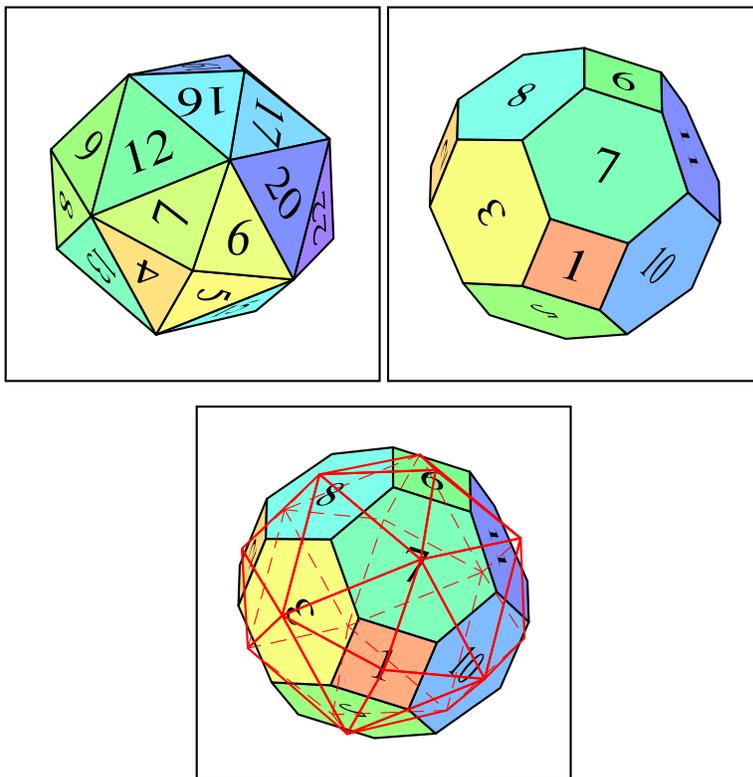


Essai d'élaboration d'une méthode de construction des géodes duales

17 janvier 2008



1 Construction d'une géode

Elle ne pose pas de sérieux problème, voir le fichier `pst-geode` et les références contenues dans la documentation.

<http://melusine.eu.org/syracuse/pstricks/pst-solides3d/#geode>

L'inconvénient c'est que si l'on prend une géode à un degré donné de récursion, puisque chaque face du polyèdre initial est divisée en quatre triangles dont on calcule les sommets, les sommets ainsi calculés sont comptés plusieurs fois et numérotés dans l'ordre obtenu. Ce n'est pas gênant pour les facettes triangulaires ainsi obtenues qui sont bien repérées et orientées. Par contre, cela pose un problème, comme nous allons le voir, pour la construction de la géode duale.

2 Construction de la géode duale

Le principe est bien expliqué sur :

<http://fr.wikipedia.org/wiki/G%C3%A9ode>

- prendre un sommet de la géode obtenue à un niveau de récursion donné ;
- déterminer tous les triangles ayant ce sommet en commun ;
- déterminer les isobarycentres de ces triangles ;
- les projeter sur la sphère ;
- tracer le polygone ayant pour sommets les points ainsi obtenus.

En prenant l'exemple de la géode du premier dessin, on voit qu'un sommet est commun à 4 ou 6 facettes.

Dans un premier temps il faut déterminer tous les sommets identiques, les re-numéroter et faire la correction correspondante des numéros des sommets dans la liste faces de la géode.

J'utilise IdePS : Integrated Development Environment For PostScript pour mettre au point le code PostScript.

<http://www.files-library.com/files/wxGhostsript-class-PS-IDE-debugger.html>

Par exemple pour sortir les faces et les sommets de la géode au premier niveau de récursion :

```
% les coordonnées des 6 sommets de l'octaèdre
/S [ 0 0 1
    1 0 0
    0 1 0
    -1 0 0
    0 -1 0
    0 0 -1]
def
% les 8 faces de l'octaèdre
/F [
  [1 5 2]
  [2 5 3]
  [3 5 4]
  [4 5 1]
  [1 0 4]
  [4 0 3]
  [3 0 2]
  [2 0 1]
] def
0 1 0 {
/Ft [] def
/St [] def
/iF 0 def
% Sortir les faces une à une
0 1 F length 1 sub { /i exch def
  /Fi F i get def
  /I {Fi 0 get 3 mul} bind def % 1er sommet de la face i
% les coordonnées du sommet 0
/x0 S I get def
/y0 S I 1 add get def
/z0 S I 2 add get def

  /I {Fi 1 get 3 mul} bind def % 2eme sommet de la face i
```

```

% les coordonnées du sommet 1
/x1 S I get def
/y1 S I 1 add get def
/z1 S I 2 add get def

/I {Fi 2 get 3 mul} bind def % 3eme sommet de la face i
% les coordonnées du sommet 2
/x2 S I get def
/y2 S I 1 add get def
/z2 S I 2 add get def

% norme des vecteurs
/N12 {x2 x1 sub dup mul
      y2 y1 sub dup mul
      z2 z1 sub dup mul
      add add sqrt} def
/N01 {x0 x1 sub dup mul
      y0 y1 sub dup mul
      z0 z1 sub dup mul
      add add sqrt} def
/N02 {x2 x0 sub dup mul
      y2 y0 sub dup mul
      z2 z0 sub dup mul
      add add sqrt} def
% les coordonnées des milieux des côtés
/X0 {x1 x2 add N12 div} def
/Y0 {y1 y2 add N12 div} def
/Z0 {z1 z2 add N12 div} def

/X1 {x0 x2 add N02 div} def
/Y1 {y0 y2 add N02 div} def
/Z1 {z0 z2 add N02 div} def

/X2 {x0 x1 add N01 div} def
/Y2 {y0 y1 add N01 div} def
/Z2 {z0 z1 add N01 div} def

% On en déduit les nouveaux sommets
/Sommets [St aload pop x0 y0 z0 x1 y1 z1 x2 y2 z2 X0 Y0 Z0 X1 Y1 Z1 X2 Y2 Z2] def
/St Sommets def
/Faces [Ft aload pop [0 iF add 5 iF add 4 iF add]
        [1 iF add 3 iF add 5 iF add]
        [2 iF add 4 iF add 3 iF add]
        [3 iF add 4 iF add 5 iF add]
        ] def
/Ft Faces def
/iF iF 6 add def
} for
/S Sommets def

```

```

/F Faces def
}for
Faces
Sommets
pstack

```

Je récupère la liste de 48 sommets et 32 faces :

```

[
  1 0 0 % sommet 0
  0 0 -1 % 1
  0 1 0 % 2
  0.0 0.707106769 -0.707106769 % 3
  0.707106769 0.707106769 0.0 % 4
  0.707106769 0.0 -0.707106769 % 5
  0 1 0 % 6
  0 0 -1 % 7
  -1 0 0 % 8
  -0.707106769 0.0 -0.707106769 % 9
  -0.707106769 0.707106769 0.0 % 10
  0.0 0.707106769 -0.707106769 % 11
  -1 0 0 % 12
  0 0 -1 % 13
  0 -1 0 % 14
  0.0 -0.707106769 -0.707106769 % 15
  -0.707106769 -0.707106769 0.0 % 16
  -0.707106769 0.0 -0.707106769 % 17
  0 -1 0 % 18
  0 0 -1 % 19
  1 0 0 % 20
  0.707106769 0.0 -0.707106769 % 21
  0.707106769 -0.707106769 0.0 % 22
  0.0 -0.707106769 -0.707106769 % 23
  1 0 0 % 24
  0 0 1 % 25
  0 -1 0 % 26
  0.0 -0.707106769 0.707106769 % 27
  0.707106769 -0.707106769 0.0 % 28
  0.707106769 0.0 0.707106769 % 29
  0 -1 0 % 30
  0 0 1 % 31
  -1 0 0 % 32
  -0.707106769 0.0 0.707106769 % 33
  -0.707106769 -0.707106769 0.0 % 34
  0.0 -0.707106769 0.707106769 % 35
  -1 0 0 % 36
  0 0 1 % 37
  0 1 0 % 38
  0.0 0.707106769 0.707106769 % 39
  -0.707106769 0.707106769 0.0 % 40

```

```

-0.707106769 0.0 0.707106769 % 41
0 1 0 % 42
0 0 1 % 43
1 0 0 % 44
0.707106769 0.0 0.707106769 % 45
0.707106769 0.707106769 0.0 % 46
0.0 0.707106769 0.707106769 % 47
]

```

[% les faces

```

[0 5 4]
[1 3 5]
[2 4 3]
[3 4 5]
[6 11 10]
[7 9 11]
[8 10 9]
[9 10 11]
[12 17 16]
[13 15 17]
[14 16 15]
[15 16 17]
[18 23 22]
[19 21 23]
[20 22 21]
[21 22 23]
[24 29 28]
[25 27 29]
[26 28 27]
[27 28 29]
[30 35 34]
[31 33 35]
[32 34 33]
[33 34 35]
[36 41 40]
[37 39 41]
[38 40 39]
[39 40 41]
[42 47 46]
[43 45 47]
[44 46 45]
[45 46 47]

```

]

On peut remarquer que le sommet '0' se retrouve 4 fois, c'est le sommet commun à quatre triangles.

Dans une première étape, il faut donc supprimer les sommets identiques pour n'en laisser qu'un seul et rénumérotter. En même temps il faut mémoriser les correspondances :

sommet 0 = 20 = 24 = 44

pour dans la liste des faces, remplacer les sommets 20, 24, 44 par 0 etc.

J'ai réalisé ces deux étapes à la "main" : *ce n'est pas bien !*.

1 0 0	% 0
0 0 -1	% 1
0 1 0	% 2
0 0.70710679 -0.70710679	% 3
0.70710679 0.70710679 0	% 4
0.70710679 0 -0.70710679	% 5
-1 0 0	% 6
-0.70710679 0 -0.70710679	% 7
-0.70710679 0.70710679 0	% 8
0 -1 0	% 9
0 -0.70710679 -0.70710679	% 10
-0.70710679 -0.70710679 0	% 11
0.70710679 -0.70710679 0	% 12
0 0 1	% 13
0 -0.70710679 0.70710679	% 14
0.70710679 0 0.70710679	% 15
-0.70710679 0 0.70710679	% 16
0 0.70710679 0.70710679	% 17

Dans une deuxième étape, on réorganise les sommets des 32 faces en regroupant toutes les faces qui tournent autour d'un même sommet, dans le sens trigonométrique :

[0 5 4]	% face 0
[0 4 15]	% face 1
[0 15 12]	% face 2
[0 12 5]	% face 3
[1 3 5]	% face 4
[1 5 10]	% face 5
[1 10 7]	% face 6
[1 7 3]	% face 7
[2 4 3]	% face 8
[2 3 8]	% face 9
[2 8 17]	% face 10
[2 17 4]	% face 11
[3 7 8]	% face 12
[3 4 5]	% face 13
[4 17 15]	% face 14
[5 12 10]	% face 15
[6 8 7]	% face 16
[6 7 11]	% face 17
[6 11 16]	% face 18
[6 16 8]	% face 19
[7 10 11]	% face 20
[8 16 17]	% face 21
[9 11 10]	% face 22
[9 10 12]	% face 23
[9 12 14]	% face 24
[9 14 11]	% face 25

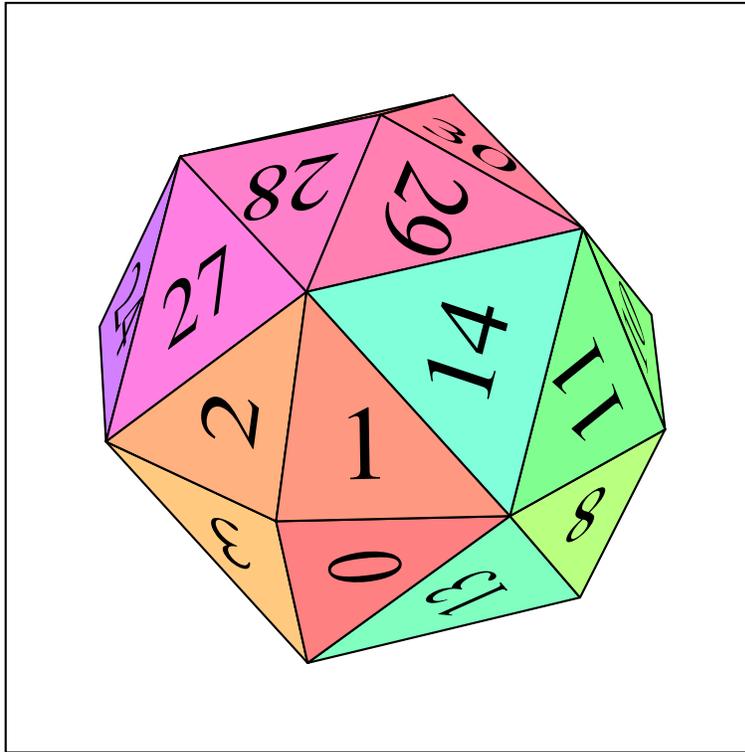
```

[11 14 16] % face 26
[12 15 14] % face 27
[13 14 15] % face 28
[13 15 17] % face 29
[13 17 16] % face 30
[13 16 14] % face 31

```

On peut constater que chacun des sommets se trouve impliqué suivant cas dans 4 ou 6 faces. Par exemple 0 est sommet commun de 4 faces, tout comme 1,2,6,9 et 13, par contre tous les autres sont "sommet commun" à 6 faces.

On va donc réécrire la liste des faces en partant d'un sommet et en écrivant et en regroupant toutes les facettes ayant ce sommet en commun. Difficulté supplémentaire : les facettes doivent être ordonnées dans le sens trigonométrique en regardant de ce sommet commun !



```

/FACES [
  [
    [0 5 4]
    [0 4 15]
    [0 15 12]
    [0 12 5]
  ]
  [
    [1 3 5]
    [1 5 10]
    [1 10 7]
  ]

```

```

    ]
    [
        [1 7 3]
        [2 4 3]
        [2 3 8]
        [2 8 17]
        [2 17 4]
    ]
    [
        [3 1 7]
        [3 7 8]
        [3 8 2]
        [3 2 4]
        [3 4 5]
        [3 5 1]
    ]
    [
        [4 0 5]
        [4 5 3]
        [4 3 2]
        [4 2 17]
        [4 17 15]
        [4 15 0]
    ]
    [
        [5 1 3]
        [5 3 4]
        [5 4 0]
        [5 0 12]
        [5 12 10]
        [5 10 1]
    ]
    [
        [6 8 7]
        [6 7 11]
        [6 11 16]
        [6 16 8]
    ]
    [
        [7 3 1]
        [7 1 10]
        [7 10 11]
        [7 11 6]
        [7 6 8]
        [7 8 3]
    ]
    [
        [8 2 3]
        [8 3 7]
    ]

```

```
    [8 7 6]
    [8 6 16]
    [8 16 17]
    [8 17 2]
  ]
  [
    [9 11 10]
    [9 10 12]
    [9 12 14]
    [9 14 11]
  ]
  [
    [10 1 5]
    [10 5 12]
    [10 12 9]
    [10 9 11]
    [10 11 7]
    [10 7 1]
  ]
  [
    [11 6 7]
    [11 7 10]
    [11 10 9]
    [11 9 14]
    [11 14 16]
    [11 16 6]
  ]
  [
    [12 9 10]
    [12 10 5]
    [12 5 0]
    [12 0 15]
    [12 15 14]
    [12 14 9]
  ]
  [
    [13 14 15]
    [13 15 17]
    [13 17 16]
    [13 16 14]
  ]
  [
    [14 15 13]
    [14 13 16]
    [14 16 11]
    [14 11 9]
    [14 9 12]
    [14 12 15]
  ]
]
```

```

[
  [15 12 0]
  [15 0 4]
  [15 4 17]
  [15 17 13]
  [15 13 14]
  [15 14 12]
]
[
  [16 14 13]
  [16 13 17]
  [16 17 8]
  [16 8 6]
  [16 6 11]
  [16 11 14]
]
[
  [17 8 16]
  [17 16 13]
  [17 13 15]
  [17 15 4]
  [17 4 2]
  [17 2 8]
]
]

```

Pourquoi faire? demanderez-vous si vous avez oublié le problème posé : déterminer les isobary-centres des facettes entourant un sommet, puis projeter ce centre sur la sphère et enfin tracer le poly-gone, carré ou hexagone suivant le sommet !

Cette partie-là, la plus facile, je l'ai réalisée avec PostScript, à partir de la liste des faces et sommets précédents :

```

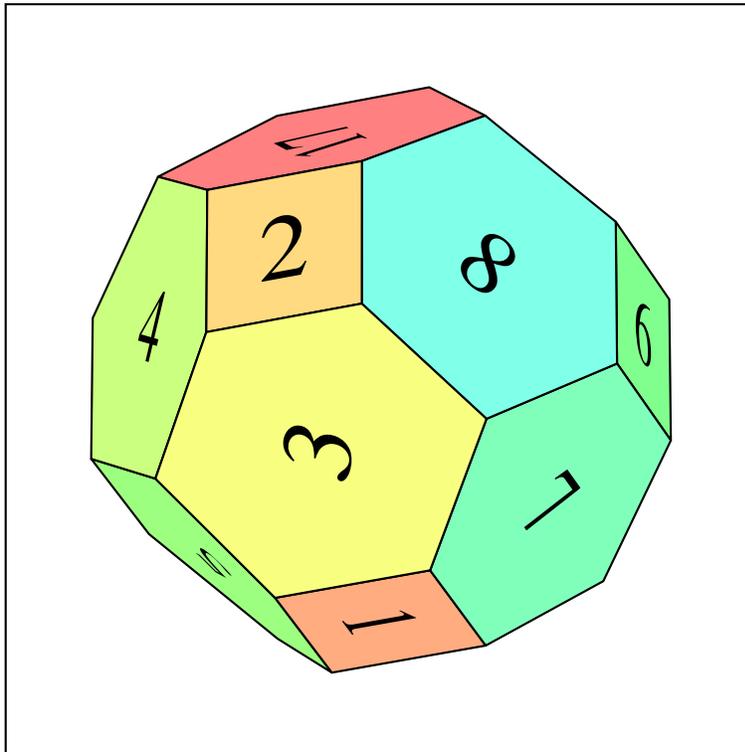
/j 0 def
/CoordTemp [] def
FACES length {
/F' FACES j get def
/i 0 def
F' length {
/FO F' i get def
/NumeroSommet {FO 0 get 3 mul} bind def % 1er sommet de la face 0
/x0 S' NumeroSommet get def
/y0 S' NumeroSommet 1 add get def
/z0 S' NumeroSommet 2 add get def
/NumeroSommet {FO 1 get 3 mul} bind def % 2eme sommet de la face 0
/x1 S' NumeroSommet get def
/y1 S' NumeroSommet 1 add get def
/z1 S' NumeroSommet 2 add get def
/NumeroSommet {FO 2 get 3 mul} bind def % 3eme sommet de la face 0
/x2 S' NumeroSommet get def
/y2 S' NumeroSommet 1 add get def
/z2 S' NumeroSommet 2 add get def

```

```

%
/xG x0 x1 x2 add add 3 div def
/yG y0 y1 y2 add add 3 div def
/zG z0 z1 z2 add add 3 div def
% norme
/norme xG dup mul yG dup mul zG dup mul add add sqrt def
/xG xG norme div def
/yG yG norme div def
/zG zG norme div def
/Coor [CoorTemp aload pop xG yG zG] def
/CoorTemp Coor def
/i i 1 add def
  } repeat
/j j 1 add def
} repeat
/ListeCoor {Coor aload pop} def
/i 0 def
/j 0 def
/T1 [] def
FACES length {
/T2 [] def
/F1 FACES i get def
F1 length {
/F' [T2 aload pop j] def
/FaCes [ T1 aload pop F' ] def
/T2 F' def
/j j 1 add def
} repeat
/i i 1 add def
/T1 FaCes def
}repeat
/listeFaces {FaCes aload pop} def%
} def

```



Ce code permet aussi de récupérer la liste des sommets et des faces. Il est convenable de faire ici aussi le *ménage*, car un même sommet se trouve reproduit plusieurs fois! Le *nettoyage* fait à la main donne :

```
sommets=0.923879504 0.270598054 -0.270598054 % 0
0.923879504 0.270598054 0.270598054 % 1
0.923879504 -0.270598054 0.270598054 % 2
0.923879504 -0.270598054 -0.270598054 % 3
0.270598054 0.270598054 -0.923879504 % 4
0.270598054 -0.270598054 -0.923879504 % 5
-0.270598054 -0.270598054 -0.923879504 % 6
-0.270598054 0.270598054 -0.923879504 % 7
0.270598054 0.923879504 -0.270598054 % 8
-0.270598054 0.923879504 -0.270598054 % 9
-0.270598054 0.923879504 0.270598054 % 10
0.270598054 0.923879504 0.270598054 % 11
-0.577350259 0.577350259 -0.577350259 % 12
0.577350259 0.577350259 -0.577350259 % 13
0.577350259 0.577350259 0.577350259 % 14
0.577350259 -0.577350259 -0.577350259 % 15
-0.923879504 0.270598054 -0.270598054 % 16
-0.923879504 -0.270598054 -0.270598054 % 17
-0.923879504 -0.270598054 0.270598054 % 18
-0.923879504 0.270598054 0.270598054 % 19
-0.577350259 -0.577350259 -0.577350259 % 20
-0.577350259 0.577350259 0.577350259 % 21
```

```

-0.270598054 -0.923879504 -0.270598054 % 22
0.270598054 -0.923879504 -0.270598054 % 23
0.270598054 -0.923879504 0.270598054 % 24
-0.270598054 -0.923879504 0.270598054 % 25
-0.577350259 -0.577350259 0.577350259 % 26
0.577350259 -0.577350259 0.577350259 % 27
0.270598054 -0.270598054 0.923879504 % 28
0.270598054 0.270598054 0.923879504 % 29
-0.270598054 0.270598054 0.923879504 % 30
-0.270598054 0.270598054 0.923879504 % 31
-0.270598054 -0.270598054 0.923879504, % 32
faces={ [0 1 2 3] % 0
[4 5 6 7] % 1
[8 9 10 11] % 2
[7 12 9 8 13 4] % 3
[0 13 8 11 14 1] % 4
[4 13 0 3 15 5] % 5
[16 17 18 19] % 6
[7 6 20 17 16 12] % 7
[9 12 16 19 21 10] % 8
[22 23 24 25] % 9
[5 15 23 22 20 6] % 10
[17 20 22 25 26 18] % 11
[23 15 3 2 27 24] % 12
[28 29 30 32] % 13
[28 32 26 25 24 27] % 14
[2 1 14 29 28 27] % 15
[21 31 29 14 11 10] % 16
[32 31 21 19 18 26] % 17

```

3 Brève conclusion

Ce n'est pas de la tarte! Mais ce n'est peut-être la meilleure des méthodes et une simplification essentielle m'a peut-être échappée?

Donc, si vous avez mis au point une méthode, de préférence en PostScript, pour déterminer automatiquement la duale d'une géode donnée, je suis preneur!