstSphereText[fillstyle=solid,fillcolor=white,linecolor=white,fontscale=17,fontname=
    Symbol,
           normale=0 0,isolatin=false](15,0,25){tiger.eps}
\parametricplotSphere[plotpoints=3600,linecolor=white,linewidth=0.5mm,normale=0 0,Xorigine
    =15,%
            Yorigine=30,Zorigine=10]{0}{360}{%
  /Radius 10 t 4 mul cos mul def
   Radius t cos mul 1 add Radius t sin mul }
\parametricplotSphere[plotpoints=3600,fillstyle=solid,fillcolor=green!50,linecolor=white,
            Xorigine=15,Yorigine=-30,Zorigine=10]{0}{360}{%
  /Radius 10 t 2 mul sin t 6 mul sin 4 div add mul def
  Radius t cos mul Radius t sin mul }
\parametricplotSphere[plotpoints=3600,linecolor=white,linewidth=0.1,
            Xorigine=15,Yorigine=0,Zorigine=-10,algebraic]{-70}{70}{
  t|10*sin(t/4) }
\end{pspicture}
```

7 Une remarque importante

Si le calcul des faces cachées est pris en compte pour chaque objet pris individuellement, il en est pas de même pour l'ensemble du panorama. Il convient donc de placer en premier les objets les plus éloignés de boule, puis ceux qui sont de plus en plus proche de la boule.



```
\begin{pspicture}(-5,-5)(5,5)
\psframe*(-5,-5)(5,5)
\pscircle*[linecolor=blue!50]{!5 2 sqrt div}
\psset{Rayon=5}
\pstSphereImage[normale=0 0](5,0,0){images/tiger-c.eps}
%\pstSphereGrid[linecolor=gray,linewidth=0.01,grille=5,Ymin=-100,Ymax=100,Xmax=100,Xmin
    =-100](5,0,0)%
```

```
\end{pspicture}
```

8 Conclusion

Nous pouvons maintenant nous atteler é la téche de construire des décors en 3D vus dans une boule qui soient intéressants et artistiques, un prolongement vers une animation est toujours possible. J'ai pensé inclure dans ce package, é l'aide d'une option, la 3D conventionnelle : il suffit de changer les formules relatives é la boule et d'y rajouter les paramétres relatifs au point de vue, mais finalement cela aurait trop alourdi cette étude – vous pouvez regardez la commande `\pnodeTroisD` et les nombreuses applications développées dans ma page personnelle.

Une remarque importante : si le calcul des faces cachées est pris en compte pour chaque objet pris individuellement, il en est pas de méme pour l'ensemble du panorama. Il convient donc de placer en premier les objets les plus éloignés de boule, puis ceux qui sont de plus en plus proche de la boule.

9 List of all optional arguments for `pst-mirror`

Key	Type	Default
visibility	boolean	true
isolatin	boolean	false
Sphere	boolean	true
Xmax	ordinary	50
Xmin	ordinary	-50
Ymax	ordinary	50
Ymin	ordinary	-50
pas	ordinary	1
scale	ordinary	1
distance	ordinary	30
Rayon	ordinary	10
grille	ordinary	10
normale	ordinary	0 0
Xorigine	ordinary	\psk@Sphere@Distance
Yorigine	ordinary	0
Zorigine	ordinary	0
RotX	ordinary	0
RotY	ordinary	0
RotZ	ordinary	0
A	ordinary	10
B	ordinary	10
C	ordinary	10
CX	ordinary	0
CY	ordinary	0
CZ	ordinary	0
Rtetraedre	ordinary	5
ColorFaceA	ordinary	magenta
ColorFaceB	ordinary	red

Continued on next page

Continued from previous page

Key	Type	Default
ColorFaceC	ordinary	blue
ColorFaceD	ordinary	cyan
ColorFaceE	ordinary	yellow
fracHcone	ordinary	1
Hpyramide	ordinary	5
fontname	ordinary	Times-Roman
fontscale	ordinary	1
xlabelsep	ordinary	-0.5
ylabelsep	ordinary	-0.5
xO	ordinary	0
yO	ordinary	0
translation	ordinary	0 0 0
path	ordinary	[none]

References

[1] Victor Eijkhout. *TEX by Topic – A TEXnician Reference*. 1st ed. Heidelberg/Berlin: DANTE – lehmanns media, 2014.

[2] Denis Girou. “Présentation de PSTricks”. In: *Cahier GUTenberg* 16 (Apr. 1994), pp. 21–70.

[3] Michel Goosens et al. *The LATEX Graphics Companion*. second. Boston, Mass.: Addison-Wesley Publishing Company, 2007.

[4] Nikolai G. Kollock. *PostScript richtig eingesetzt: vom Konzept zum praktischen Einsatz*. Vaterstetten: IWT, 1989.

[5] Herbert Voß. *PSTricks – Grafik für TEX und LATEX*. 6th ed. Heidelberg/Berlin: DANTE – lehmanns media, 2010.

[6] Herbert Voß. *PSTricks – Graphics for LATEX*. 1. Cambridge/UK: UIT, 2011.

[7] Herbert Voß. *LATEX Quick Reference*. 1. Cambridge/UK: UIT, 2011.

[8] Herbert Voß. *LATEX Referenz*. 2. Heidelberg/Hamburg: DANTE – lehmanns media, 2010.

[9] Timothy Van Zandt. *multido.tex - a loop macro, that supports fixed-point addition*. CTAN:/macros/generic/multido.tex, 1997.

[10] Timothy Van Zandt and Denis Girou. “Inside PSTricks”. In: *TUGboat* 15 (Sept. 1994), pp. 239–246.

Index