Stars around I—PostScript straightaway

Kees van der Laan

Abstract

Drawing the outline of stars is discussed. A METAFONT/Post and a PostScript program are included, next to a Colombus' egg PostScript solution. The appendices contain compositions of stars and a few versions of a PostScript operator to calculate the intersection point of two lines determined by four points.

Keywords:EPS, fractal, intersection point 2 lines, Koch island, METAFONT/Post, MFTOEPS, outline, PostScript, ROEX, star, tiling.

1 Introduction

This note arose as a comment, well asked for feedback, on NTG's Graphics course of 1996, where the teacher Jackowksi—the TEX, METAFONT, POSTSCRIPT guru of the Polish GUST user group—extensively used a star as example.[1]

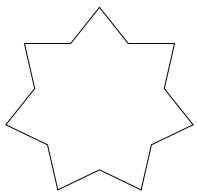
The included small, and hopefully intelligible, codes are in POSTSCRIPT—the portable graphics language, similar to FORTRAN as the portable numerical language—ready to be enriched by other tools such as Adobe Illustrator, to be included as EPS in (La)TEX scripts, or to be viewed by ghostscript as such.[2] However, now and then a higher level METAFONT code has been given as well, because of special features of METAFONT, and because Jacko used METAFONT as user language.

In Adobe's POSTSCRIPT tutorial—see their blue book—a star is used as example to illustrate, how to

- translate, rotate and use the CTM[3]
- execute repeatedly an executable array
- how to fill up the 'contour.'

Jackowski used even-ordered stars to illustrate aspects of METAFONT programming, MFTOEPS—transformation of METAFONT into EPS, and his ROEX—removing overlap utilities.

Example *(Contour of 7-pointed star)*

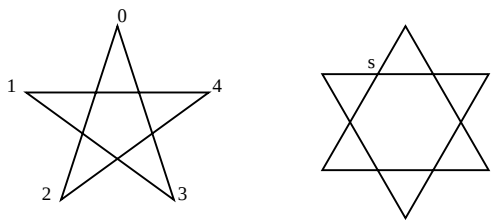


1.1 Notations

What is meant precisely by a star is left to the intuition. The above is a regular, simple and straight odd-ordered star, meaning a symmetric star with an odd number of vertices. Vertices are denoted by $z_0, z_1, \ldots z_n$. Each z is a pair (x, y), the conventional way to denote a point in the xy-plane.

For the METAFONT program I did not build a character from it. On my Mac with BlueSky's METAFONT this is not necessary.

Example *(5-pointed and 6-pointed star)*



2 How to draw a star?

A natural approach is to start at a vertex and draw, but . . . what are the coordinates of the next turning point?

[1] For the material Jacko left behind after the course consult: http: //www.ntg.nl/course18/ or the more elaborate submission from GUST on the CTAN . . .

[2] METAFONT codes need to be converted via MFTOEPS, or MetaPost has to be used in order to yield the PostScript codes.

[3] CTM stands for Current Transformation Matrix, a central issue in PostScript to relate the so-called *user* space to the *device* space (screen or printer). See the tutorial in Adobe's blue book for how this works.

Adobe circumvents this for the 5-point star by drawing the diagonals of a pentagon—a so-called pentagram—that is

$$z_0 \to z_2 \to z_4 \to z_1 \to z_3 \to z_0$$

This is sufficient for filling up the star. A nice way for odd-ordered stars. Their POSTSCRIPT program reads as follows.[4]

```
%!PS-Adobe- 5-pointed star in one-stroke
%%BoundingBox: 0 -50 75 25
/diagonal{72 0 lineto   %add to path
 currentpoint translate %move origin
 -144 rotate            %rotate dev space
}def
/star5{moveto currentpoint translate
 4{diagonal}repeat closepath
 gsave .5 setgray fill grestore stroke
}def
200 200 star5 %showpage
```

Jackowski added his ROEX functionality in order to retain the structure of the contour, read remove the 'overlap,' the inner lines.[5]

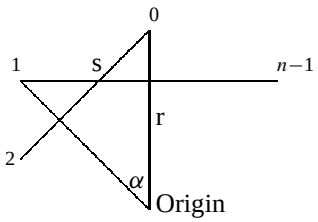
The above 2 stars need an eraser—Jackowski's ROEX—when one wants just the contour for filling up or coloring purposes. We can do without an eraser in various ways, hang on.

2.1 The point z_s

The coordinates of point z_s obey the following formula,[6] with $\alpha = 360/n$, n the order,

$$r\cos\alpha\,(-\tan\frac{\alpha}{2}, 1).$$

y_s can be derived from the sketch below.[7]



x_s follows from 'solving the equation' for the intersection point z_s of the lines $\overline{z_0\,z_2}$ and $\overline{z_1\,z_{n-1}}$. This can be done by METAFONT for sure, but applying a little backside of the envelope geometry, yields

$$x_s = x_2\,\frac{y_0 - y_s}{y_0 - y_2}.$$

Drawing the star comes down to drawing the first side $\overline{z_0\,z_s\,z_1}$ and rotate this n times.

3 Star in METAFONT/Post

In METAFONT the computation of the coordinates of the point z_s can be easily specified as can be seen from the following program.

```
%Star contour. Oct 1996. cgl@rc.service.rug.nl.
%Only points z0, z1, z2, z[n-1] are needed.
%Rest follows from symmetry (no ROEX needed.)
n=6; alfa=360/n;
z0=(0,100);
z1=z0 rotated alfa;
z2=z1 rotated alfa;
z[n-1]=z1 reflectedabout (origin, z0);
pair zs;
%equation to be solved by METAFONT
zs=whatever[z0,z2]=whatever[z1,z[n-1]];
%essential side
path p; p=z0--zs--z1;
draw for k=0 upto n-1:
     p rotated (k*alfa)--endfor cycle;
showit;
end
```

The above implies solving 2 equations, although hidden by the declarative specification.[8] We can do without solving equations. Hang on.

4 Star in PostScript

In POSTSCRIPT the star can be drawn as follows with the use of the formula for the point z_s.[9]

```
%!PS-Adobe- star outline, labels 0, 1,...n-1
%%BoundingBox: -100 -100 100 100
/n 7 def /r 100 def
/alfa 360 n div def
/cosa alfa cos def
/cos2a cosa dup mul 2 mul 1 sub def
/sina alfa sin def
/sin2a cosa sina mul 2 mul def
/x1 sina r mul neg def %abscis 1st vertex
%inward corner
```

[4] See their blue book p51-53. I used the name diagonal instead of starside, because that is what it is of the pentagon. Jackowski varied these odd-ordered one-stroke star drawings by joins of a spline under tension, yielding pictures which do remind us of flowers instead of stars. Nice. My Malbork window alludes to this in PostScript. Also in 'Just a little bit of PostScript' I've shown how flowers can be drawn straightaway in PostScript as variant of a line bundle. So polygons, stars and flowers form some sort of continuum.

[5] As he demonstrated during NTG's Graphics and TEX course, October 1996. He draws a half-ordered polygon and reflects this in order to draw a star.

[6] In 'Afterthoughts' the choice of s is loosened, and restricted only by having equal distance to z_0 and z_1.

[7] Note that the line $\overline{z_0\,z_2}$ is not realistic. So don't measure the coordinates of z_s. The flavour of the picture is OK and sufficient for supporting a little math.

[8] In PostScript one has to write an operator. Given the operator the specification goes equally simple, modulo some syntactic sugar.

[9] Of course a similar METAFONT program could have been written, equivalent to the PostScript one modula some syntactic sugar.

```
/xs cosa 1 sub cos2a 1 sub div
       sin2a mul r mul neg def
/ys cosa r mul def %also ordinate 1st vertex
%
0 r moveto
n{xs ys lineto x1 ys lineto
  alfa rotate}repeat
stroke %showpage
```

Remarks. For inclusion in a script more than once one can reuse the POSTSCRIPT program parameterized over the order by splitting off the `/n ... def` line and repeat this with the actual value for n before each invoke of the constant part. Some sort of 'two-part' POSTSCRIPT codes.

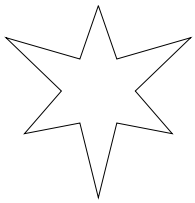
5 Afterthoughts

Looking more closely into the algorithm, biased by the formula for the point s, reveals degeneracy. Instead of the star one gets for

- $n = 3$ a triangle
- $n = 4$ a cross.

A more general—and flexible—approach is to consider the star as a polygon with as side a broken line.[10]

We can also variate the location of the corners of the polygon. Because while writing this note it was about X-mas time I have included the following 'irregular' star.



The above irregular star is obtained as follows.[11]

```
%!PS-Adobe- 6-pointed irregular star
%%BoundingBox: -125 -125 125 100
/n 6 def /r 100 def /alfa 360 n div def
/alfah alfa 2 div def
%my choice of rin, make yours
/rin r 2 div alfah cos mul def
0 r moveto
1 1 n{alfah rotate
  0 rin lineto
  alfah rotate
  2 mod 0 eq {0 r lineto}
             {0 r 1.25 mul lineto}ifelse
  }for stroke %showpage
```

Do you see how to adjust the above for a 'pointing' star, with only one corner stratched?

6 Acknowledgements

Thank you Jackowski for paying us a visit and sharing your experience, next to giving your comments on this note, especially in suggesting the use of itransform for determining the intersection point of two lines. Joseph Romanovsky hinted at other POSTSCRIPT's features such as the stack and index. As usual Jos Winnink proofed the paper and lend a helping hand in procrusting the BLUe script into MAPS.

7 Conclusions

Graphics is all about the right strokes and fills. After all no math was needed to draw stars. They can be drawn straightforwardly, as it should be.

For real-life problems the use of Jackowski's functionalities can be advantageous, however. For his arguments see his lecture 'A METAFONT-EPS interface,' EuroTEX 95, Arnhem.

Detailing with stars was nice and familiarized me further with POSTSCRIPT.

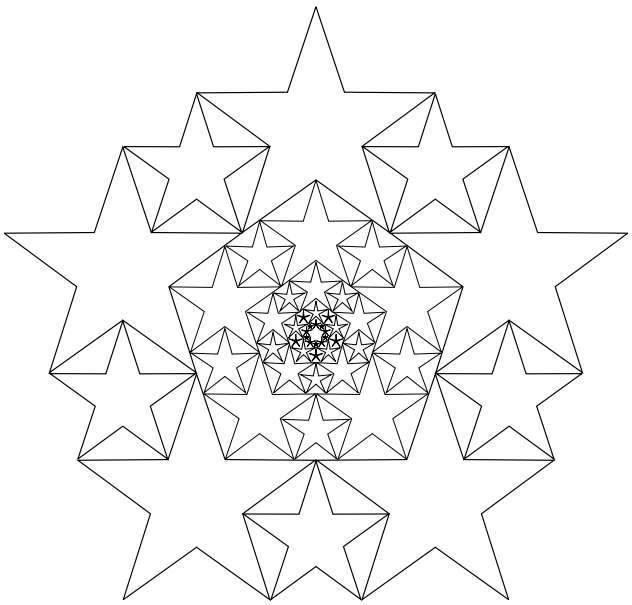
My case rest.

Have fun, and all the best.

Appendix I: Composition of stars

Because we can draw stars so easily it is fun to play a little longer with it. Only the code for the Koch islands is incuded. I'll come back on the issues in special notes about tilings and fractals.

Example *(Tiling with stars)*

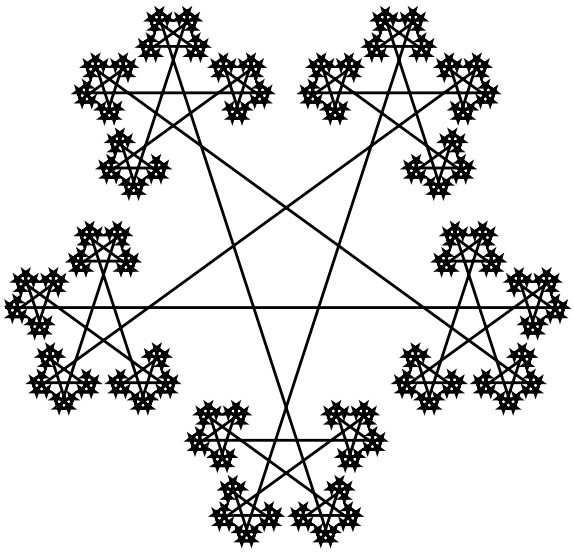


[10] Jackowski introduced flexes, that is polygons with splines as side, already at EuroTEX 94.

[11] The Colombus' egg solution: a star seen as a polygon with a broken line as side.

Example *(Star fractal)*

The star fractal picture (and code) is borrowed from Lauwerier's Fractals.[12]

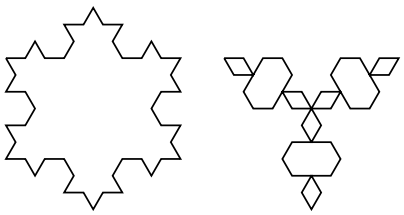


Example *(Koch islands)*

Another class of stars result from wiggy-line fractals as sides of a polygon. A nice example is the so-called Koch island. A Koch fractal—the wiggy-line—results from repeatedly replacing the middle third piece of a line by two in a hat shape.[13]

The star fractal picture (and code) is borrowed from Lauwerier's Fractals.

Below are Koch islands, outbound and inbound, composed from Koch fractals of order 2 as sides of a triangle.



The order 1 outward island is equivalent to the 6-pointed star, discussed in the beginning of this note.

For those who can't wait to play with it the POSTSCRIPT code is appended below. The explanation will be given in the upcoming note about Fractals.[14]

```
%!PS-Adobe- Koch wiggy fractal, well islands
%%Creator: Lauwerier; in PS cgl
%%BoundingBox: 0 -80 225 20
/order 2 def
/base 4 def /length 100 def
/linesize length 3 order exp div def
/up base order exp 1 sub def
/koch{0 -60 60 0 %put angles on stack
0 1 up{/k exch def
  direction
  angle rotate
  linesize 0 rlineto
  angle neg rotate
}for
}def
/direction{
  /angle 0 def
  {/angle angle k base mod 2 add
          index add def
   /k k base idiv def
   k 0 eq {exit}if}loop
}def
gsave
0 0 moveto  koch
-120 rotate koch
-120 rotate koch
stroke
grestore
125 0 translate
0 0 moveto
 -60 rotate koch
 120 rotate koch
 120 rotate koch
stroke %showpage
```

Remark. The angles can be equally well stored in an array.

Appendix II: The intersection point of two lines

I pondered about a POSTSCRIPT operator for calculating the intersection point of 2 straight lines. Joseph Romanovsky pointed me into the right direction—well, stack-oriented—and provided a tiny POSTSCRIPT program, based on Kramer's rule. I have adapted his code into LU decomposition with partial pivoting.[15] Three variant solutions have been appended.

Algorithm

The process consists of 2 steps

[12] Fractals—Meetkundige figuren in eindeloze herhaling. Aramith. (It contains codes in BASIC.) When I borrowed material to illustrate turtle graphics in TEX, this fractal was too complicated, because of the drawing at the awkward angles, $144k^\circ$, $k = 0, 1, \ldots$. Note that it is just one broken line. The transcription from BASIC into PostScript is a nice exercise for increasing your PostScript fluency.

[13] This scheme of replacing a middle part repeatedly has been used before by Cantor—in his so-called dust fractals—and Sierpiński—in his so-called carpets.

[14] What about nesting scaled versions and color these? Bound to be beautiful.

[15] The degenerate cases—parallel or co-linear lines—are not taken care of, but can be accounted for, although I don't know how to inform the user. Joseph also suggested to make use of index, to curtail unnecessary roll-s.

- $z_1, z_2 \rightarrow e, a, b$, ditto for the other points, that is from points to coefficients of the equation of the line determined by those points

$$e = x_1 y_2 - x_2 y_1$$
$$a = y_2 - y_1$$
$$b = x_1 - x_2$$

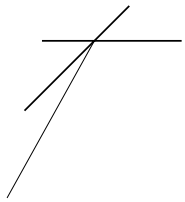
- solving the equations

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} e \\ f \end{pmatrix}.$$

PostScript code

```
%!PS
/makecoef{%z1 z2 -> e a b
4 copy           %x1 y1 x2 y2 x1 y1 x2 y2
4 -1 roll mul    %x1 y1 x2 y2 y1 x2 (y2x1)
3 1 roll mul sub %x1 y1 x2 y2 (y2x1-y1x2)
5 1 roll 3 -1 roll sub
                 %(y2x1-y1x2) x1 x2 y2-y1
3 1 roll sub     %(y2x1-y1x2) y2-y1 x1-x2
}def
/solveit{%e a b f c d  -> x y,
         %intermediate p is pivot
%Equations: ax + by = e
%           cx + dy = f
%pivot handling  %e a b f c d
1 index abs      %e a b f c d |c|
5 index abs      %e a b f c d |c| |a|
gt {6 3 roll} if %exchange 'equations'
%stack: e a b f c d or f c d e a b,
%first is in comments below
exch 4 index     %e a b f d c a
div              %e a b f d p
6 -1 roll dup 6 1 roll 3 1 roll
                 %a e b f e d p
4 index exch     %a e b f e d b p
dup 4 1 roll     %a e b f e p d b p
mul sub          %a e b f e p (d-b.p)
4 1 roll mul sub  exch div
%a e b (f-e.p)/(d-b.p) = a e b y
dup 5 1 roll mul sub exch div exch
%stack: x y
}def
/intersect {%p1 p2 p3 p4 -> x y
makecoef 7 3 roll
makecoef
solveit
}def
%
%Example
%
100 600 translate
/z1 {10 50} def
/z2 {70 110} def
/z3 {100 90} def
/z4 {20 90} def
z1 moveto z2 lineto
z3 moveto z4 lineto
0 0 moveto
z1 z2 z3 z4 intersect lineto
% exchanged equations
%z3 z4 z1 z2 intersect lineto
.1 setlinewidth stroke %showpage
```

with result



Although after all I did not use the above code. I think it is worthwhile nonetheless, and I hope you like it too, if not for the example of how one can do a few calculations in POSTSCRIPT.

Alternative code

After having studied examples in the blue book POSTSCRIPT's parameter mechanism became clear to me, with the advantage of using names instead of muddling around with the stack.[16] Appended is an alternative for /solveit.

```
/solveit{%e a b f c d -> x y
% store stack values by their name
 /d exch def /c exch def /f exch def
 /b exch def /a exch def /e exch def
a abs c abs lt{exchangeequations}if
/p c a div def
%y=(f-p.e)/(d-p.b), x=(e-b.y)/a
/y f e p mul sub
   d b p mul sub div
def
e b y mul sub a div %x
y                   %y
}def
%
/exchangeequations
{c /c a def /a exch def
 d /d b def /b exch def
 f /f e def /e exch def}def
```

The variables can be made local by using a dictionary. The question arises whether the stack version is sufficient, circumventing variables and pushing an extra dictionary. Make your choice.

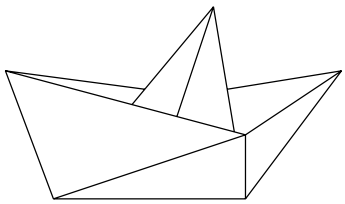
[16] We can also use arrays in PostScript. For 2-dimensional arrays we must nest 1-dimensional arrays. At the moment I don't know an elegant way of overwriting elements in a nested array. Of course one can store the array stratched and write the algorithm in those terms. However, for this small problem the direct use of the stack looks clear enough.

Using PostScript's itransform

As a comment on this note Jackowski suggested to use itransfrom. Indeed that is possible.[17] The variant intersect is appended below, without the details of how the POSTSCRIPT matrix should be filled. I have simplified the equations for the occasion by normalizing them, $e = 1$, and $f = 1$.

```
%!PS-Adobe- cgl Dec 1996
%Intersection of lines spanned by
%z1, z2 and  z3, z4, via itransform
/intersect{%z1 z2 z3 z4 -> x y
 /y4 exch def /x4 exch def
 /y3 exch def /x3 exch def
 /y2 exch def /x2 exch def
 /y1 exch def /x1 exch def
 %makecoef
 1 1 [x1 x2 y1 y2 0 0] itransform
          /b exch def /a exch def
 1 1 [x3 x4 y3 y4 0 0] itransform
          /d exch def /c exch def
 %solveit
 1 1 [a c b d 0 0] itransform
}def
%test
100 600 translate
/z1 {10 50} def
/z2 {70 110} def
/z3 {100 90} def
/z4 {20 90} def
z1 moveto z2 lineto
z3 moveto z4 lineto
0 0 moveto
z1 z2 z3 z4 intersect lineto
stroke showpage
```

Remark. In reproducing the GUST battleship—the EuroTEX'94 logo—the variant with the itransform did not work???



```
%!PS-Adobe- cgl, Mrt 97
%%BoundingBox: -20 0 112.5 62.5
%/intersect{%see above...
%}def
/s 25 def %scaling parameter
/mean{%p0 p1 on stack -> .5[p0, p1]
      exch 4 -1 roll add .5 mul
           3  1 roll add .5 mul
}def
/p0{0 0}def
/p1{3 s mul 0}def
/p2{4.5 s mul 2 s mul}def
/p3{3 s mul s}def
/p4{-.75 s mul 2 s mul}def
/top{2.5 s mul 3 s mul}def
/p5{p0 top p3 p4 intersect}def
/p6{p0 p1 mean top p3 p4 intersect}def
/p7{top p1 p3 p4 intersect}def
/p8{p2 p5 top p1 intersect}def
/p9{p8 dup 0 exch top p0 intersect}def
gsave
p0 moveto p1 lineto p2 lineto p3 lineto
p0 lineto p1 moveto p3 lineto p4 lineto
p0 lineto
top moveto p5 lineto
top moveto p6 lineto
top moveto p7 lineto
p2 moveto p8 lineto
p4 moveto p9 lineto
stroke grestore %showpage
```

[17] Whether to consider this as a feature or a (mis)use that is up to the reader. I guess the reader who is just interested in intersect does not mind what is under the hood.