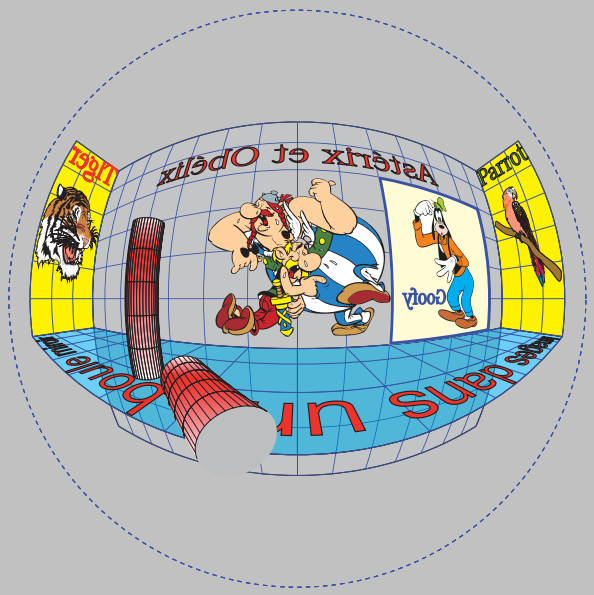
pst-mirror

Images dans un miroir sphérique : liste des macros et des options



Contributeurs

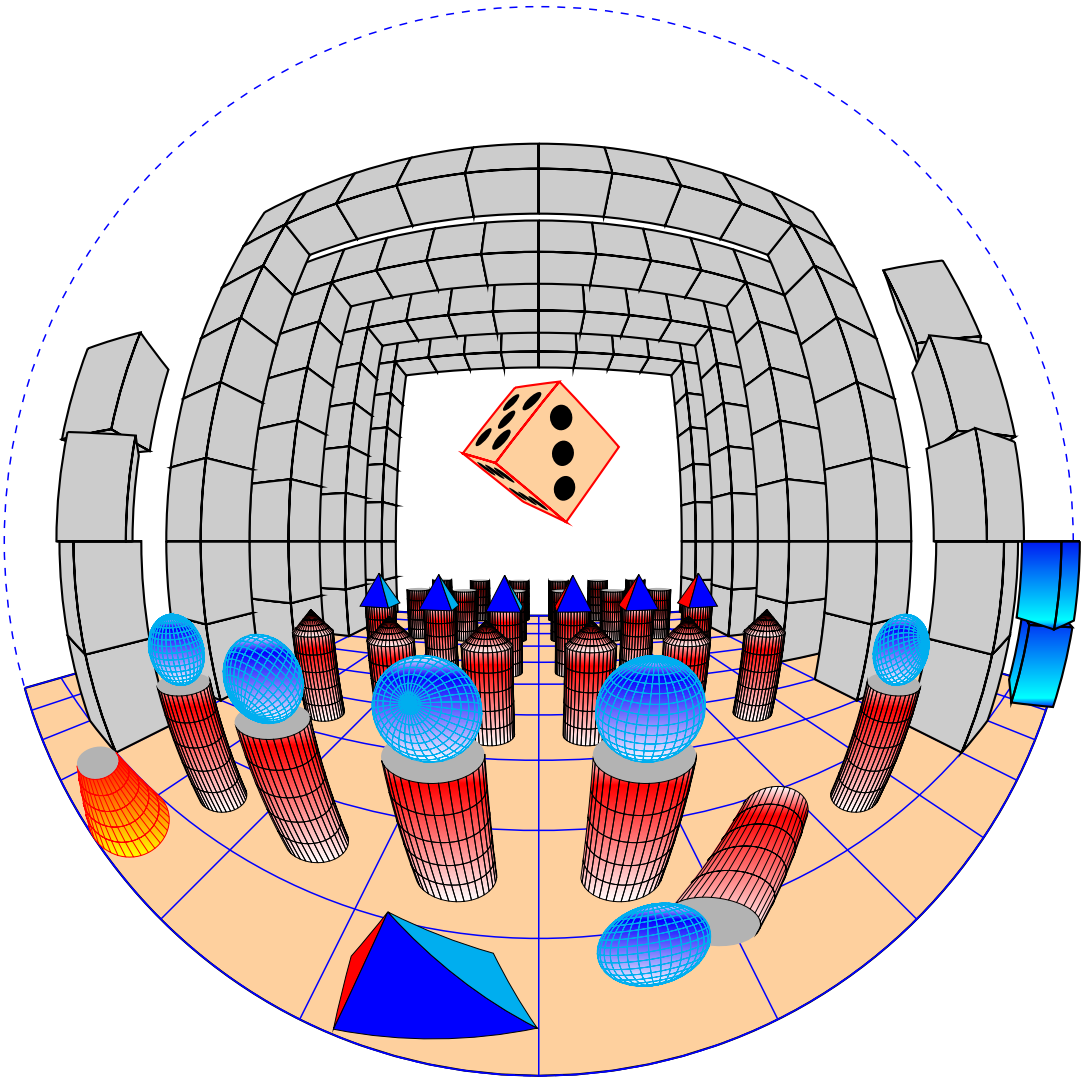
Jürgen Gilg
Manuel Luque
Jean-Michel Sarlat

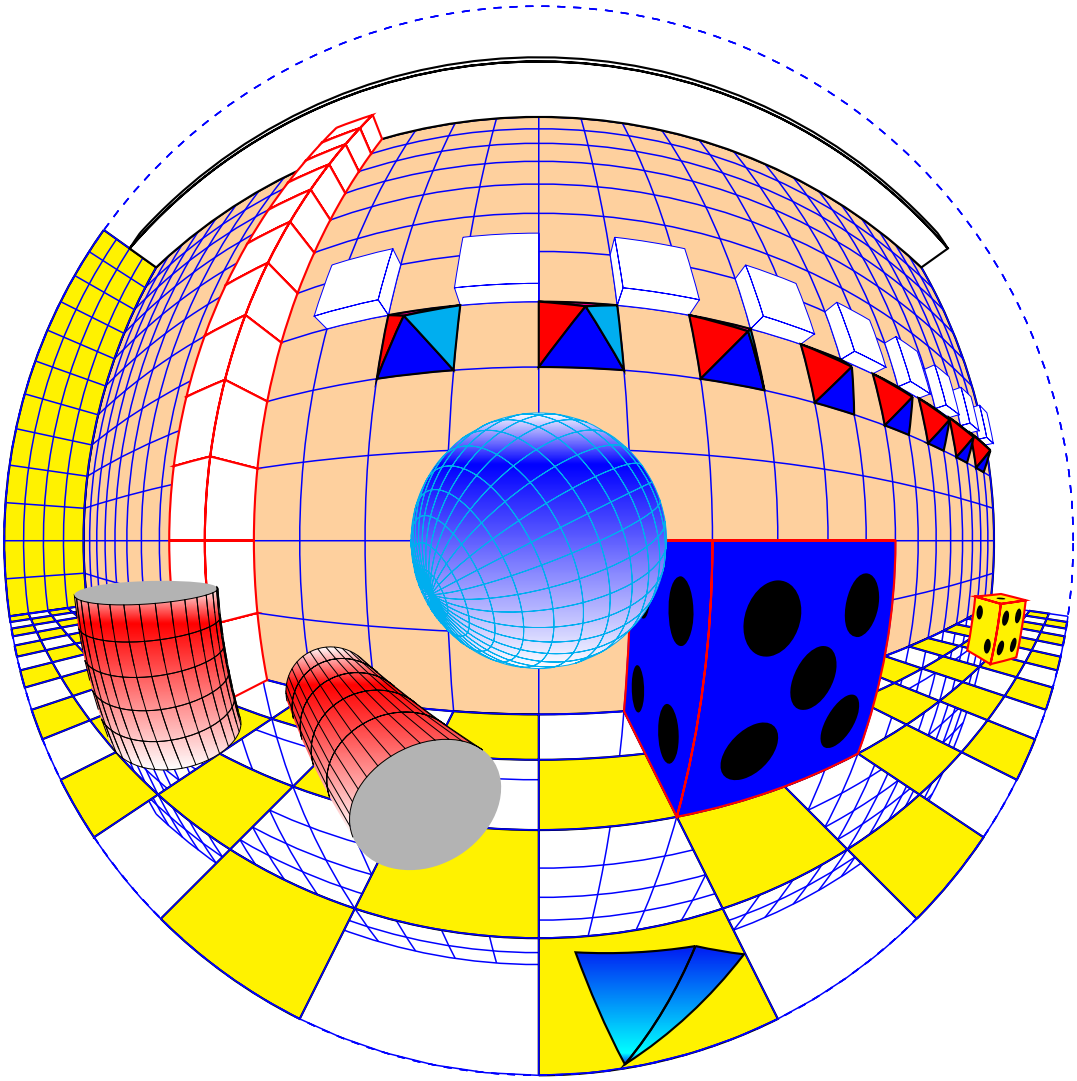
2 novembre 2011

http://melusine.eu.org/syracuse/G/pstricks/

Syracuse

1 Exemples

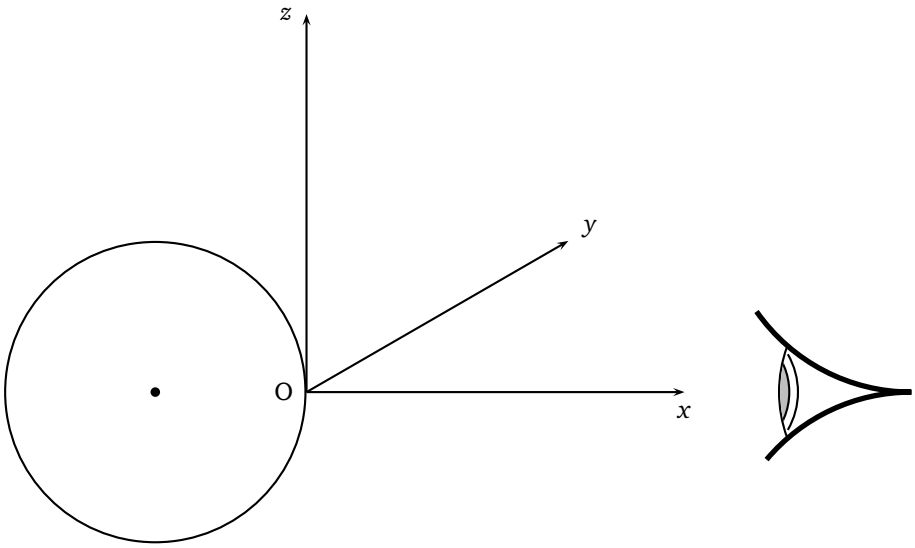




2 Utilisation des briques

2.1 Le repère

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



2.2 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 $=\theta\ \phi$
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 à un plan de front, c'est-à-dire parallèle à Oyz, est le vecteur $\vec{i}$ ou $-\vec{i}$,

$$\vec{u}(\theta = 0 \text{ ou } \pi, \phi = 0).$$

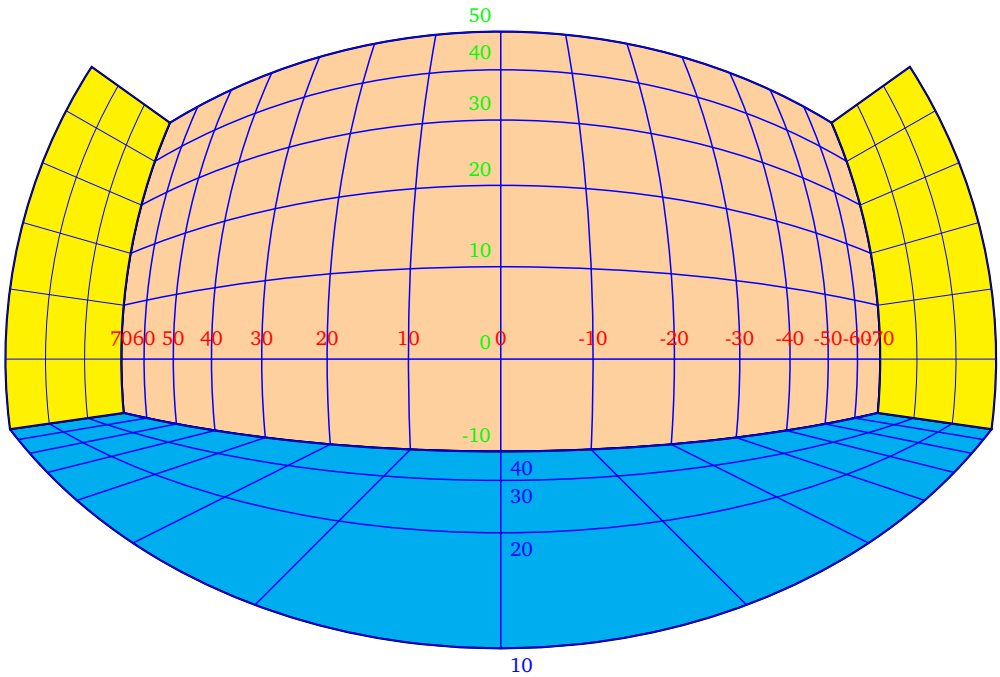
Un plan horizontal est parallèle à Oxy, un vecteur normal est soit $\vec{k}$, soit $-\vec{k}$, il est défini par

$$\vec{u}(\theta = \text{valeur indifférente}(0), \phi = \pi/2 \text{ ou } -\pi/2).$$

Un plan vertical parallèle à 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 règles habituelles d'orientation : le trièdre $(\vec{I}, \vec{J}, \vec{u})$ est direct, *mais le vecteur $\vec{I}$ a été pris dans le plan de référence Oxy.*

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



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 :

```
\psBowlGrid[normale=0 0,% par d\'{e}faut
                linecolor=blue,linewidth=0.2mm,grille=10,%
                Ymin=-50,Ymax=10,Xmax=70,Xmin=-70](40,0,0)%
```

Le quadrillage du plan horizontal par :

```
\psBowlGrid[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 :

```
\psBowlGrid[normale=90 0,%
                linecolor=blue,linewidth=0.05mm,grille=10,%
                Ymin=-60,Ymax=0,Xmin=-30,Xmax=0](40,70,-10)%
\psBowlGrid[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 même chose que précédemment en faisant :

```
\begin{pspicture}(-5,-4)(5,4)
% plan de front
\psset{Boule=false}
{\psset{normale=180 0,%
                Xorigine=40,%
                Yorigine=0,%
                Zorigine=0}
\psBowlFrame[fillstyle=solid,fillcolor=OrangePale](-70,-50)(70,10)
\psBowlGrid[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}
\psBowlFrame[fillstyle=solid,fillcolor=cyan](70,30)(-70,0)
\psBowlGrid[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}
\psBowlFrame[fillstyle=solid,fillcolor=yellow](0,0)(30,-60)
\psBowlGrid[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}
\psBowlFrame[fillstyle=solid,fillcolor=yellow](0,0)(30,-60)
\psBowlGrid[linecolor=blue,linewidth=0.05mm,grille=10,%
                  Ymin=-60,Ymax=0,Xmin=0,Xmax=30](40,-70,-10)}
\end{pspicture}
```

2.3 Dessiner un rectangle

Dans l'exemple précédent, j'ai dessiné les rectangles correspondants aux quadrillages. La commande est identique à celle du quadrillage avec les mêmes paramètres 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.

2.4 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 plaçant les points à la suite, comme dans la commande de PSTricks `\psline`.

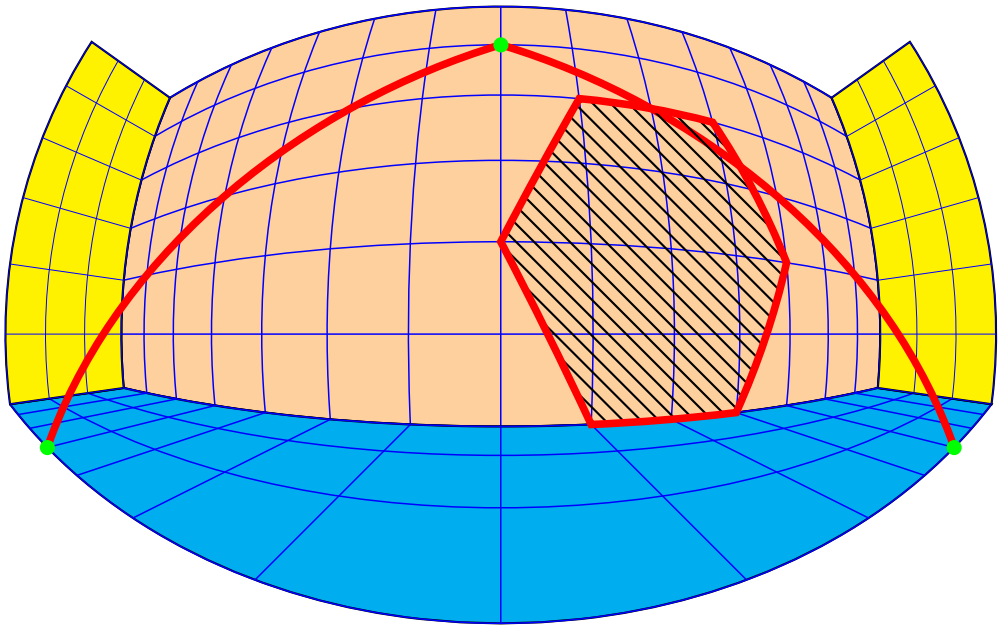
```
\pnodeBowl(40,0,40){A}
\pnodeBowl(10,-40,-10){B}
\pnodeBowl(10,40,-10){C}
\psdots[dotsize=1mm,linecolor=green](A)(B)(C)
\psBowlLine[linecolor=red](10,40,-10)(40,0,40)(10,-40,-10)
```

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

2.5 Dessiner un polygone

Commande identique à celle de PSTricks.

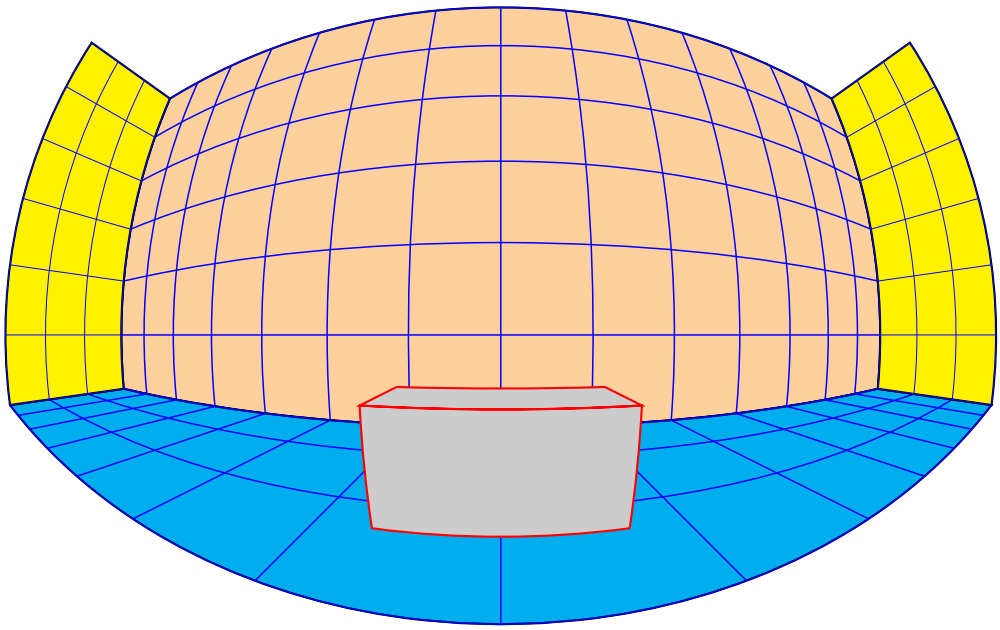
```
\psBowlPolygon[options](x0,y0,z0)(x1,y1,z1)...(xn,yn,zn)
```



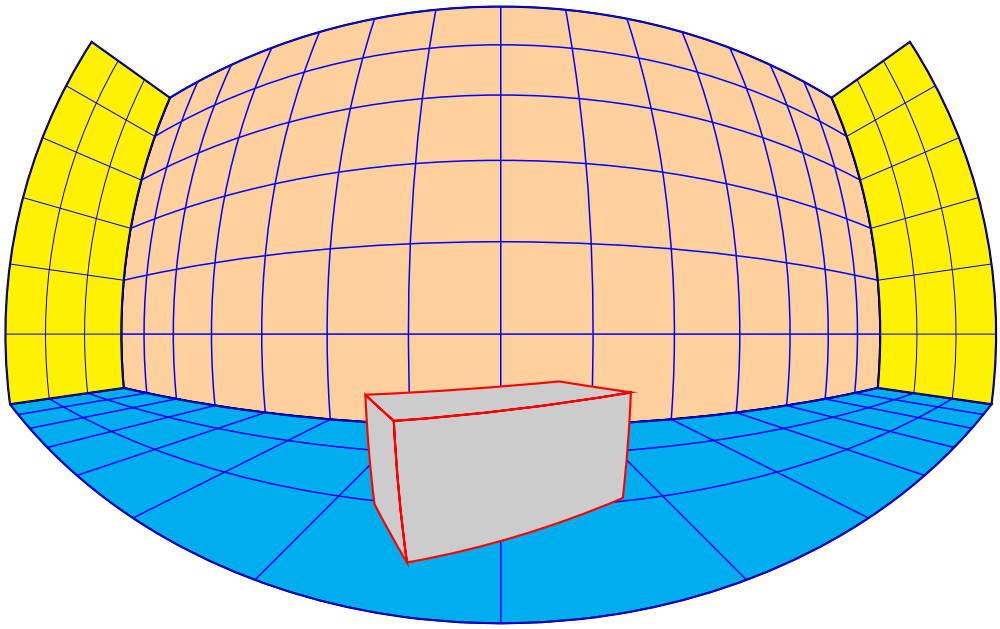
2.6 Parallélépipède

Le parallélépipède est défini par ses trois dimensions A,B,C. On peut le déplacer en fixant les coordonnées de son centre et le faire tourner avec autour des axes avec : RotX,RotY,RotZ dont les valeurs sont par défauts, positionnées à 0. `\psBowlCube[A=2.5,B=5,C=A](x,y,z)`

```
\psset{A=2.5,B=5,C=A,fillstyle=solid,fillcolor=GrisClair,linecolor=red}
\psBowlCube[A=2.5,B=5,C=A](10,0,5)
```

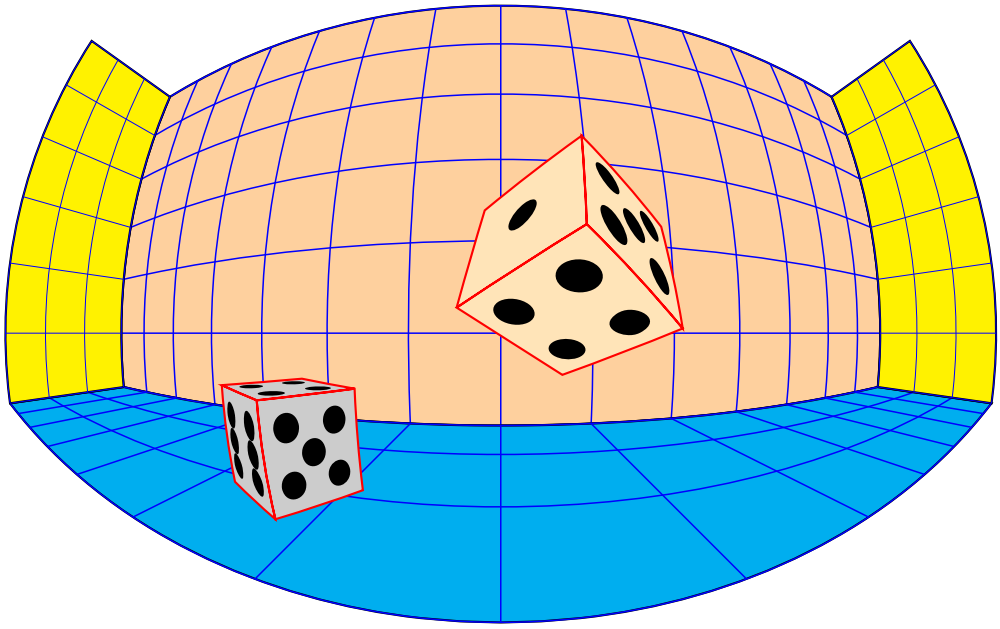


On le fait tourner autour de Oz avec RotZ=30 etc.



2.7 Le dé

Le dé possède une commande propre `\psBowlDie`, qui en interne utilise celle du `\psBowlCube` en affichant les points. Voici quelques dés.



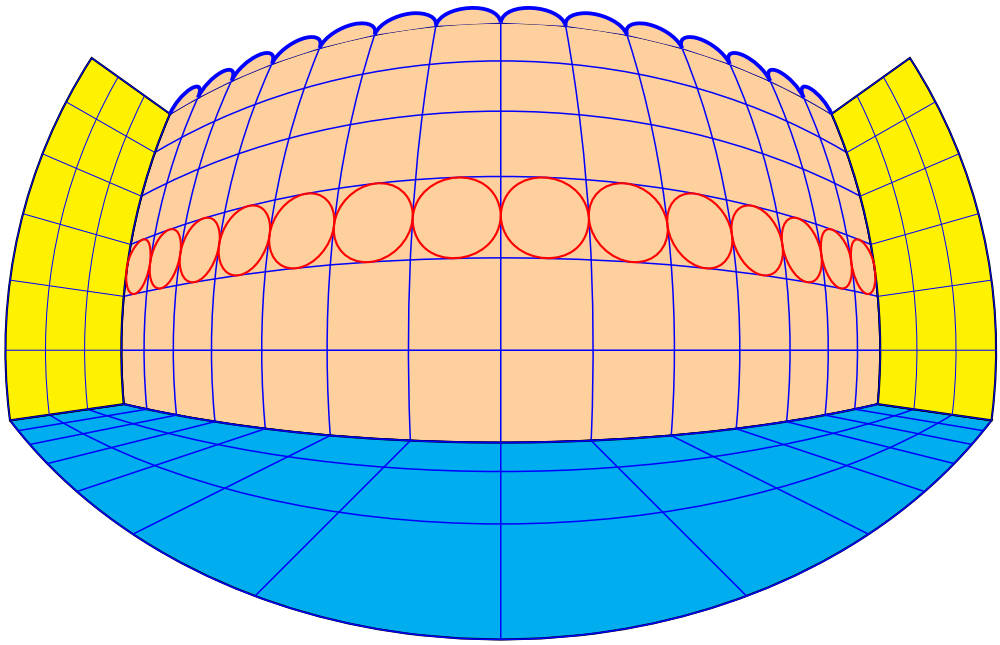
```
\psset{fillstyle=solid,fillcolor=GrisClair,linecolor=red}
\psBowlDie[A=2.5,B=A,C=A,RotZ=60,RotX=90](15,10,-5)
\psset{fillstyle=solid,fillcolor=OrangeTresPale,linecolor=red}
\psBowlDie[A=5,B=A,C=A,RotZ=45,RotY=20,RotX=-60](25,-5,5)
```

2.8 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 à ce plan. Ces données étant posées, il n'y plus que le rayon du cercle à fixer et les deux angles limites s'il s'agit de l'arc de cercle.

`\psBowlCircle[...](x,y,z){radius}`

`\psBowlArc[...](x,y,z){radius}{angle_min}{angle_max}` : respectez l'ordre des angles limites sinon l'interpréteur PostScript signalera une erreur.

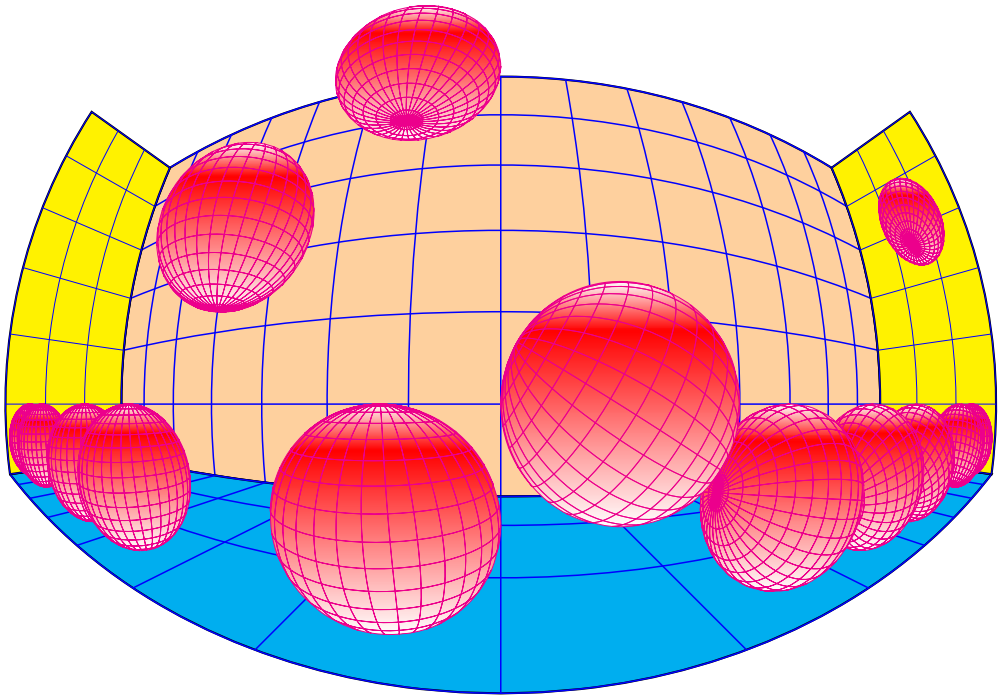


Des cercles et des arcs de cercle sur le plan de front :

```
\multido{\iYorigine=-65+10}{14}{%
\psBowlCircle[linecolor=red](40,\iYorigine,15){5}}
\multido{\iYorigine=-65+10}{14}{%
\psBowlArc[linecolor=blue,linewidth=0.5mm,%
         fillstyle=solid,fillcolor=OrangePale](40,\iYorigine,50){5}{180}{360}}
```

2.9 La sphère

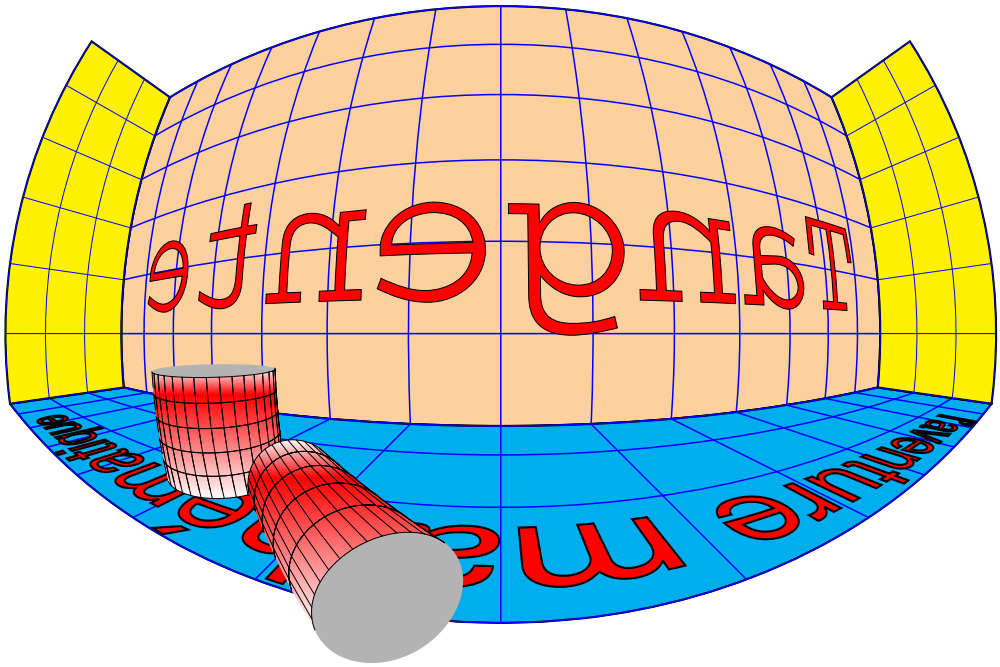
Elle est définie par son rayon, `\psBowlSphere[options](x,y,z){rayon}`, on la fait tourner autour de son centre avec [RotX,RotY,RotZ].



2.10 Dessiner un cylindre

Le cylindre sera donné par le rayon de sa base et sa hauteur. `\psBowlCylinder[options](x,y,z){rayon}{hauteur}`, on le fait tourner autour autour des axes avec RotX,RotY,RotZ.

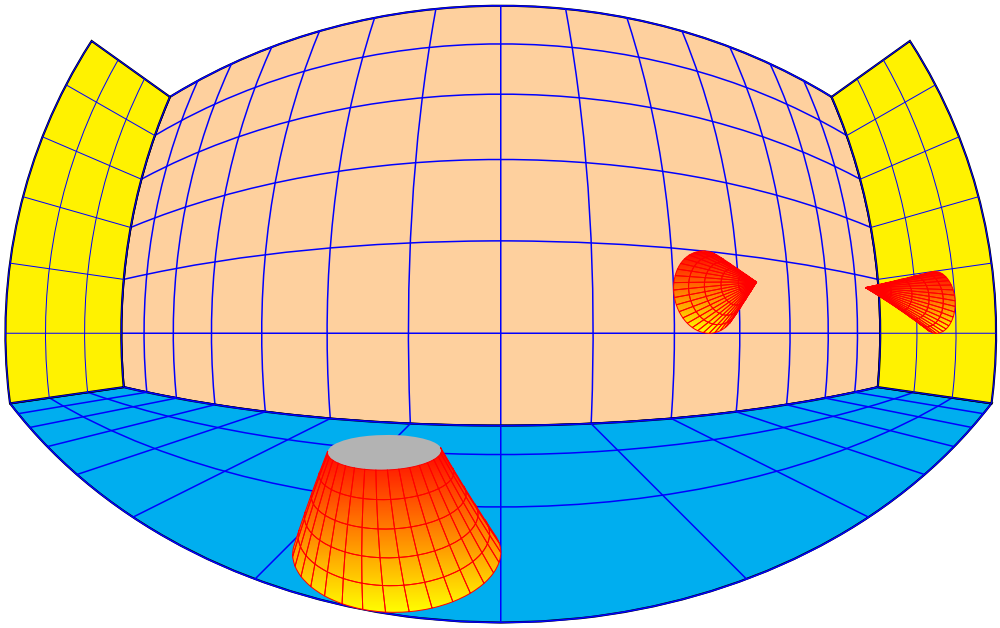
`\psBowlCylinder[options](x,y,z){radius}{hauteur}`



2.11 Dessiner un cône

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

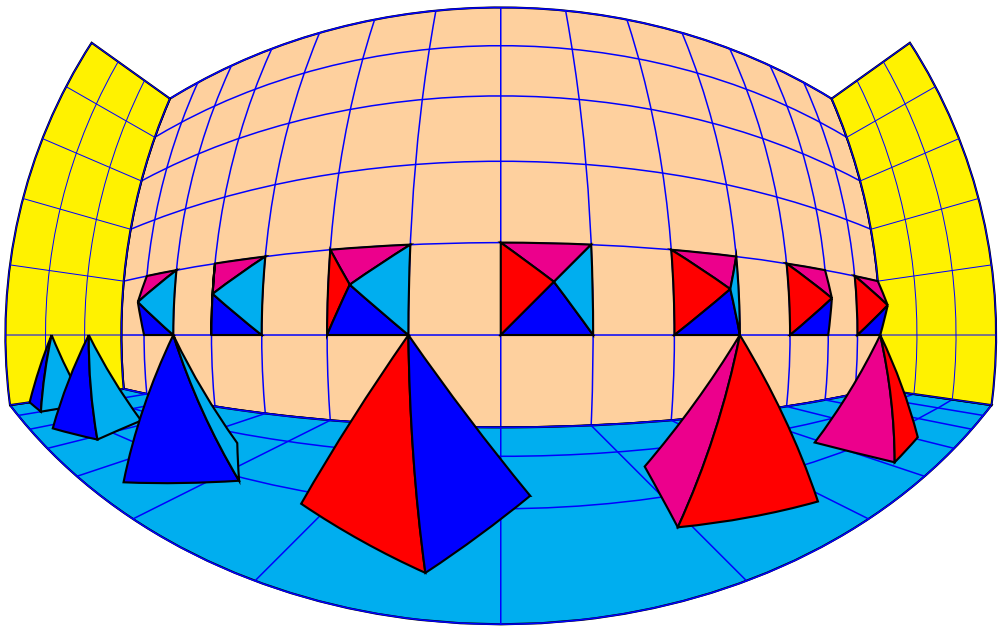
`\psBowlCone[fracHcone=0.5](x,y,z){radius}{hauteur}` fracHcone est obligatoirement un nombre compris entre 0 et 1.



```
\psset{linecolor=red,fillstyle=gradient,gradbegin=yellow,%
      gradend=red,gradmidpoint=0,linewidth=0.05mm}
\psBowlCone[fracHcone=0.5](15,5,-10){5}{10}
\psBowlCone[RotY=-90](40,-25,5){5}{10}
\psBowlCone[RotX=-90](25,-70,5){5}{30}
```

2.12 Tracer une pyramide

La pyramide à base carrée est donnée par le demi-côté (A)de sa base et sa hauteur (Hpyramide), que l'on peut faire tourner et placer où on veut, comme les objets précédents.`\psBowlPyramid[options](x,y,z){A}{H}`



```
\psset{fillstyle=solid,linecolor=black,A=5}
\multido{\nY=-65+20.0}{7}{%
\psset{RotX=0,RotZ=0,RotY=-90,CX=40,CZ=5,CY=\nY,Hpyramide=5}
\psBowlPyramid[RotX=0,RotZ=0,RotY=-90](40,\nY,5){5}{5}}
\multido{\nY=65+-20.0,\iRotZ=0+20}{6}{%
\psBowlPyramid[RotX=0,RotZ=\iRotZ,RotY=0](20,\nY,-10){5}{10}}
```

3 Image d'un texte

C'est la commande `\psBowlText[options](x,y,z){texte}` qui permet d'afficher le texte souhaité avec les paramètres suivants :

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

3.1 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.

`\psBowlImage[options](x,y,z){fichier.eps}`

4 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.