"A Little Printer Exercise on Line Arrangements and Zonotopal Tilings"

By Jürgen Richter-Gebert

1. Introduction

Planar line arrangements as well as *zonotopes* have been studied in the mathematical literature since many years. Many interesting articles have been written and many beautiful pictures have been drawn. In the last decade more and more connections between these two areas of geometry have been worked out. Especially, it turns out that there is a very close connection between planar tiling patterns build from zonotopal tiles and planar line arrangements. As a first instance of this fact in 1980 N.G. deBruijn [B] proved that the well known *Penrose Tilings* can be generated by dualizing certain infinite line arrangements the so called *Pentagrids*.

The full equivalence statement between general finite arrangements and zonotopal tilings in its most general and beautiful extend was proven in 1991 by J. Bohne and A.D.W. Dress. After defining zonotopal tilings carefully, the 2-dimensional case can be roughly stated as follows [BD], [RZ]:

> *The (pseudo-)line arrangements in the euclidean plane are in bijection with the (combinatorial types of) zonotopal tilings of regular* $2n$*-gons.*

The correct version of this theorem deals with *pseudoline arrangements* and *combinatorial types of tilings* rather then line arrangements and concrete zonotopes. This is the case since the main bridge between these two structures uses their shared combinatorial content of *oriented matroids* [BLSWZ] (a theory that nicely encodes the combinatorial properties of oriented (pseudo-) hyperplane arrangements).

In this little article we want to show how the passage from straight line arrangements via their combinatorial content to their corresponding zonotopal tiling can be most easily described. Moreover, we describe how this methods can be used to directly implement a short PostScript program that draws the line arrangement and the corresponding tiling directly from the metric line data. The purpose of this article is by no means mathematical novelty, but mathematical beauty and easiness.

2. Mathematical background

A planar line arrangement $\mathcal{L} = \{l_1 \ldots, l_n\}$ on n lines can be easily described by a collection of n homogeneous coordinates $l_i := (x_i, y_i, z_i)$ with $1 \leq i \leq n$. If we assume that $x_i^2 + y_i^2 = 1$ then l_i is a line with direction orthogonal to the vector (x_i, y_i) located at an (oriented) distance z_i from the origin. Since (x, y, z) describes the same line as $(-x, -y, -z)$, we can furthermore assume that the z coordinates are always non-negative. The line arrangement dissects the plane into regions and thereby generates a certain cell complex $\Delta\mathcal{L}$ consisting of faces, line segments and points together with the corresponding inclusion relation. The polar $\Delta\mathcal{L}^*$ of $\Delta\mathcal{L}$ is a cell complex where the faces, line segments and points of $\Delta\mathcal{L}$ are mapped to the points, line segments and faces of $\Delta\mathcal{L}^*$, respectively. The inclusion relations have to be reversed under this mapping. One possible realization of this polar cell complex is given by the construction of the zonotopal tiling associated to $\mathcal{L}$ as described below.

Our definition of lines by their homogeneous coordinates implicitly induces an orientation on each line. If the z coordinate of a line $l = (x, y, z)$ was chosen to be non-negative, then we define the *positive side* of l as the side where the vector (x, y) points to. Considering our lines in $\mathcal{L}$ oriented, we can assign to each point p in the plane a vector consisting of n signs "+", "−" or "0" describing for every line l_i whether the point lies on the left on the right or on it. Let $p := (x, y)$ be a point and let $p' := (x, y, 1)$ be homogeneous coordinates for this point. The corresponding sign-vector, describing the relative position of the point p with respect to the lines in $\mathcal{L}$ can be easily obtained by:

$$\sigma(p, \mathcal{L}) := \big(\text{sign}(l_1 \cdot p'), \text{sign}(l_2 \cdot p'), \dots, \text{sign}(l_n \cdot p')\big).$$

Here (l_i, p') denotes the standard inner product in $\mathbb{R}^3$. Notice that different points in the same cell of $\Delta\mathcal{L}$ have identical sign-vectors. Therefore the collection of all such sign vectors

$$C(\mathcal{L}) := \big\{\sigma(p, \mathcal{L}) \mid p \in \mathbb{R}^2\big\}$$

can be used as a combinatorial description of $\Delta\mathcal{L}$. Using this fact we can consider the elements in $C(\mathcal{L})$ directly as the labels for the cells in the cell complex $\Delta\mathcal{L}$. This collection of sign-vectors is also called the *oriented matroid* of $\mathcal{L}$. A face $c \in C(\mathcal{L})$ is contained in a face $c' \in C(\mathcal{L})$ if and only if the zero-positions of c are a subset of the zero-positions of c'.

Now we define what we mean by *zonotopes* and *zonotopal tilings.* The *Minkowski Sum* $A + B$ of two convex sets A and B in $\mathbb{R}^d$ is defined by

$$A + B := \{a + b \mid a \in A, b \in B\}.$$

For $d < 0$ a d-dimensional zonotope can be defined as the Minkowski Sum of finitely many line segments spanning $\mathbb{R}^d$. A 0-dimensional zonotope is simply a point. In the plane, besides the points and the line segments themself, centrally symmetric $2n$-gons are the only possible zonotopes. A zonotopal tiling $\mathcal{Z}$ of a zonotope Z is a collection of zonotopes (the tiles) covering Z, with the additional property, that the faces of a tile lie again in $\mathcal{Z}$ and that the intersection of two tiles (if non-empty) is again in $\mathcal{Z}$.

We now can describe directly the zonotopal tiling associated to a given line arrangement $\mathcal{L} = \{l_1 \dots, l_n\}$. For a line $l = (x, y, z)$ we abbreviate its normal vector by $l^* := (x, y)$. By $[a, b]$ we denote the line segment from $a \in \mathbb{R}^2$ to $b \in \mathbb{R}^2$. For a given sign-vector $S \in \{+, -, 0\}$ we define an associated zonotope by:

$$Z(S) := \sum_{X_i=0} [-l_i^*, +l_i^*] + \sum_{X_i=+} l_i^* - \sum_{X_i=-} l_i^*.$$

Especially, $\widehat{Z} := Z((0, 0, \dots, 0))$ is the biggest zonotope achievable that way. For all other sign-vectors we get zonotopes that are contained in $\widehat{Z}$. The zero entries of S describe the shape of $Z(S)$, and the non-zero entries describe how $Z(S)$ is translated with respect to the origin.

Finally, we get the zonotopal tiling $\mathcal{Z}(\mathcal{L})$ of $\widehat{Z}$ associated with $\mathcal{L}$ simply by taking all zonotopes associated to the sign-vectors in $C(\mathcal{L})$:

$$\mathcal{Z}(\mathcal{L}) := \big\{Z(S) \mid S \in C(\mathcal{L})\big\}.$$

3. The program

The last two pages list of this article a simple PostScript program that is capable of drawing the line arrangements and the associated zonotopal tilings from the given line data. In principle it should run on every computer equipment where the computer or the printer is able to understand usual PostScript code (Most Sun workstations, Macintoshes, systems with PostScript-printers, all NeXTs). Simply type in the code, store it to a file and send it to a PostScript printer. On sun workstations you can for instance preview the file by typing: `pageview <filename>`.

The program makes use of the mathematical backgrounds described above. It defines special functions to calculate *signs, determinants, sign-vectors,* etc. A reader familiar with the basics of PostScript may easily decode its function.

Readers who want to play with the program and generate other arrangements and tilings, have to modify the line data starting at line 17 of the printed code. The line data lists simply a sequence of homogeneous coordinates describing the positions of the lines. It is recommended to define the slope of a line by its angle using the `sin` and `cos` operators of PostScript and to add as third coordinate the (positive or negative) distance from the origin (as done in the example). You must make sure that the slopes of the lines are sorted in increasing order and lie between 0 and 180 degree. *Now have fun and play !*

4. The examples

Clearly, there are infinitely many line arrangement and still infinitely many of them generate interesting and nicely looking zonotopal tilings. Therefore it is difficult to restrict yourself to only a few interesting favorite candidates. The selected ones have some relation to different fields of geometric research. The first example shows a finite *Penrose pattern* generated by five bundles of parallel lines. The second example is related to *cyclic polytopes* and shows how very structured portions of the tiling can mutate to almost chaotical structures. The third example is the line arrangement 21_4 with exactly four 4-valent vertices on each of the lines. The last two are taken from a list of simple line arrangements given by B. Grünbaum [G]. These two arrangements share the property that in the arrangement each face is a triangle. Observe that this implies that in the tiling every vertex is three-valent.

5. Literature

[BLSWZ] A. Björner, M. Las Vergnas, B. Sturmfels, N. White & G.M. Ziegler "Oriented Matroids", Cambridge University Press, *Encyclopedia of Mathematics and its Applications*, Vol. 46, 1993.

[BD] J. Bohne: "Eine kombinatorische Analyse zonotopaler Raumaufteilungen", Ph.D. Thesis, University of Bielefeld, 1992.

[B] N.D.de Bruijn: "Algebraic Theory of Penrose's Non-Periodic Tiligns of the Plane", *Ned. Akad. Wetensch. Proc. Ser.A*, **43** (1981), 38–66.

[G] B. Grünbaum: "Arrangements of Hyperplanes", *Proceedings of the Second Lousiana Conference on Combinatorics, Graph Theory and Computing*, Lousiana State University, Baton Rouge, 1971, 42–106.

[RZ] J. Richter-Gebert & G.M. Ziegler: "Zonotopal Tilings and the Bohne-Dress Theorem", *in preparation.*

Authors address:

Forschungs- und Technologiezentrum der Telekom, Am Kavalleriesand 3, D-64295 Darmstadt, Gemany

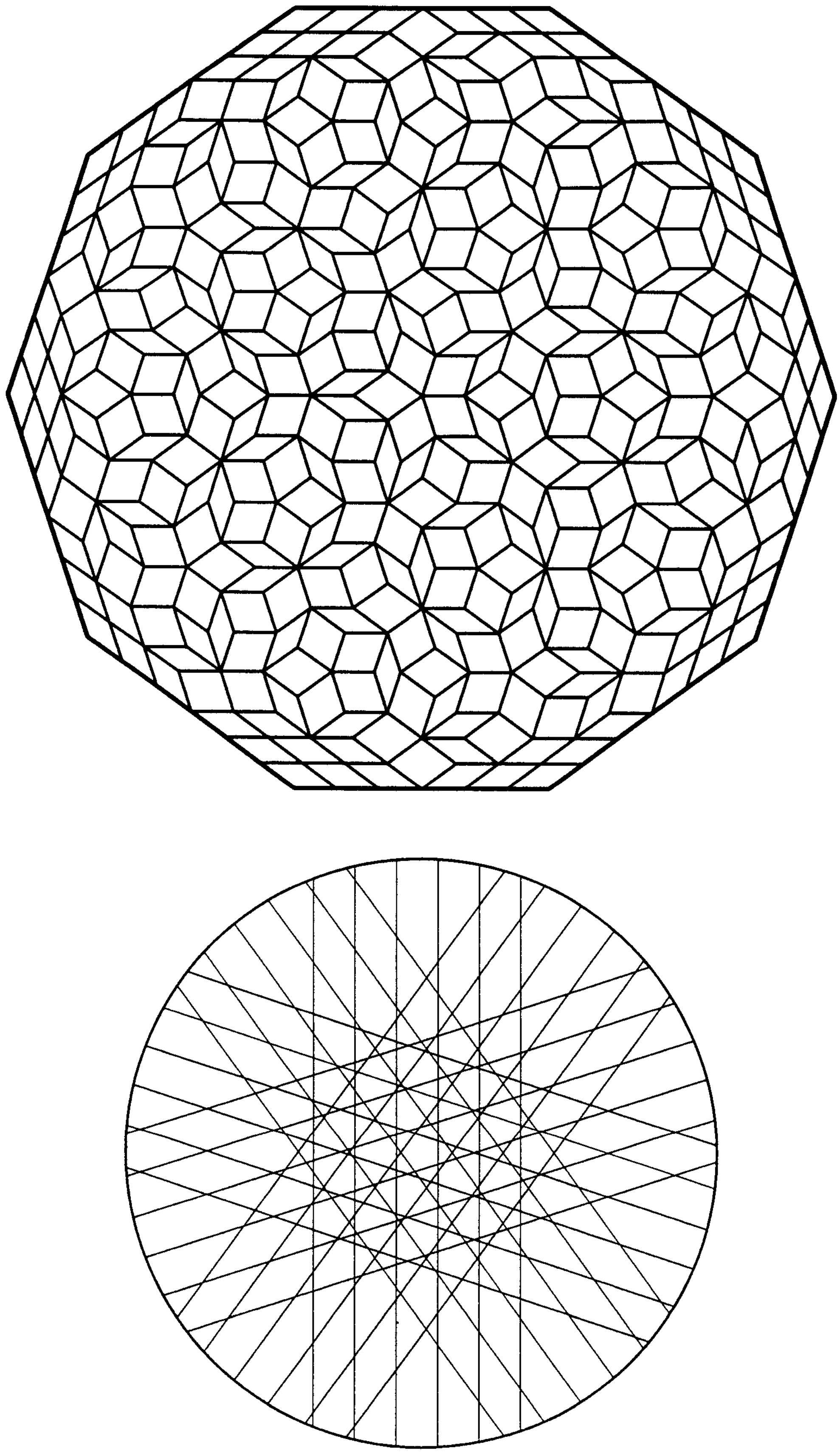

Figure 1. A finite Penrose pattern.

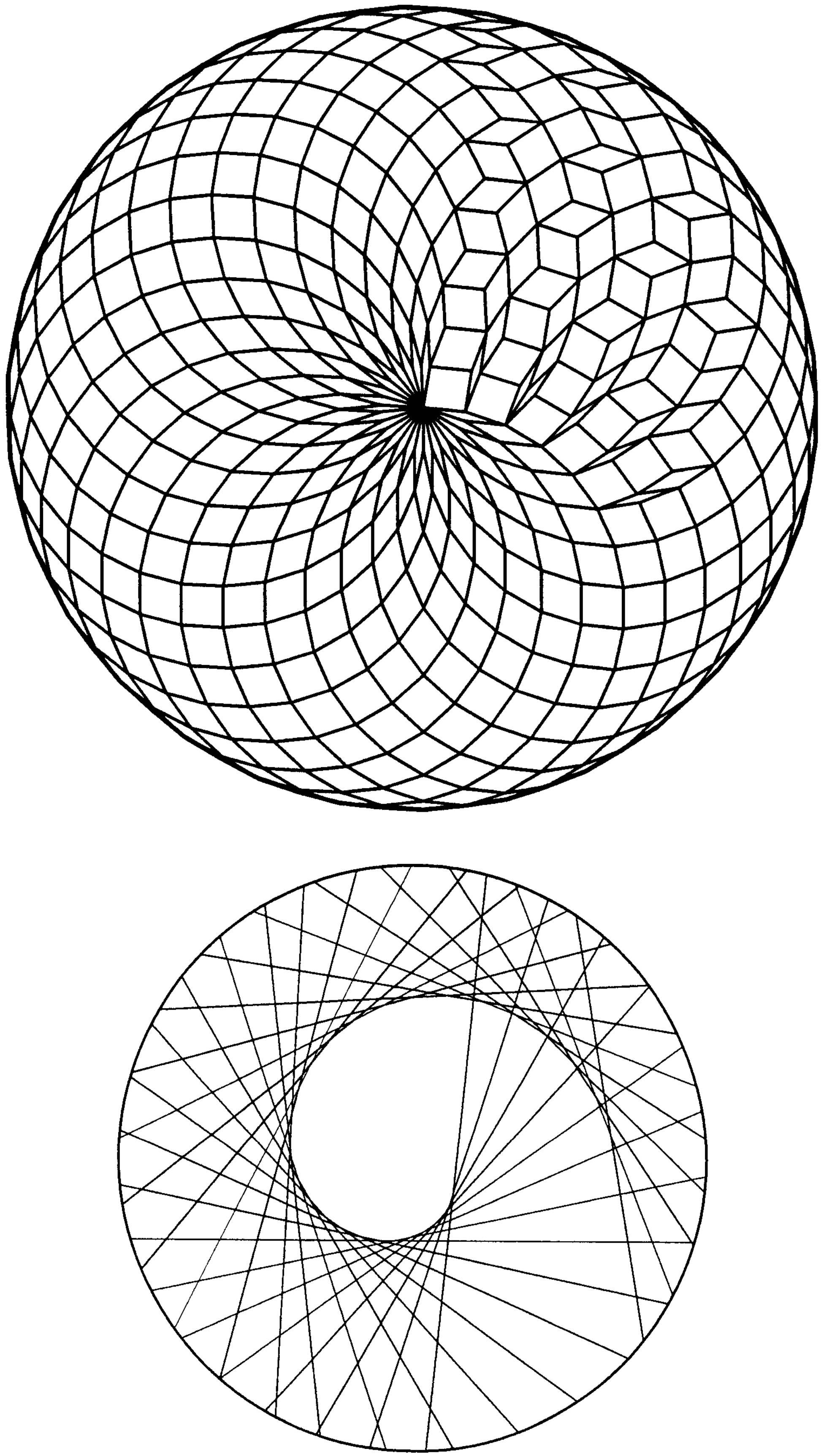

Figure 2. A “linear line spiral”.

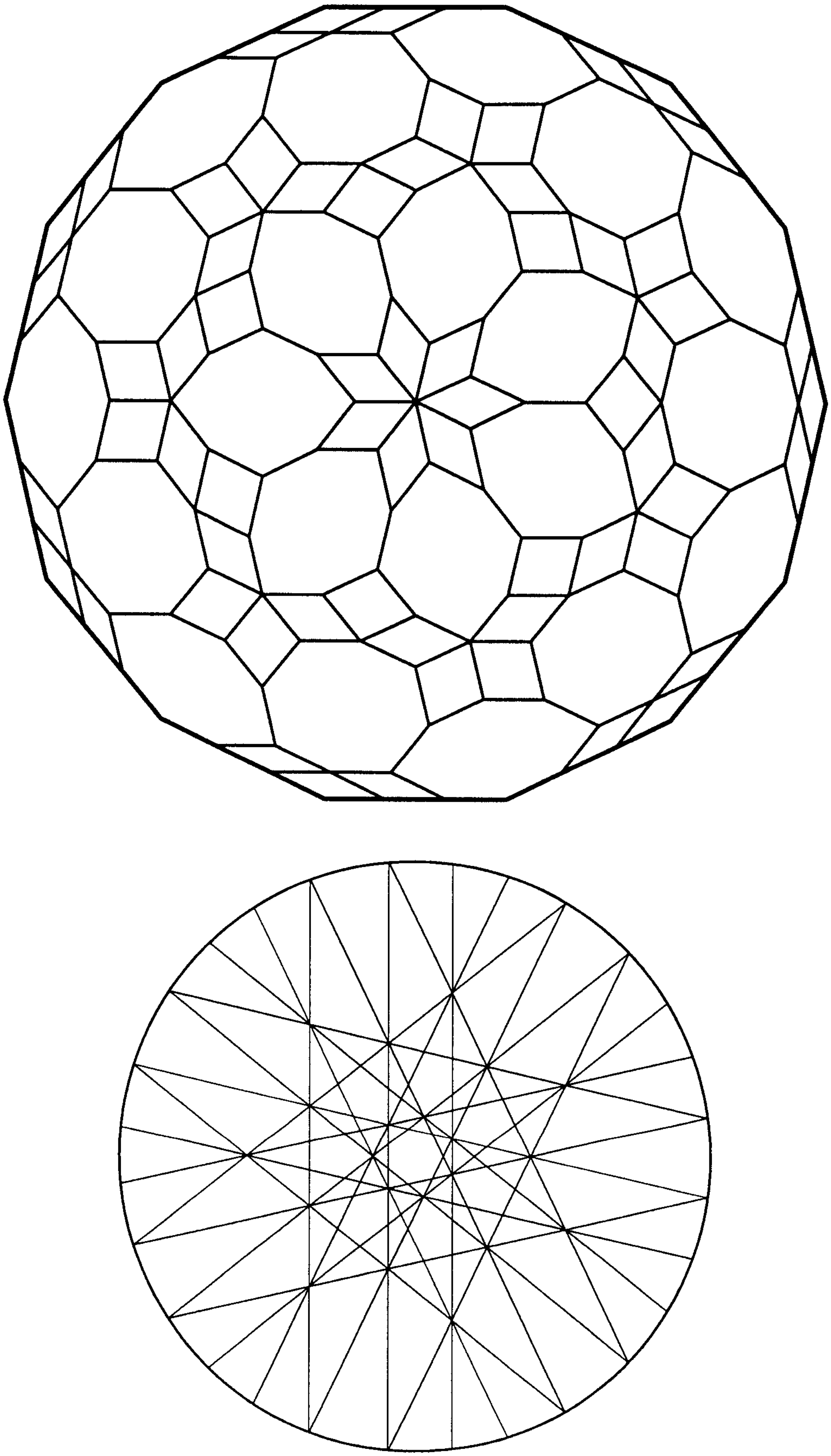

Figure 3. The line arrangement 21_4.

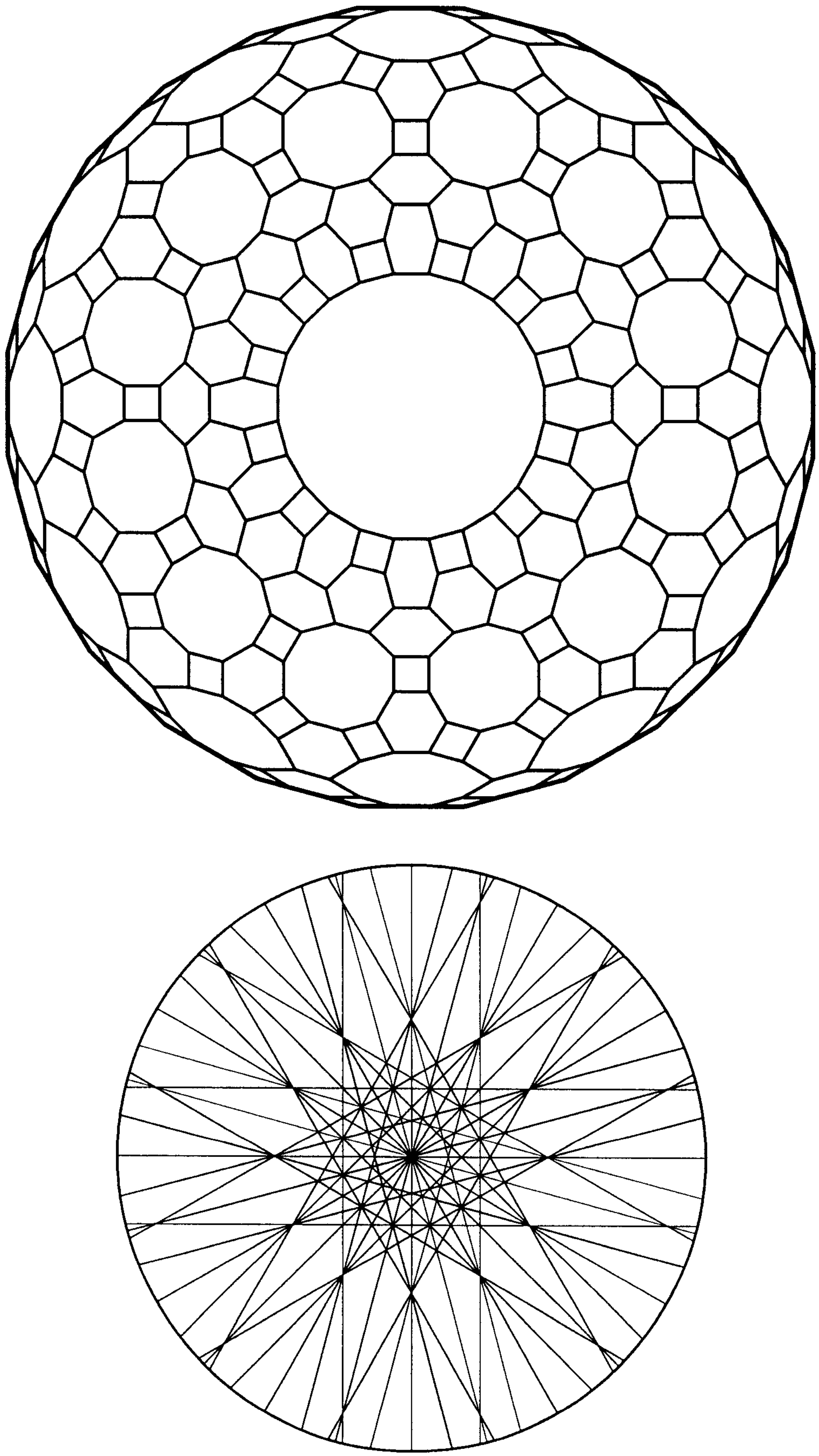

Figure 4. The simple line arrangement $A37_2^*$.

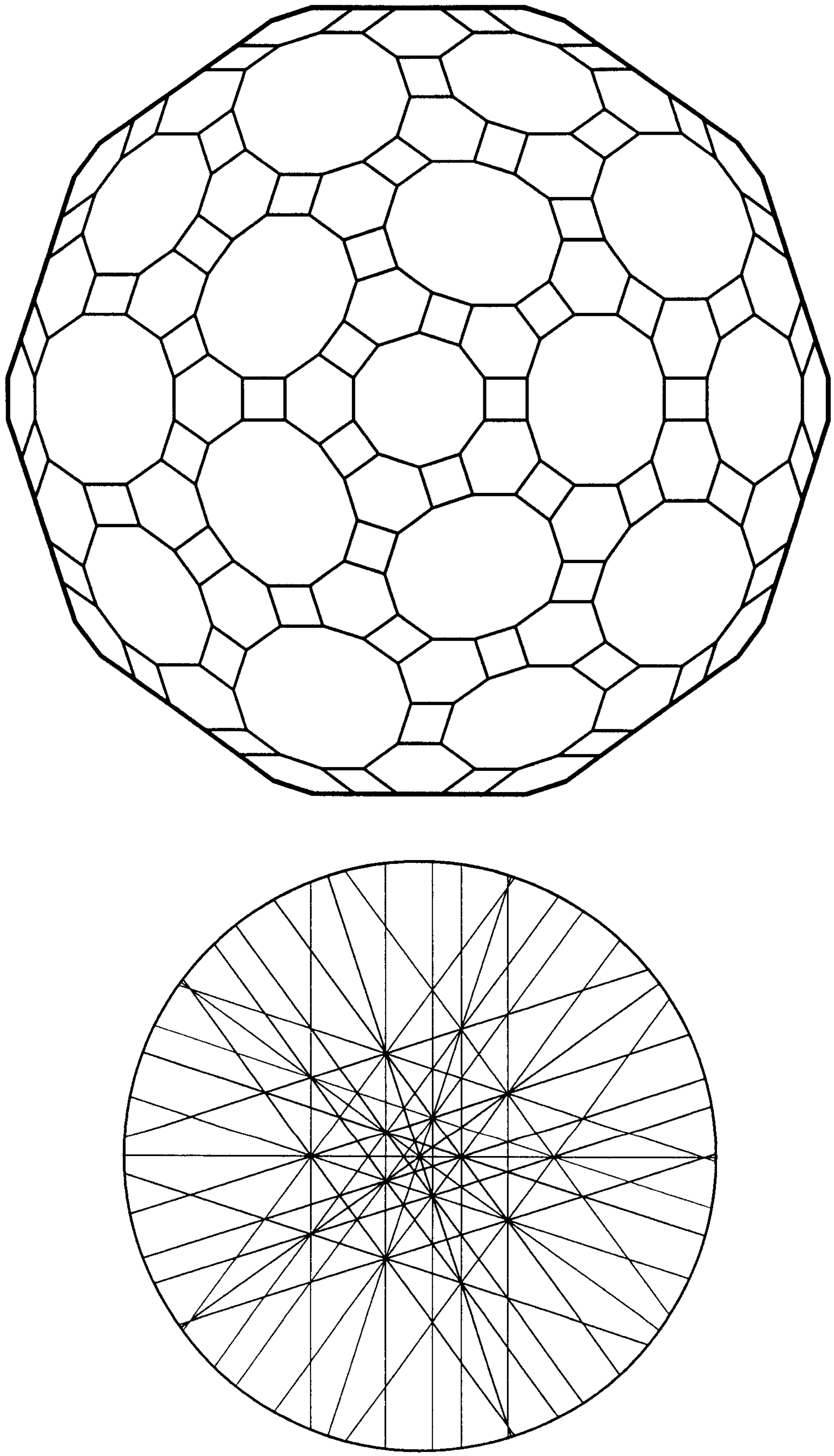

Figure 5. The simple line arrangement $A31_1^*$.

```
%!PS-Adobe-2.0 EPSF-2.0

300 570 translate
0.8 0.8 scale
0 setgray

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  Here start the definition of the line data                     %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

/Lines [
%  Homogous Coordinates (x,y,d),
%  (x,y) shoud be the normalVector,
%  d     should be the distance from the origin:
%  The lines must be sorted by increasing slope.

%  x:       y:        d:
   0 sin    0 cos     0.03
   0 sin    0 cos     0.13
   0 sin    0 cos    -0.07
  36 sin   36 cos     0.03
  36 sin   36 cos     0.13
  36 sin   36 cos    -0.07
  72 sin   72 cos     0.03
  72 sin   72 cos     0.13
  72 sin   72 cos    -0.07
 108 sin  108 cos     0.03
 108 sin  108 cos     0.13
 108 sin  108 cos    -0.07
 142 sin  142 cos     0.03
 142 sin  142 cos     0.13
 142 sin  142 cos    -0.07

] def

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  Here start many useful functions and definitions               %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

/defvar {exch def} def
/size 70 def

/NumberOfLines Lines length 3 div .1 add  cvi def

/LineWithIndex
   {/lineIndex defvar
    Lines lineIndex 3 mul get
    Lines lineIndex 3 mul 1 add get
    Lines lineIndex 3 mul 2 add get } def

/DirectionWithIndex
   {LineWithIndex pop} def

/LineScale
   {/ss defvar
    ss mul exch ss mul exch} def

/MovePositiveInDirection
    {DirectionWithIndex 1 size mul LineScale rmoveto} def

/MoveNegativeInDirection
    {DirectionWithIndex -1 size mul LineScale rmoveto} def

/LinePositiveInDirection
    {DirectionWithIndex 1 size mul LineScale rlineto} def

/LineNegativeInDirection
    {DirectionWithIndex -1 size mul LineScale rlineto} def

/Det {  /m11 defvar /m12 defvar /m13 defvar
        /m21 defvar /m22 defvar /m23 defvar
        /m31 defvar /m32 defvar /m33 defvar

        m11 m22 m33 mul mul
        m12 m23 m31 mul mul add
        m13 m21 m32 mul mul add
        m11 m23 m32 mul mul neg add
        m12 m21 m33 mul mul neg add
        m13 m22 m31 mul mul neg add
} def

/Sign {dup .00001 ge {pop 1} {-.00001 le {-1} {0} ifelse} ifelse} def
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  Here starts two very high level functions                      %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

/SignVector{
      [
         0 1 NumberOfLines 1 sub {
            LineWithIndex
            index1 LineWithIndex
            index2 LineWithIndex Det Sign } for]
        } def

/DrawTile {
         /cv defvar
         0 0 moveto
         0 1 NumberOfLines 1 sub {
            /index defvar
            /sign {cv index get} def

             1 sign eq {index MovePositiveInDirection}
                       {index MoveNegativeInDirection} ifelse
            } for

         0 1 NumberOfLines 1 sub {
            /index defvar
            /sign {cv index get} def
             0 sign eq {index LinePositiveInDirection
                        index LinePositiveInDirection} if
            } for

         0 1 NumberOfLines 1 sub {
            /index defvar
            /sign {cv index get} def
             0 sign eq {index LineNegativeInDirection
                        index LineNegativeInDirection} if
            } for
         } def

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  Here starts the drawing of the line arrangement               %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

gsave
0 -480 translate
1.3 1.3 scale
1 setlinewidth
0 0 150 0 360 arc stroke
0 0 150 0 360 arc clip
0.5 setlinewidth

0 1 NumberOfLines 1 sub {
          LineWithIndex
          gsave
            5 mul size mul /distance defvar
            atan rotate
            -10000 distance moveto 10000 distance lineto stroke
          grestore
           } for

grestore

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  Here starts the drawing of the zonozopal tiling                %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

NumberOfLines 3 div setlinewidth
6 NumberOfLines div 6 NumberOfLines div scale
1 setlinejoin

0 1 NumberOfLines 2 sub {/index1 defvar
     index1 1 add 1 NumberOfLines 1 sub {/index2 defvar
      SignVector DrawTile} for } for stroke

NumberOfLines 2 div setlinewidth
[0 1 NumberOfLines 1 sub{pop 0} for] DrawTile stroke

showpage
```