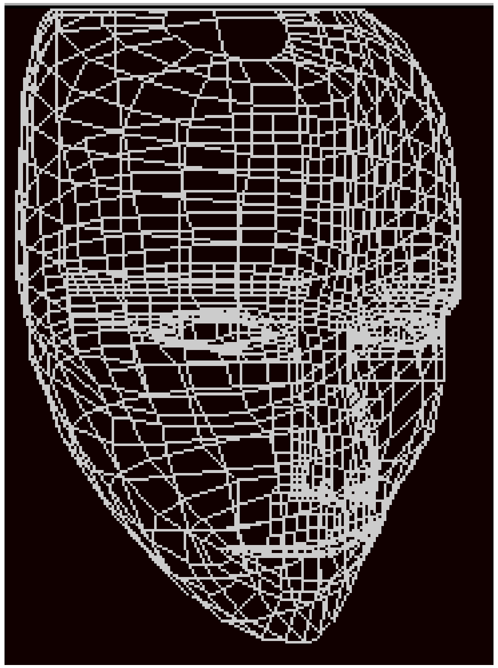
Synthèse d'images

(2) Polyèdres, facettage



Plan de l'exposé :

1- Polyèdres - définitions

2 - Polyèdres - représentations

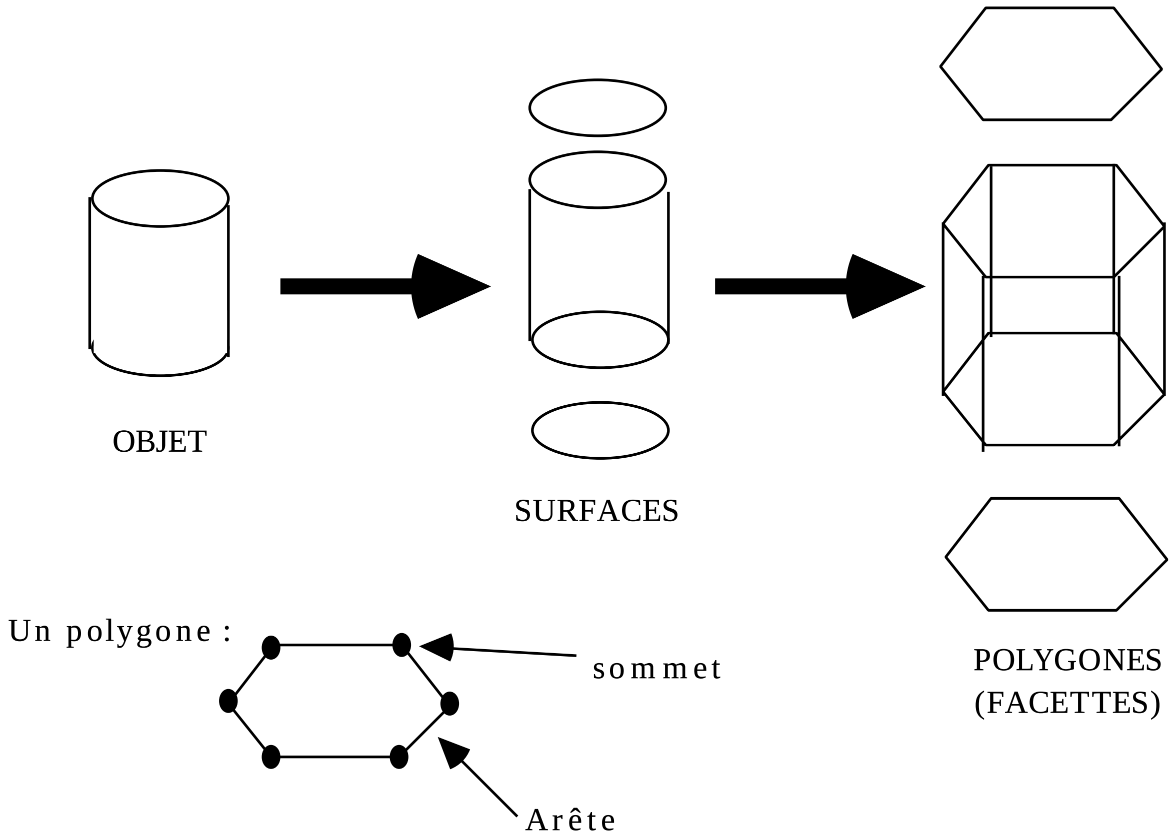
3- Surfaces d'élévation

4- Facettage (tesselation)

5- Exercices

v5 P. CUBAUD <cubaud@cnam.fr> Dept. informatique CNAM, Paris

1. Définitions



- Polyèdre

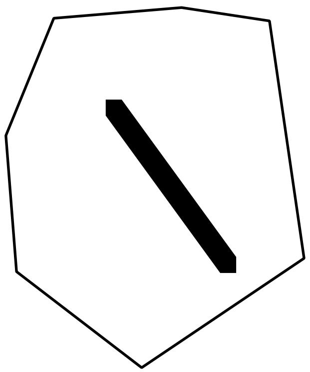
"solide limité de toutes parts par des polygones plans." (Petit Robert)

"solide limité par des polygones plans dont chaque côté (arête) est commun à exactement deux polygones." (Quid)

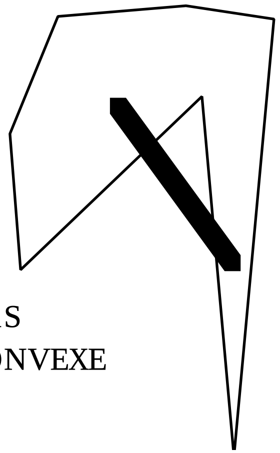
- Polyèdre convexe :

"Polyèdre situé tout entier du même côté du plan de chacune de ses faces" (Quid)

"Une droite reliant n'importe quels points de l'intérieur du polyèdre reste à l'intérieur" (Cubaud)



CONVEXE



PAS CONVEXE

- Formule d'Euler pour tout polyèdre convexe

S = nombre de sommets A = nombre d'arêtes F = nombre de faces

$$\text{On a : } S + F = A + 2$$

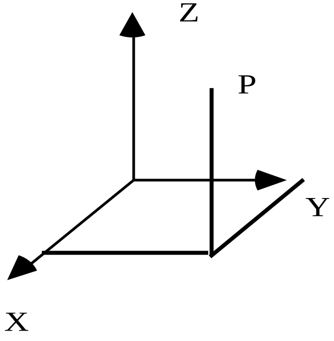
exemple : cube : S=8, F=6, A=12

Utile pour détecter un oubli dans la description. Condition nécessaire mais pas suffisante de convexité !!! En plus : au plus 3 arêtes ont un sommet donné en commun

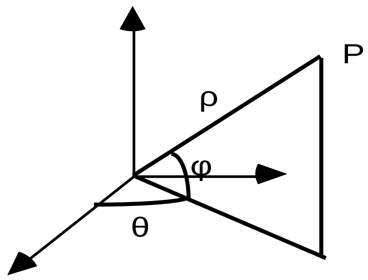
2. Représentations

- Systèmes de coordonnées

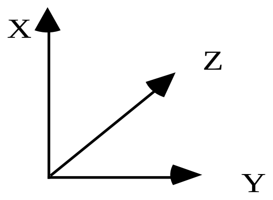
SYSTEME CARTESIEN



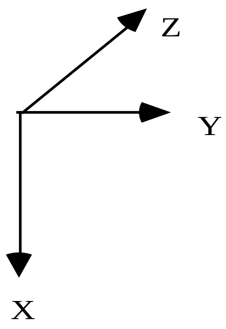
SYSTEME SPHERIQUE



SYSTEME DIRECT
(MAIN DROITE)



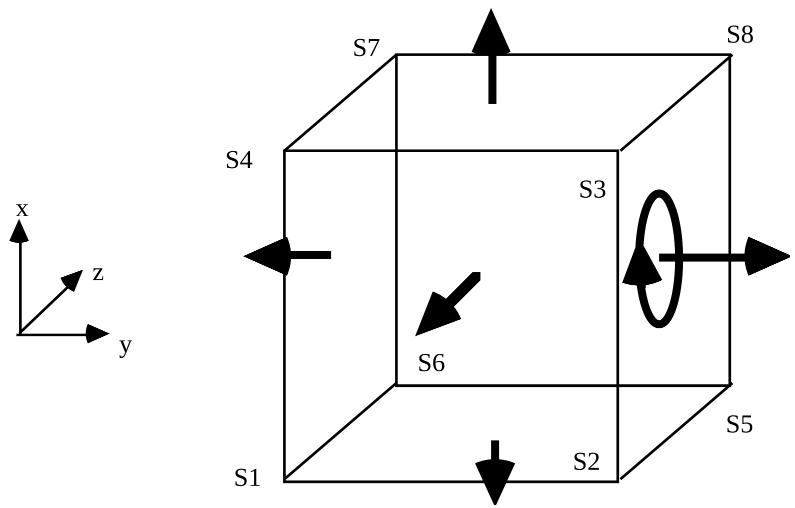
SYSTEME INDIRECT
(MAIN GAUCHE)



- Représentation explicite :

  - Chaque face F est décrite par une liste de sommets :

  F = ((x1,y1,z1), …, (xn,yn,zn))



  - Les sommets sont donnés dans l'ordre pour décrire la frontière de la face, et on suppose qu'il y a une arête entre (xn,yn,zn) et (x1,y1,z1)

Problèmes :

  - Gaspillage, car les arêtes communes sont décrites deux fois

  - Comment "prendre" le polyèdre par un de ses sommets sans connaître les faces concernées ?

- Représentation par liste de sommets
  - C'est la méthode la plus commune (PHIGS, VRML, PovRay)
  - On gère deux tables : TS = table des sommets et TF = table des faces
  - Exemple du cube :

TS	X	Y	Z
1	0	0	0
2	0	1	0
3	1	1	0
4	1	0	0
5	0	1	1
6	0	0	1
7	1	0	1
8	1	1	2

TF	
1	1, 2, 3, 4
2	2, 3 8, 5
3	5, 8, 7, 6
4	6, 7, 4, 1
5	3, 4, 7, 8
6	1, 2, 5, 6

Problèmes :

- Décrire les faces toujours dans le même sens de parcours des sommets (anti horlogique)
- Les arêtes communes sont toujours dessinées deux fois
- Comment retrouver facilement quelles faces ont telle arête en commun ?

- Représentation par liste d'arêtes

On gère 3 tables : TS = table des sommets, TF = table des faces, TA = table des arêtes

Exemple du cube :

TA	
1	S1, S2, F1, F6
2	S2, S3, F1, F2
3	S3, S4, F1, F5
4	S4, S1, F1, F4
5	S2, S5, F2, F6
6	S5, S8, F2, F5
7	S8, S3, F2, F5
8	S8, S7, F3, F5
9	S7, S6, F3, F4
	etc…

Résoud les problèmes précédents, mais plus "lourd"

Technique utilisée aussi en VRML

3. Surfaces d'élévation

• Une grille où chaque point représente une altitude. Très utilisé en topographie (cartographie).

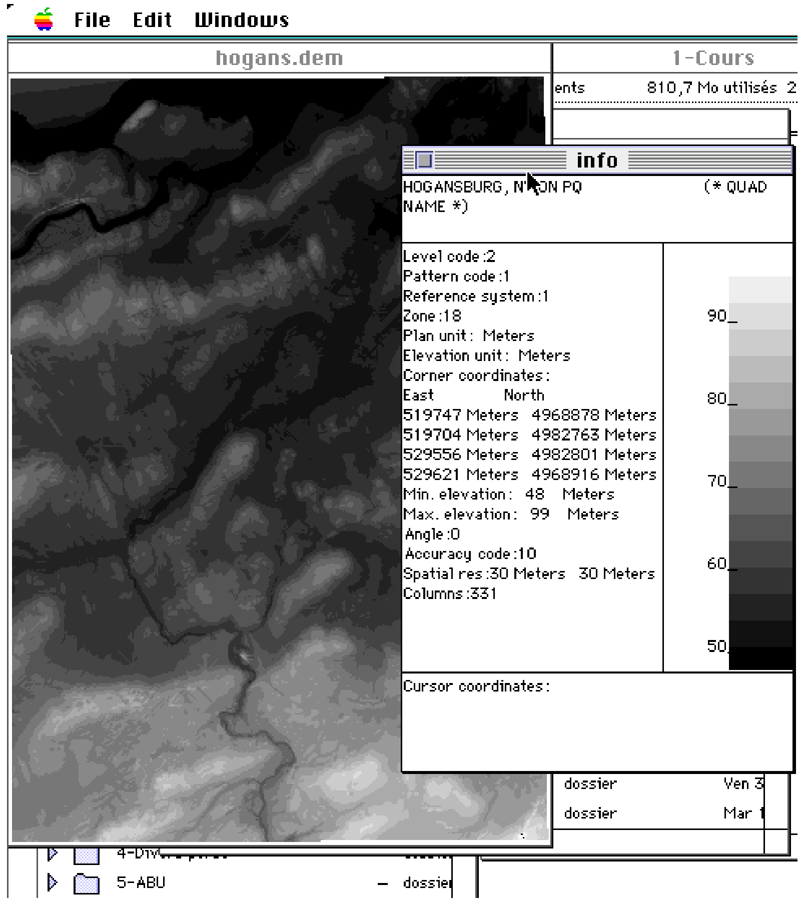
- Il existe de nombreux formats :

DEM (digital elevation model) issus du US Geological Survey (http://www.usgs.gov)

- Logiciels de visualisation :

MICRODEM (Windows 95 et NT)

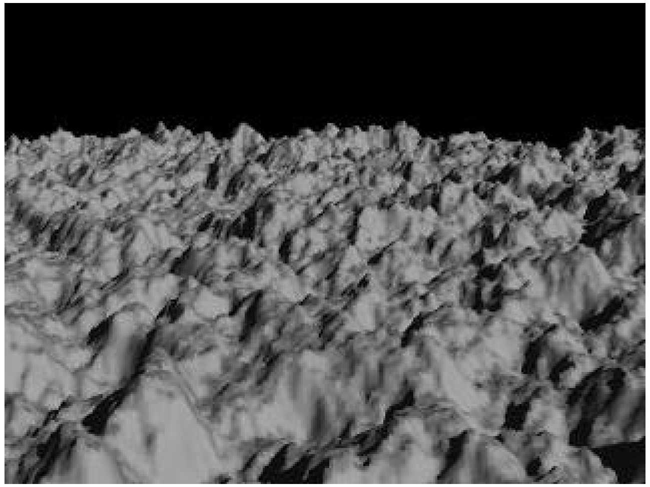
DEM Reader 1.5 (Mac)
www.electriciti.com/~brianw/DEM_Reader.html



• On peut utiliser un tableau de pixel pour représenter l'altitude en chaque point de la grille

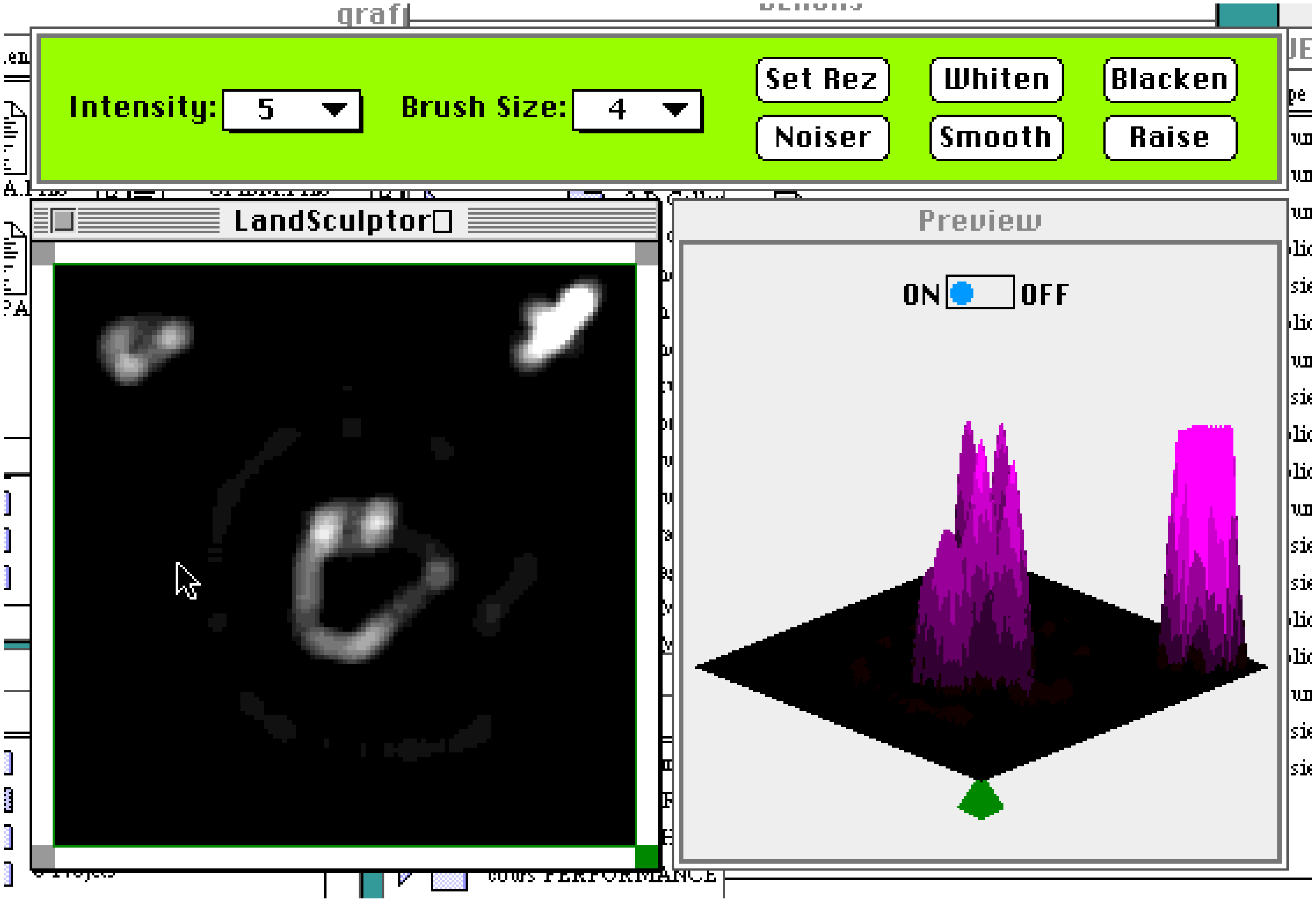
Exemple de POV :

```
  #include "colors.inc"

  camera{
    location <0, 2, -10>
    look_at 0
    angle 15
  }

light_source{ <1000,1000,-1000> White }

  height_field {
    tga "image.tga"
    smooth
    pigment { White }
    translate <-.5, -.5, -.5>
    scale <17, 1.75, 17>
  }
```



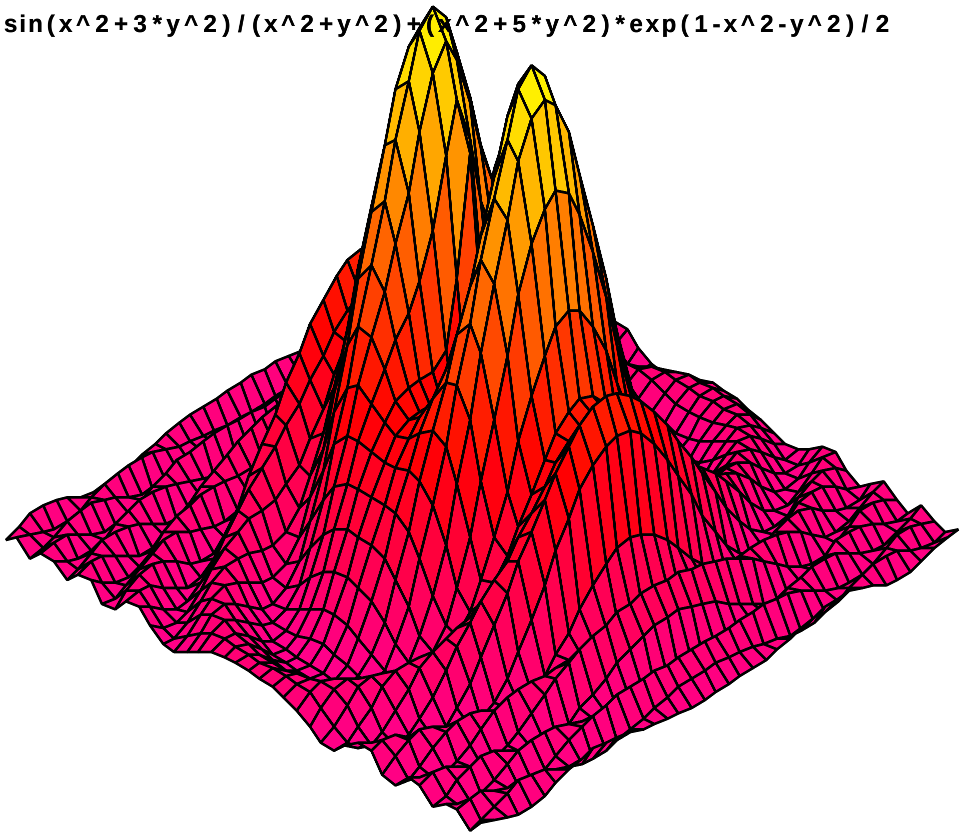
• Un petit logiciel de génération manuelle :

LandSculptor3D pour Macintosh



- Une autre approche : équations de surfaces z=f(x,y)

  Exemple : "3D Surfaces" pour Macintosh



4. Facettage d'une forme non plane

- Les algorithmes de rendu sont souvent limités aux faces planes (polyèdres)

  Cas de OpenGL pour les sphères et les surfaces paramétrées (Bezier)

  => les surfaces plus complexes doivent être approximés ("tessaleted") par des polyèdres

- Pour la sphère : un algorithme naïf (comme en géographie)

```
procedure drawsphere(xc,yc,zc,r:real);
var r1,r2,h:real;    i,j:integer;    p1,p2,p3,p4:vector;
begin
 r1:=r;h:=0;
 for j:=1 to 4 do begin
     h:=h+r/4;
     r2:=sqrt(r*r-h*h);
     for i:=0 to 31 do begin
         p1[1]:=r1*cos32[i]+xc;     p1[2]:=r1*sin32[i]+yc;     p1[3]:=zc+h-r/4;
         p2[1]:=r1*cos32[i+1]+xc;   p2[2]:=r1*sin32[i+1]+yc;   p2[3]:=zc+h-r/4;
         p3[1]:=r2*cos32[i]+xc;     p3[2]:=r2*sin32[i]+yc;     p3[3]:=zc+h;
         p4[1]:=r2*cos32[i+1]+xc;   p4[2]:=r2*sin32[i+1]+yc;   p4[3]:=zc+h;

         line3d(p1,p2);line3d(p2,p4);line3d(p4,p3);line3d(p3,p1);

         p1[3]:=zc-h+r/4;   p2[3]:=zc-h+r/4;   p3[3]:=zc-h;   p4[3]:=zc-h;
         line3d(p1,p2);line3d(p2,p4);line3d(p4,p3);line3d(p3,p1);
     end;
     r1:=r2
     end
end;
```

- Un algorithme plus efficace : Jon Leech (1989)

ftp://wuarchive.wustl.edu/graphics/graphics/misc/sphere-tesselate/sphere.c.Z

- partir d'un polyèdre régulier (octaèdre)

- découper chaque facette triangulaire en 4 triangles plus petits

- continuer la récursion jusqu'au niveau voulu

```
/* Subdivide each triangle in the old approximation and normalize
*  the new points thus generated to lie on the surface of the unit
*  sphere.
* Each input triangle with vertices labelled [0,1,2] as shown
*  below will be turned into four new triangles:
*
*                       Make new points
*                           a = (0+2)/2
*                           b = (0+1)/2
*                           c = (1+2)/2
*         1
*        /\             Normalize a, b, c
*       /  \
*    b/____\ c          Construct new triangles
*    /\    /\               [0,b,a]
*   /  \  /  \              [b,1,c]
*  /____\/____\             [a,b,c]
* 0      a     2            [a,c,2]
*/
```

On peut adapter le source pour la génération dans un autre format

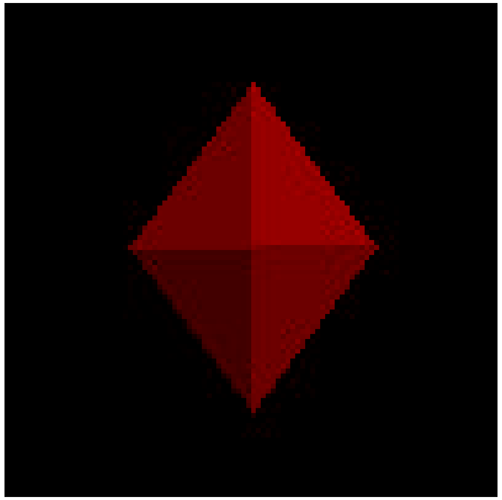
Résultat avec 32 faces en source POV :

```
camera {
  location  <0.0 , 0.0 ,-5.0>
  look_at   <0.0 , 0.0 , 0.0>
}
polygon {32, //nombre de points
// chaque ligne = un triangle
<1, 0, 0>, <0, 0, 1>, <0, 1, 0>, <1, 0, 0>
<0, 1, 0>, <0, 0, 1>, <-1, 0, 0>, <0, 1, 0>
<-1, 0, 0>, <0, 0, 1>, <0, -1, 0>, <-1, 0, 0>
<0, -1, 0>, <0, 0, 1>, <1, 0, 0>, <0, -1, 0>
<1, 0, 0>, <0, 1, 0>, <0, 0, -1>, <1, 0, 0>
<0, 1, 0>, <-1, 0, 0>, <0, 0, -1>, <0, 1, 0>
<-1, 0, 0>, <0, -1, 0>, <0, 0, -1>, <-1, 0, 0>
<0, -1, 0>, <1, 0, 0>, <0, 0, -1>, <0, -1, 0>
color red 1
}
```

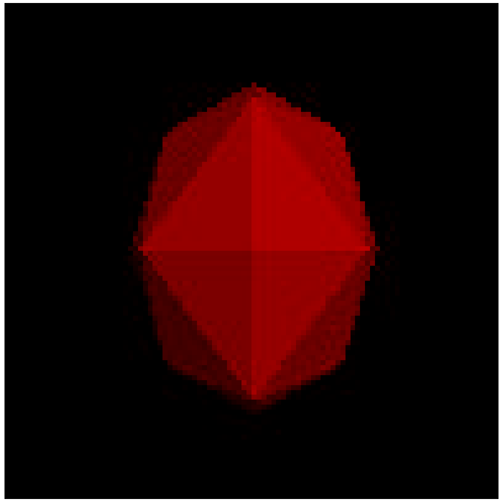
Idem en VRML :

```
#VRML V1.0 ascii
Separator{
    DirectionalLight { direction 0 0 -10}
    Separator {
        Material {diffuseColor 1 0 0  #Rouge}
        Coordinate3 { point [
       1 0 0 , 0 0 1 ,0 1 0 ,0 1 0 ,0 0 1 ,-1 0 0 ,-1 0 0 ,
       0 0 1 , 0 -1 0 , 0 -1 0 ,0 0 1 ,1 0 0 ,1 0 0 ,0 1 0 ,
       0 0 -1 ,0 1 0 ,-1 0 0 ,0 0 -1 ,-1 0 0 ,0 -1 0 ,0 0 -1,
       0 -1 0 ,1 0 0 ,0 0 -1 ] }
  IndexedFaceSet { coordIndex [
       0,1,2,-1, 3,4,5,-1, 6,7,8,-1, 9,10,11,-1,12,13,14,-1,
       15,16,17,-1,18,19,20,-1, 21,22,23,-1 ]}
}
```

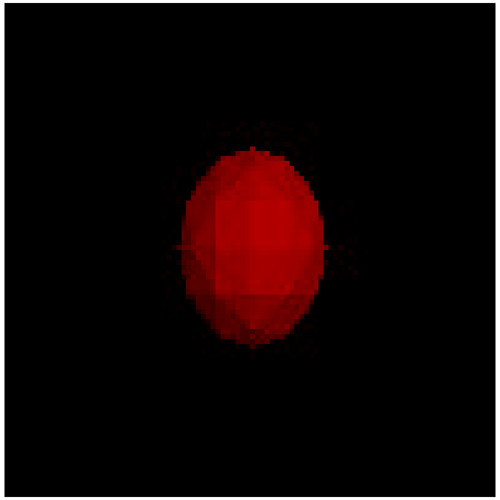
- Résultats avec POV



8 faces



32 faces

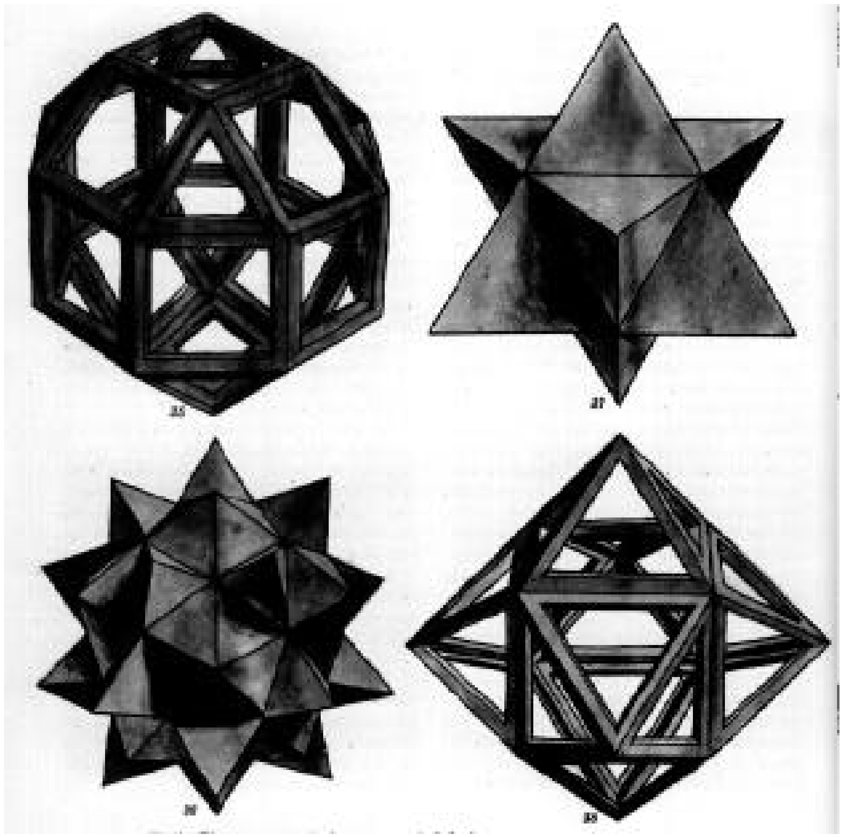


128 faces

5. Exercices

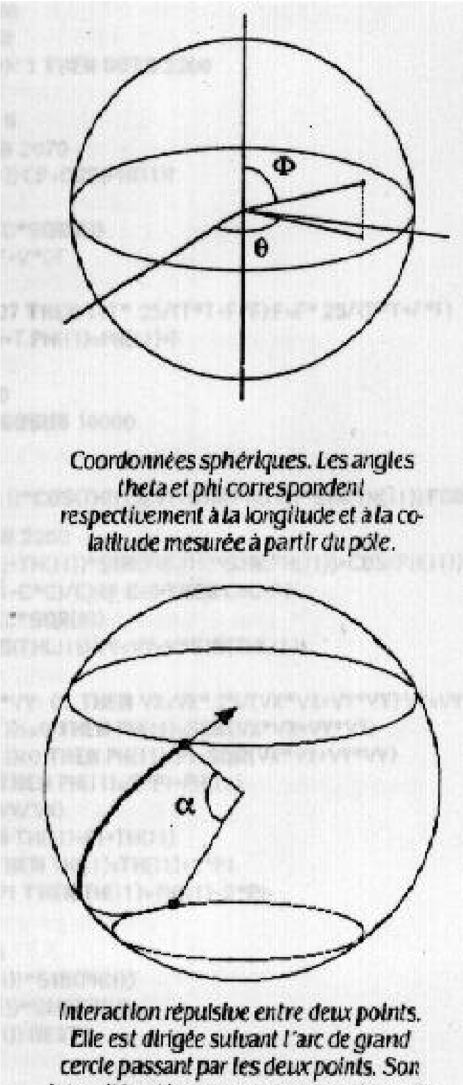
- Calculer les coordonnées des 5 polyèdres réguliers (platoniques), des polyèdres archimédiens, des solides de Johnson…

  source : http://netlib.att.com/netlib/polyhedra/index.html



attribués à Léonard de Vinci - in C.A. Reichen "Histoire de la physique" Ed. Rencontre, 1963.

- Appliquer une autre technique de triangulation de la sphère



- on pose N points au hasard sur la sphère
- ceux-ci se "répulsent" selon une loi en 1 / d^2
- on arrête quand le critère de convergence est vérifié

source :

"Sphere FAQ" du groupe sci.math

`http://www.math.niu.edu/~rusin/papers/known-math/spheres/`

<= également :

F. NEUVILLE "Le polisseur de boules"

Science et Vie Micro n°39, Mai 1987, pp. 83-87

- Modèles de cristaux

source : LAPPARENT "Précis de minéralogie", Blanchard, 1965 (Nouveau tirage)

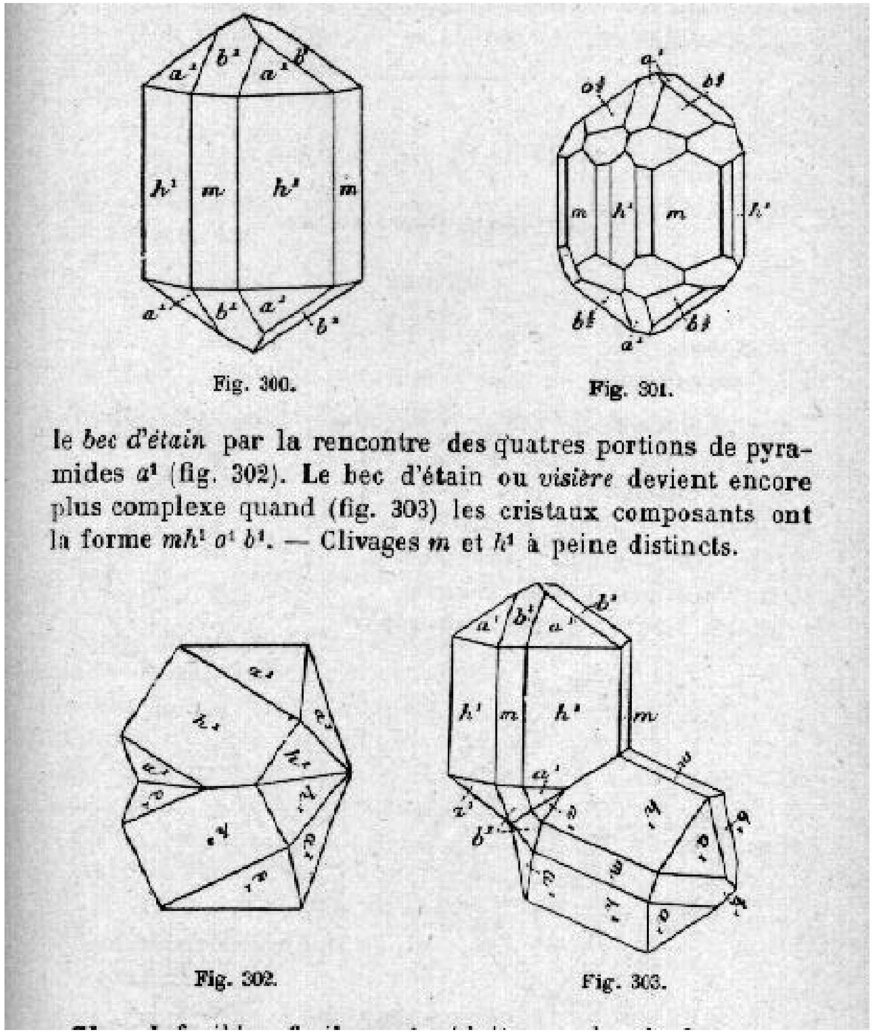


Fig. 300.

Fig. 301.

le *bec d'étain* par la rencontre des quatres portions de pyramides a^1 (fig. 302). Le bec d'étain ou *visière* devient encore plus complexe quand (fig. 303) les cristaux composants ont la forme $mh^1 a^1 b^1$. — Clivages m et h^1 à peine distincts.

Fig. 302.

Fig. 303.

Lapparent, op. cit., p. 301