3D view with pst-vue3d

Manuel Luque (mluque5130@aol.com)
in cooperation with Herbert Voß (voss@perce.de)

13th August 2004

1 Presentation

The 3D representation of an object or a landscape is one of the most interesting subject in computer science and have many industrial applications (car and plane design, video game etc. . .). In a smaller way, one can obtain very didactic realizations using PSTricks with two peculiarities:

- using PostScript ;
- being manageable through LaTeX.

Package `pst-key` of David Carlisle allows to write commands with parameters. Using this as an interface, one can observe the result of little modifications of some parameters. Our parameters being here: the position of the watcher, the choice of an solid (cube, sphere etc. . .) and many other things. I want to signal that

- Regarding 3D representation, one does not forget the package pst-3d by Timothy Van Zandt who has used the best part of PostScript. Althrought limited to parallel projections, this package allows to draw very interesting 3D figure.[1]
- Thanks to Denis Girou, i have discovered the package `pst-key` and I have learned it.
- I have written another package for drawing picture reflecting in spherical mirrors.[2]

  It is a french paper which illustrate a study of Pr. Henri Bouasse from this book *Optique supérieure*, edited in 1917 by Delagrave.

2 Aims

First, we want to draw the 3D representation with elimination of the hidden parts of some objects.

The position of the watcher will be defined by its spherical coordinates: the distances from the origin, the longitude θ and the latitude ϕ. We will choose, too, the distance of the projection screen from this point.

Second, we want to define some $3D$ elements of the scene : the bricks.

The following bricks are already defined

- A box given by its three dimensions `A,B,C`: it could be turn into a cube or a dice.
- A point which can be defined it two ways
  - By cartesian coordinates (x, y, z)
  - Or by spherical coordinates (R, θ, ϕ) (θ, ϕ are, respectively, longitude and latitude).
- A rectangle.
- A circle defined by the normal line to its plane, its center and its radius. An arc is defined as the circle with two limit angles.
- A tetrahedron given by the coordinates of the center of its base and the radius of the circle containing the vertex of each faces. We can make it rotate.
- A square pyramid given by the half of the length of the side of its base and its height. We can make it rotate and move.
- A sphere given by the coordinates of its center `\SphereThreeD(x,y,z){Radius}` and its radius. We can make it rotate with the parameters `RotX=...`, `RotY=...`, `RotZ=...` We can choose to draw only some meridians and parallel circles.
- A solid or empty half-sphere (same parameters than a sphere)
- A vertical cylinder defined by its radius and its height. We can make it rotate using the parameters `RotX=...`, `RotY=...`, `RotZ=...` An we can choose the center of its base in the same way than the Sphere.

[1] A lot of different examples for 3D images are available at: http://members.aol.com/Mluque5130/

[2] http://melusine.eu.org/syracuse/mluque/BouleMiroir/boulemiroir.html

- A cone and a truncated cone defined by the radius of their base, the height and the height of the truncature.

To construct a scene, one may choose himself the order of the objects. For example, if an object 1 is partially hidden by an object 2, we write, in the list of commands, first object 1 and second object 2.

3 Rotating in the 3D space

A 3D object can be rotated around every axes with the `RotX`, `RotY` and `RotZ` option. They can be mixed in every combination. Figure 1 shows how a rotation around the z-axes works.

```
\begin{pspicture}(-1.5,-1.5)(1.5,1.5)
 \psset{THETA=70,PHI=30,Dobs=200,Decran=10}
 \psset{A=5,B=5,C=A,fillstyle=solid,%
  fillcolor=GrisClair,%
  linecolor=red, RotZ=\iRotZ}
 \tapis\DieThreeD(0,0,0)%
 \LineThreeD[linecolor=red,linestyle=dashed,%
  arrows=->](0,0,0)(0,0,25)
 \uput[180](Z'){\texttt{RotZ=\iRotZ}}
\end{pspicture}\hfill %
```

4 Location of the cube in the space

Suppose that one wants to place a 10-units edge cube at the point $(x = 40, y = 40, z = 35)$. First, the half edge of the cube will be define by the parameters : `A=5,B=5,C=5`, and next the coordinates of its center by `(40,40,35)`. On the figure, the period of the grid is 10 units (figure 2).

To make it rotate of around OX , one adds the parameter `RotX=90`(figure 4).

Three successive rotations around three axes with: `RotX=60,RotY=20,RotZ=110`, are illustrate in figure 5.

5 Constructions using cubes

This section was done after a book first published in 1873 and titled:

for children at infant school! One can not be surprised that theses kinds of pedagogue gave rise to the generation of Eintein, Maxwell, Bohr etc.

Observing figure from off :

```
\psset{PHI=90,THETA=0}
```

one obtains classical geometric figures :

(14) (15) (16) (17) (18) (19) (20) (21).

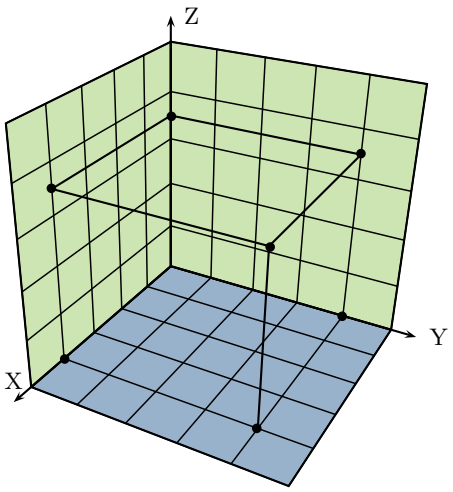

Figure 2: Origin `(40,40,35)`

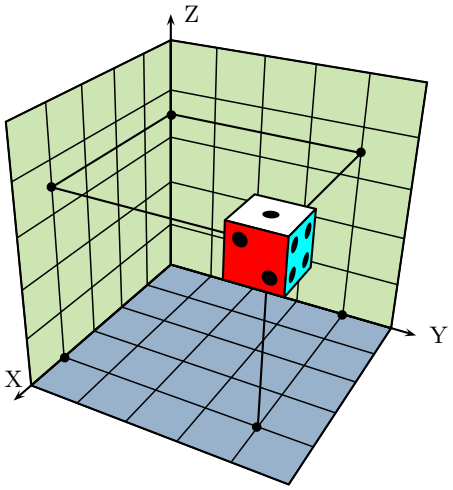

Figure 3: The placed cube.

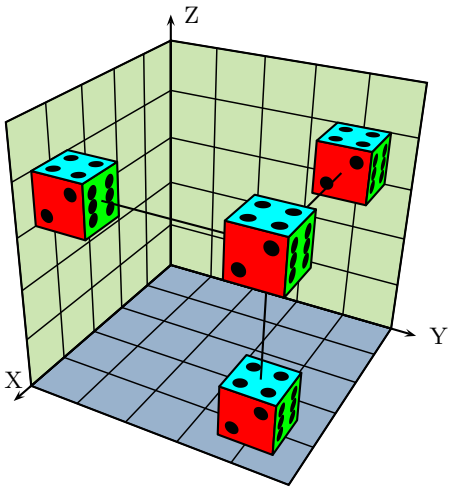

Figure 4: 90° rotation around OX and plane projections.

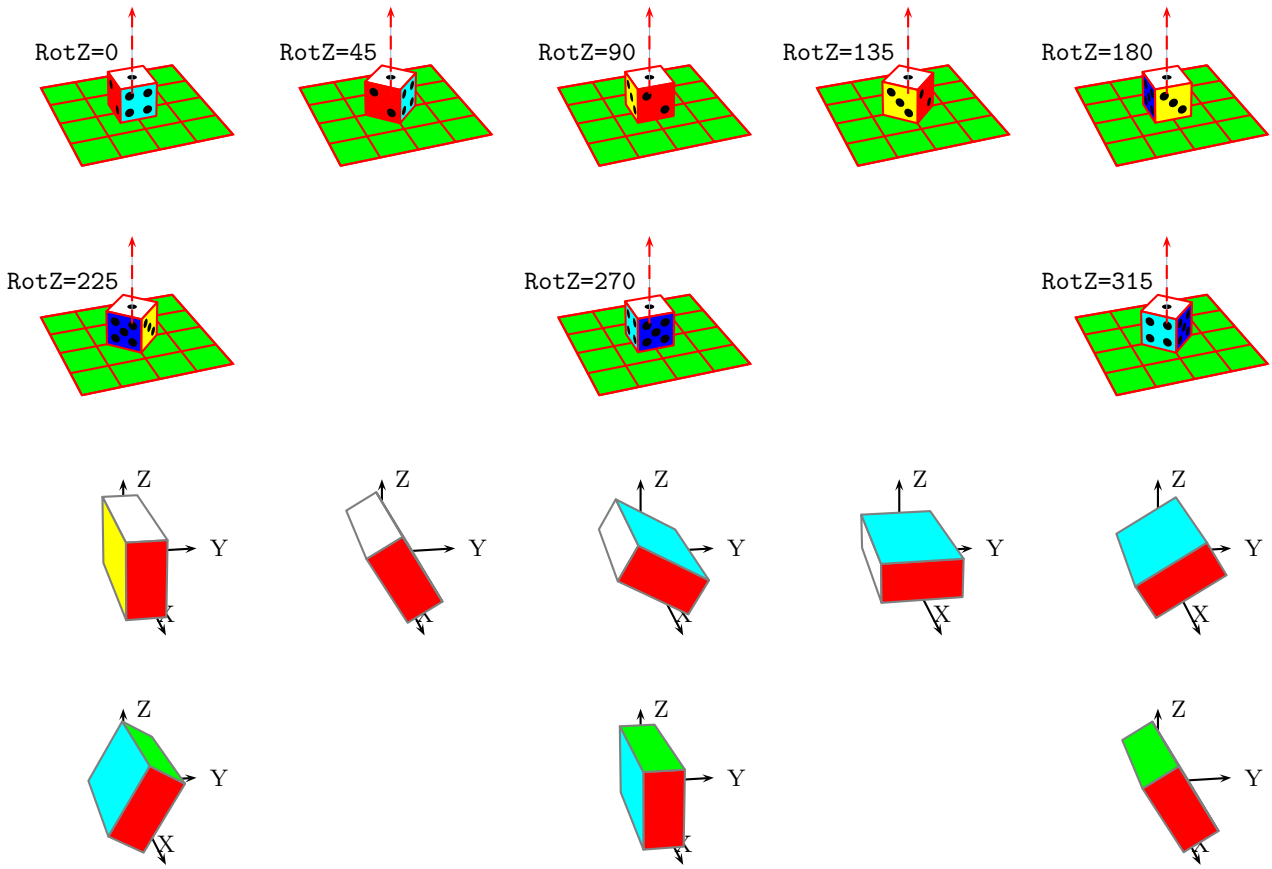

Figure 1: Diffenerent views of a die and a cube

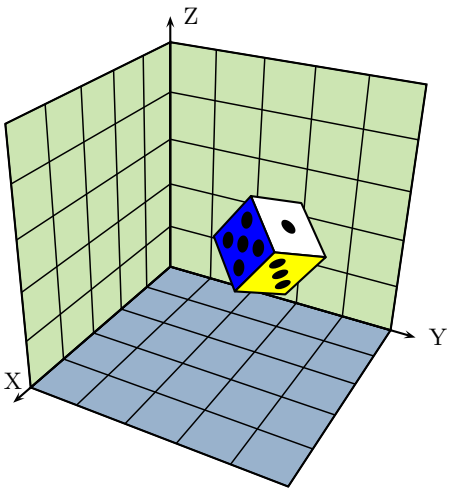

Figure 5: rotations around OX, OY et OZ : `RotX=60,RotY=20,RotZ=110`.

MÉTHODE INTUITIVE

EXERCICES ET TRAVAUX

POUR LES ENFANTS

SELON LA MÉTHODE ET LES PROCÉDÉS

de **PESTALOZZI et FRŒBEL**

M^{me} FANNY DELON

Directrice d'une École professionnelle à Paris

M. CH. DELON

Licencié ès sciences

PARIS

1873

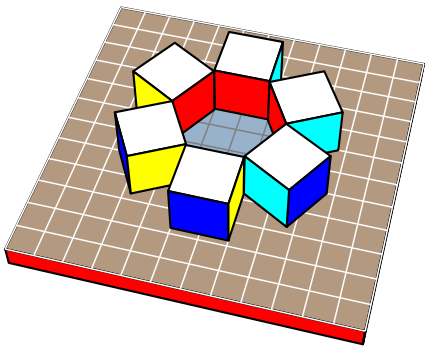

Figure 6: hexagon.

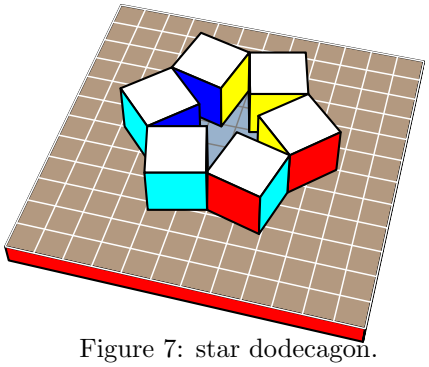

Figure 7: star dodecagon.

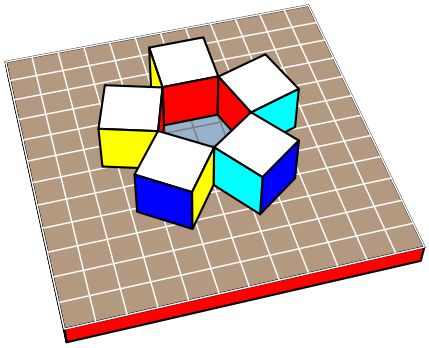

Figure 8: pentagon.

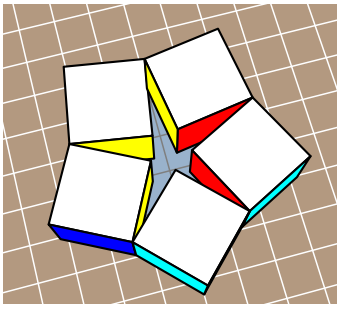

Figure 9: star decagon.

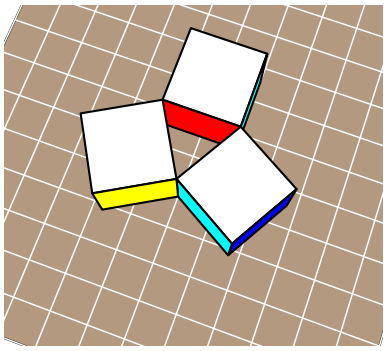

Figure 10: triangle.

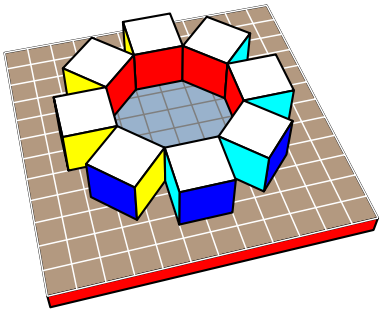

Figure 11: octogon.

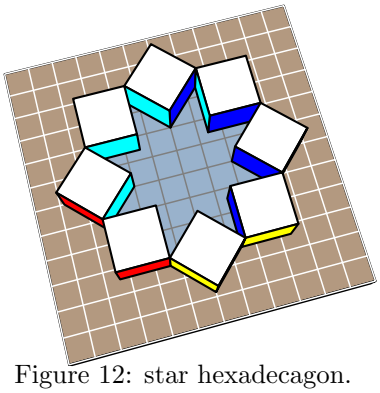

Figure 12: star hexadecagon.

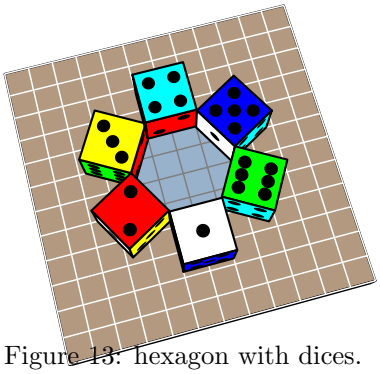

Figure 13: hexagon with dices.

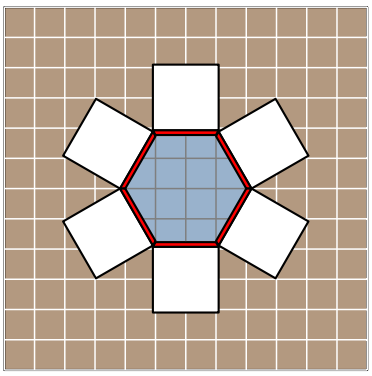

Figure 14: “flat” hexagon.

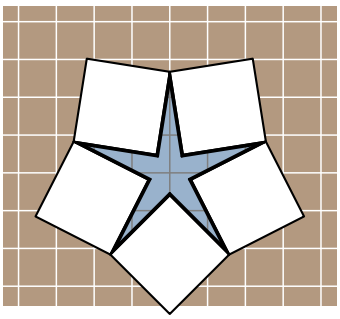

Figure 15: “flat” star dodecagone.

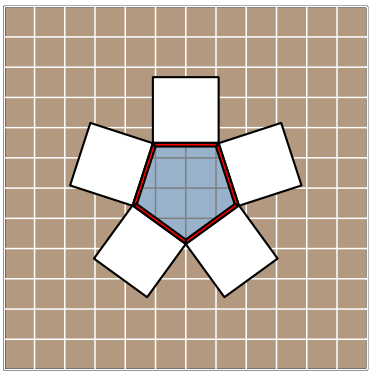

Figure 16: “flat” pentagon.

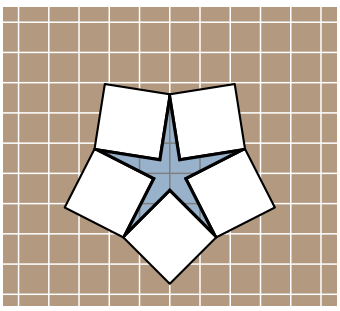

Figure 17: “flat” star decagon.

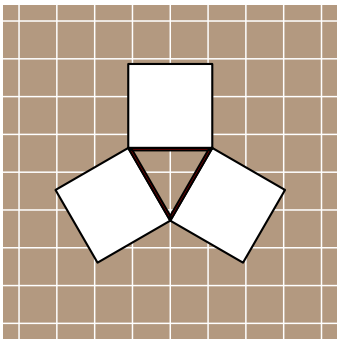

Figure 18: “flat” triangle.

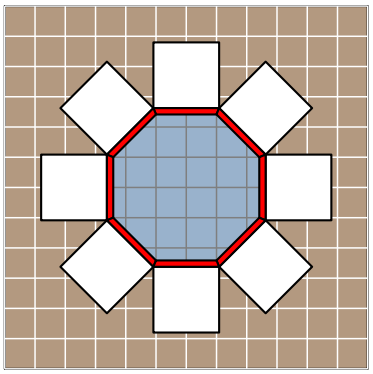

Figure 19: “flat” octogon.

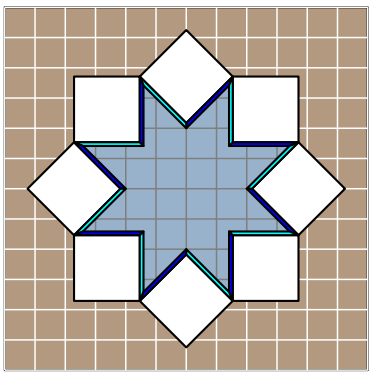

Figure 20: “flat” star hexadecagon.

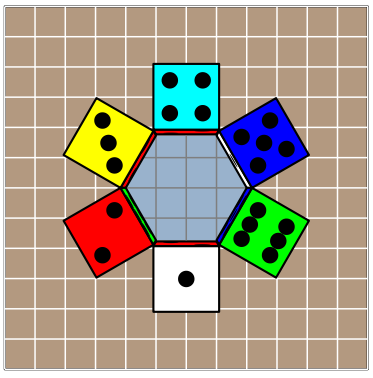

Figure 21: “flat” hexagon with dices.

6 Sphere, part of sphere, half-sphere, parallels and meridians

Beside `sphereThreeD` there exist several macro for spheres:

- `SphereInverseThreeD`
- `\SphereCercleThreeD`
- `\SphereMeridienThreeD`
- `\DemiSphereThreeDThreeD`
- `\SphereCreuseThreeD`
- `\PortionSphereThreeD`

The macro:

```
\SphereThreeD(10,30,20){20}
```

draws the sphere defined by the coordinates of its centre and its radius which is shown in figure 22 together with the macro

```
\PortionSphereThreeD(0,0,0){20}
```

and some more additional lines.

```
\begin{pspicture}(-3,-3.5)(3,5)
\psset{THETA=30,PHI=30,Dobs=100,Decran=10}
{\psset{style=GradGrayWhite}%
\SphereThreeD(0,0,0){20}
\psset{fillstyle=solid,fillcolor=gray}
\PortionSphereThreeD(0,0,0){20}
\pNodeThreeD(20;10;10){C1}
\pNodeThreeD(40;10;10){D1}
\psline(C1)(D1)
\pNodeThreeD(20;10;-10){C2}
\pNodeThreeD(40;10;-10){D2}
\psline(C2)(D2)
\pNodeThreeD(20;-10;-10){C3}
\pNodeThreeD(40;-10;-10){D3}
\psline(C3)(D3)
\pNodeThreeD(20;-10;10){C4}
\pNodeThreeD(40;-10;10){D4}
\psline(C4)(D4)
\PortionSphereThreeD%
 [style=GradGrayWhite](0,0,0){40}}
% PhiCercle=latitude of the cercle
% \SphereCercle[PhiCercle=...]{radius}
\psset{linecolor=white,PhiCercle=45}
\SphereCercleThreeD(0,0,0){20}
% ThetaMeridien=longitude of the meridian
% \SphereMeridien[ThetaMeridien=...]{radius}
\SphereMeridienThreeD%
 [ThetaMeridien=45](0,0,0){20}
\pNodeThreeD(20;45;45){A}
\pNodeThreeD(50;45;45){B}
\psline[linecolor=black]{->}(A)(B)
\pNodeThreeD(20;0;90){Nord}
\pNodeThreeD(40;0;90){Nord1}
\psline[linecolor=black]{->}(Nord)(Nord1)
\SphereCercleThreeD[PhiCercle=0](0,0,0){20}
\SphereMeridienThreeD%
 [ThetaMeridien=0](0,0,0){20}
\end{pspicture}
```

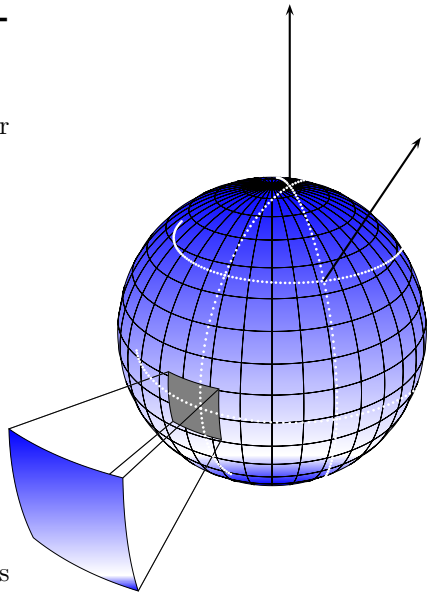

Figure 22: A Sphere.

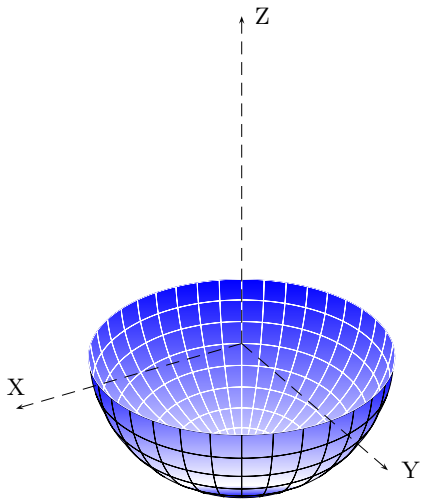

Figure 23: half-sphere.

7 A Hole in a sphere

It is a rectangular hole whose the size are meridian and parallels arcs (figure 25).

We define the part of the sphere setting its radius, the center of the sphere and the $\Delta\phi$ and $\Delta\theta$.

```
\PortionSphereThreeD[PortionSpherePHI=45,%
                PortionSphereTHETA=0,%
                        DeltaPHI=45,%
                   DeltaTHETA=20](0,0,0){20}
```

There are the parameters of the first hole. The radius is 20.

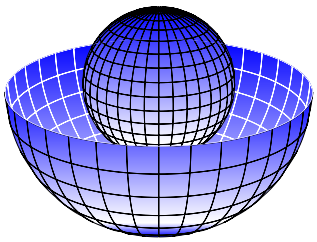

Figure 24: levitation

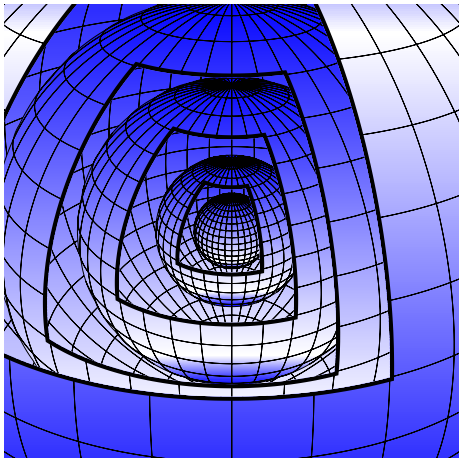

Figure 25: A Hole in a sphere.

```
{\psset{fillstyle=gradient,%
            gradbegin=white,%
               gradend=blue,%
            gradmidpoint=0.2,%
              linecolor=cyan,%
              linewidth=0.1mm}
\SphereThreeD(0,0,0){20}}%
\begin{psclip}{%
\PortionSphereThreeD[PortionSpherePHI=45,%
    DeltaPHI=45,DeltaTHETA=20](0,0,0){20}}
\SphereInverseThreeD[fillstyle=solid,%
                  fillcolor=red,%
                linecolor=blue](0,0,0){20}%
\end{psclip}%
```

This is the tricks to see the inner of the sphere. `\SphereInverse` define the hidden part of the sphere.

8 Drawing a cylinder

A cylinder is defined by the radius of its base and its height. The center of the base is set in the usual way, and RotX,RotY,RotZ make it rotate around the axes.

```
\CylindreThreeD(x,y,z){radius}{hauteur}
```

```
\CylindreThreeD(0,0,-5){10}{15}}
\psset{RotY=90}
\CylindreThreeD(15,15,-5){5}{20}
```

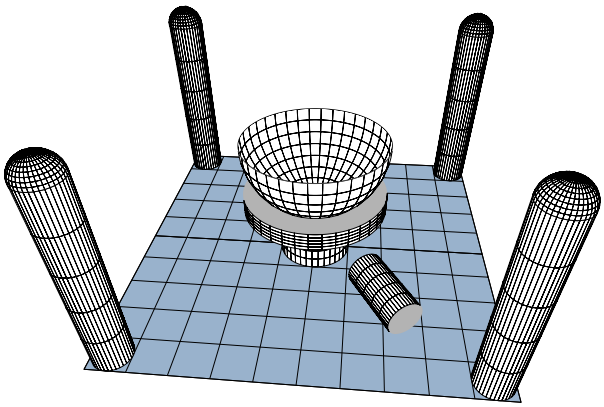

Figure 26: cylinders.

9 Tetrahedron, cone and square pyramid

9.1 square pyramid

```
\psset{A=...,Hpyramide=...}
\Pyramide
```

See the examples of figures (27) (28).

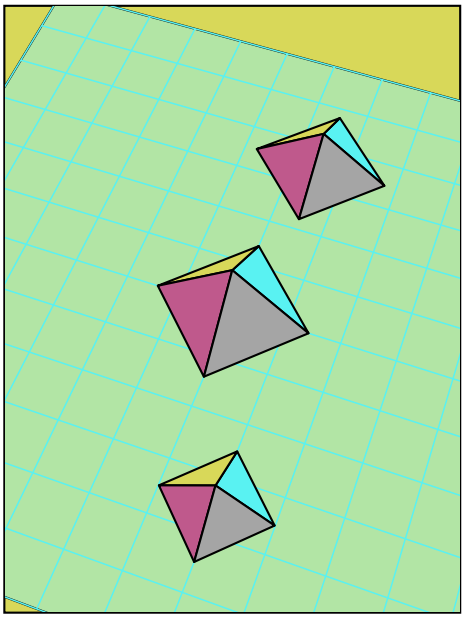

Figure 27: Pyramids of Egypt.

9.2 Cone

```
\ConeThreeD[fracHeight=...]
          (x,y,z){radius}{Height}
```

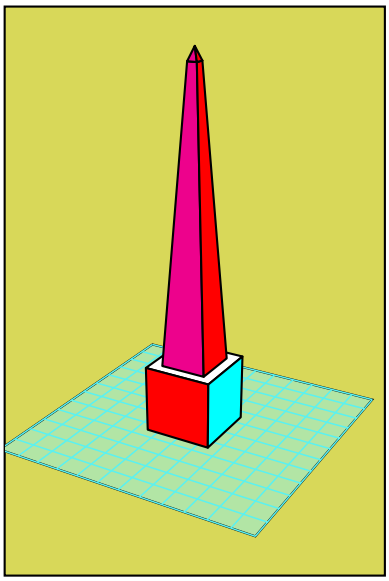

Figure 28: Obelisk of Egypt.

by default `fracHeight=1` : figure 29.

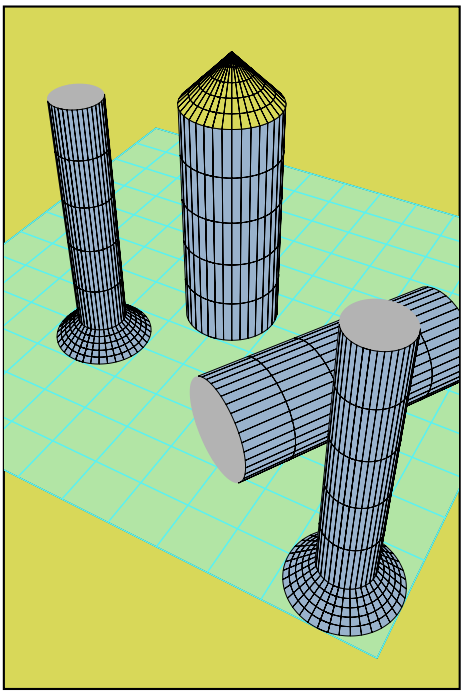

Figure 29: Cones and cylinders.

10 Points and lines

The command allowing to mark points and thus to draw lines and polygons can be used of two manners, either with the Cartesian coordinates

```
\pNodeThreeD(x,y,z){name}
```

or with the spherical coordinates :

```
\pNodeThreeD(radius;longitude;latitude)%
            {name of the point}
```

For example `\pNodeThreeD(25,-25,25){A}`, the point $A(25, 25, 25)$ places. Points being positioned, just to write `\psline(A)(B)`, to draw the segment AB.

On the figure 30, one drew a cube with its diagonals.

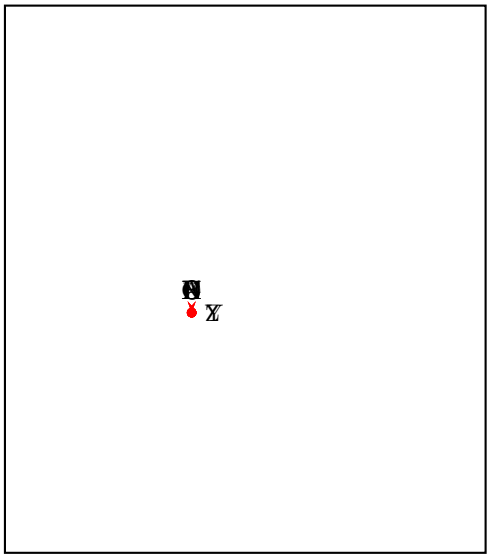

Figure 30: Points and lines.

11 Circles

A circle is defined by a vector normal for its plan by (θ, φ), with the following parameters for example:

```
normaleLongitude=60,normaleLatitude=90
```

The coordinates of his centre as well as his radius.

```
\CircleThreeD(x,y,z){radius}
```

The circles of the figure 31, were drawn with the following commands:

```
\psset{normaleLongitude=0,%
       normaleLatitude=90}
\multido{\iXorigine=-65+10}{14}{%
    \multido{\iYorigine=5+10}{5}{%
        \CircleThreeD[linecolor=red]%
            (\iXorigine,\iYorigine,0){5}}}
```

12 The macros and the options

12.1 The colors of the cube, the pyramid and tetraedre

The predefined colors for the different sides of a cube are always set in the `rgb` mode :

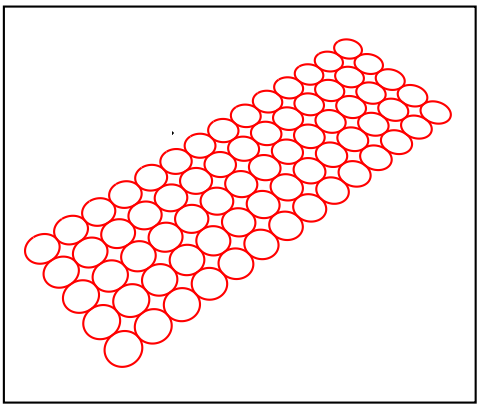

Figure 31: circles.

```
CubeColorFaceOne=1 1 0,%
CubeColorFaceTwo=0.9 0.9 0,%
CubeColorFaceThree=0.8 0.8 0,%
CubeColorFaceFour=0.7 0.7 0,%
CubeColorFaceFive=0.65 0.65 0,%
CubeColorFaceSix=0.75 0.75 0
```

The colors for the pyramid and the tetraedre are taken from the predefined ones:

```
ColorFaceD=cyan,
ColorFaceA=magenta,
ColorFaceB=red,
ColorFaceC=blue,
ColorFaceE=yellow
```

They can be changed in the usual way with the `\psset` macro.

12.2 Common parameters

`RotX=<value>, RotY=<value>, RotZ=<value>`

The predefined value is zero, means no rotation.

12.3 Cube

The following command places a parallelepiped with a length of $a = 40$, $b = 20$ and $c = 10$ units and it is placed with its center at the point $x = 25$, $y = 25$ and $z = 25$

```
\CubeThreeD[A=20,B=10,C=5](25,25,25)
```

In other words: the length of the sides is `2A,2B,2C` (see figure 32).

For rotations, let us consider the result of a rotation around one of the axes, while knowing that it is possible to combine them. The corresponding rotation of projection on the horizontal level is obtained with the parameter: `normaleLongitude=<degrees>` (figure 33).

There is no difference to a die, except that all sides have the same length.

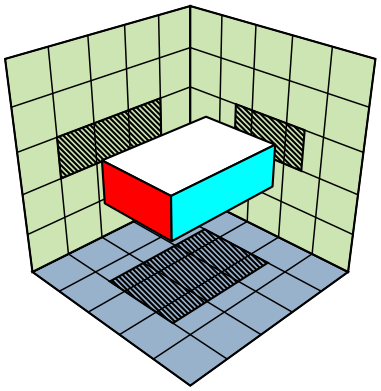

Figure 32: Parallelepiped

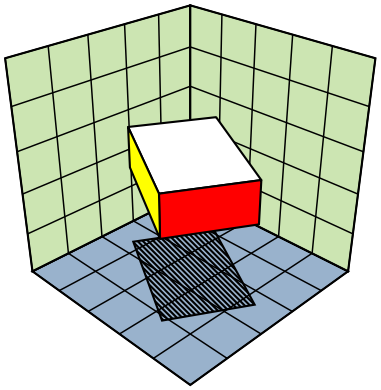

Figure 33: The same parallelepiped rotated with `RotZ=60`.

12.4 Cylinder and circle

In addition to the already quoted optional parameters the cylinder requires the obligatory parameters:

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

Projection on the horizontal level is obtained with the following values:

```
\CircleThreeD[normaleLongitude=0,%
              normaleLatitude=90,%
              fillstyle=vlines,%
              hatchsep=0.4mm](30,30,0){10}
```

The circle macro needs the following parameters:

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

Figure 35 shows an example of the above macros.

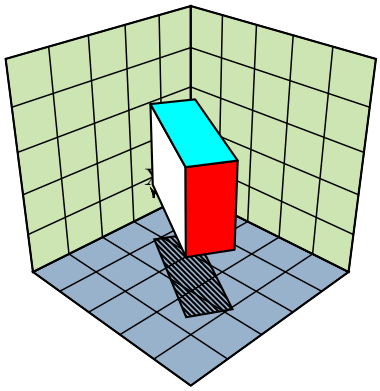

Figure 34: The same parallelepiped, rotated with the values `RotX=90,RotZ=60`

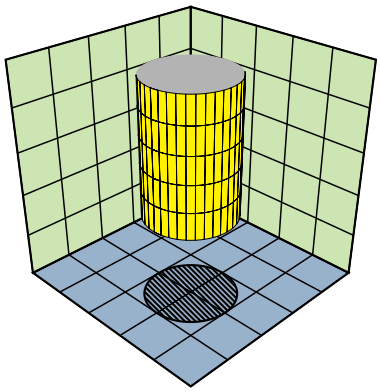

Figure 35: A cylinder with a radius of 10 units and a height of 50 units with its base center at `(30,30,20)`.

13 See the interior of a cube

The following option makes it possible to visualize the interior of the box, the result is seen in the figure 36 :

```
\DieThreeD(0,0,0)%
\begin{psclip}{%
\FrameThreeD[normaleLongitude=0,%
             normaleLatitude=90]%
    (0,0,10)(-10,-10)(10,10)}%
\DieThreeD[CubeInside=true](0,0,0)%
\end{psclip}%
```

Figure 36: An empty box.

14 Another package about the 3D

Concerning 3D and PSTricks, I indicate package of Herbert Voss (`pst-3dplot`), rather dedicated towards the representation of the mathematical functions.