

Stars around II—what a little math can do

Kees van der Laan

Abstract

The problems as treated by Jackowski in his lecture at NTG's 1996 fall meeting: a tangent to a circle, intersection of circles, V-like logo, and (circular) clipping of waves, are provided with alternative solutions, along with straightaway PostScript codes.

Keywords: Analytic geometry, clipping, intersection, linear equations, logo design, METAFONT/Post, MFTOEPS, outline, PostScript, ROEX, tangent.

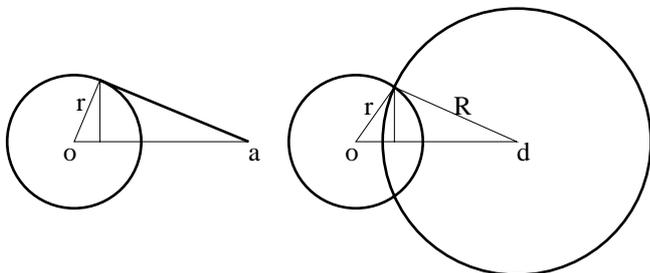
1 Introduction

This note is a summary—and a little more—of Jackowski's lecture on the occasion of him being invited by NTG for the 1996 fall meeting. His lecture comprised the following problems. Drawing

- tangent to a circle
- intersection of two circles
- V-like logo
- clipping by a circle.

I have chosen the origin appropriately and so on, in order to arrive at the simplest representations, unblurred by notational overhead. Furthermore, in the picture I have stressed the circles.

Jackowski emphasized METAFONT's facility to solve (linear) equations. He provided METAFONT codes and transformed these into PostScript. I'll include straightaway PostScript.



2 Tangent to a circle

The problem is: given a circle with radius r , and a point outside the circle at distance a from the center, find the coordinates of the tangent point(s) and draw the picture.¹

For the (left) figure above the tangent points are

$$\frac{r}{a} (r, \pm\sqrt{a^2 - r^2}), \quad r \leq a.$$

The limit situations

$$\begin{aligned} a = r, & \quad (x, y) = (r, 0) \\ a \rightarrow \infty, & \quad (x, y) = (0, \pm r) \end{aligned}$$

can be easily verified.

2.1 PostScript code

The associated PostScript code reads as follows.²

```

%!PS-Adobe- Tangent to a circle
%%BoundingBox: -25 -25 75 25
/Times-Roman findfont 10 scalefont setfont
/r 25 def /a 65 def /rdiva r a div def
/x r rdiva mul def
/y r 1 rdiva dup mul sub sqrt mul def
/angle y x atan def
a 0 moveto 0 0 r angle 360 angle add arc
stroke
%auxiliary lines
a 0 moveto x 0 lineto x y lineto
0 0 lineto x 0 lineto
.1 setlinewidth stroke
%labeling extras
x 2 div 4 sub y 2 div moveto (r) show
-4 -7 moveto (o) show
a -7 moveto (a) show %showpage

```

¹I simplified it by assuming that the origin is at the centre of the circle and that the point a is along the x-axis. Not essential because of the symmetry. Hopefully, confusing non-relevant details have been kept at bay.

²Note how PostScript allows—the operator `arcto`—drawing a tangent to a circle once the tangent point is known. `arcto` is used for drawing rounded corners.

2.2 Jackowski's pedagogical METAFONT code

I have adapted Jacko's codes towards my simplified choice of origin etc. l is the length of the tangent line. Building a character out of it is not so relevant in order to show Jacko's approach—basically equating vectors, declaratively—and therefore I have omitted these aspects, next to the labeling issues.

```
%Tangent to a circle in MF
r:=25; a:=65; l:=a + r;
%Equation in z to be solved by MF
((z-(a,0))rotated-90)/l=z/r;
%Visualize
pickup pencircle scaled 4;
draw fullcircle scaled 2r;
draw z--(a,0);
pickup pencircle scaled 1;
draw z--origin--(a,0);
draw (xpart z,0)--z;
showit
```

3 Intersection of circles

The problem is: given 2 circles with radius r and R , with centres d apart, find the intersection points and draw the picture.

A variant formulation of this problem is: given the length of the sides of a triangle, draw the triangle.³

A little planimetry yields for the intersection points

$$\frac{1}{2d} (d^2 - (R^2 - r^2), \pm \sqrt{(d^2 - (R - r)^2)((R + r)^2 - d^2)})$$

with the restriction $R - r \leq d \leq R + r$.

Limit situations are for $d = R \pm r$ with $(\pm r, 0)$ as intersection points.

3.1 PostScript code

In the PostScript below I've omitted the labeling extras.

```
%!PS-Adobe- Intersection of circles
%%BoundingBox: -25 -50 110 50
/r 25 def /R 50 def /d 60 def
/x d R d div R mul r d div r mul sub sub
  2 div def
/y 1 x r div dup mul sub sqrt r mul def
r 0 moveto 0 0 r 0 360 arc
d R add 0 moveto d 0 R 0 360 arc
stroke
%auxiliary lines
x y moveto x 0 lineto 0 0 lineto
x y lineto d 0 lineto x 0 lineto
.1 setlinewidth stroke %showpage
```

Remark. As can be seen from the illustration the ordinate can be expressed in its abscissa $\pm \sqrt{r^2 - x^2}$, which is computationally simpler, or its variant $\pm r \sqrt{1 - (x/r)^2}$, which

³The pre-computer tool was a pair of compasses. Jackowski called his lecture for that reason I presume 'Virtual ruler, virtual compasses, virtual eraser.'

⁴Once the abscissa is known it's no use to muddle around with vectors. Just a simple Pythagorean subtraction.

is computationally and numerically better, circumventing intermediate overflow. For mathematical purposes the earlier given formula is more suited.

3.2 Jackowski's pedagogical METAFONT code

I have simplified the code for the determination of the abscissa.⁴

```
%Intersection of circles
r:=25; d:=60; R:= 50;
xpart z = .5(d -(R*R - r*r)/d);
ypart z = r + xpart z;
%visualize
pickup pencircle scaled 4;
draw halfcircle rotated -90 scaled 2r;
draw halfcircle rotated 90 scaled 2R
  shifted (d,0);
pickup pencircle scaled 1;
draw z--origin--(d,0)--cycle;
draw (xpart z,0)--z;
showit
```

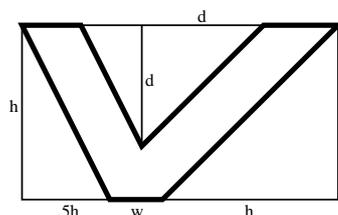
Remarks. In view of METAFONT's limited number range better change in practice

$$(R * R - r * r) / d \rightarrow (R / d * R - r / d * r).$$

METAFONT's `intersectionpoint` can be used, but that is a bit overdone, especially when the analytical solution is at hand. Moreover, the accuracy is low.

4 V-biased logo

The problem is given the height, h , the angles of the legs next to the width of the legs, w , and that the base of the V equals w . Draw the V-biased logo.



The point where the legs meet—determined by d —is implicit. A little backside of the envelope calculation yields

$$d = h - \frac{\sqrt{5} + 2(\sqrt{2} - 1)}{3} w.$$

Remark. If the distance of the incision point to the base should equal w then the formulas are even simpler, and the drawing can better end with a closepath for the base.

4.1 PostScript code

```

%!PS-Adobe- Jacko's V-biased logo
%%BoundingBox: -10 -10 200 100
/h 100 def /w 30 def
/d h 1.021 w mul sub def
0 h moveto
h 2 div 0 lineto
w 0 rlineto
h h rlineto
-1.4142 w mul 0 rlineto
d neg dup rlineto
1.1180 w mul h lineto
closepath
3 setlinewidth stroke
%auxiliary lines
0 0 moveto
0 h lineto
h 1.5 mul w add h lineto
0 h neg rlineto
closepath
1.1180 w mul d 2 div add h moveto
0 d neg rlineto
1 setlinewidth stroke
%labeling extras
/Times-Roman findfont 10 scalefont setfont
-7 h 2 div moveto (h)show
h 5 div -10 moveto (.5h)show
h 2 div w 2 div add 3 sub -10 moveto (w)show
h w add 3 sub -10 moveto (h)show
h 1.5 mul w add 2 add .5 h mul moveto (h)show
1.1180 w mul d 2 div add 2 add
h d 2 div sub moveto (d)show
h h 2 add moveto (d)show %showpage

```

Jacko's METAFONT code solves equations, which is a bit overdone IMHO, with all respect.

5 Clipping by a circle

The problem—a wavy sun—is: create new contours for coloring purposes after clipping the old ones by a circle. Here I made use of simple a priori contours: just broad strokes to conceive the idea.

I color, well... first and then clip. I don't care whether the contours extend the 'window.' This is a different processing order than with Jacko. He clips, rebuilds contours, and then colors or whatever. Definitely more to the point, but

⁵Maybe the problem is induced by METAFONT's lack of a clipping path primitive?

⁶Jacko warned that ROEX is not compatible with MetaPost. The reason is that MetaPost does not make use of pixel values. A year ago or so I experimented with the examples from the ROEX package and could achieve most of the effects if not all in METAFONT alone.

sometimes we can do without. Just be aware of it.



5.1 PostScript code

In PostScript this can be done straightaway as follows.⁵

```

%!PS Tiling Jacko's -s.mf example
%%BoundingBox: -50 -50 50 50
/tile {-2 0 moveto %+/-2 To prevent notch
%any contour or this broad stroke (laziness);
%mind the notch
25 10 25 10 50 0 curveto
75 -10 75 -10 102 0 curveto
} def
/frame{%circle around origin, r=50
50 0 moveto 0 0 50 0 360 arc}def
/dotiling{-50 -50 translate
.9 -.1 .1{setgray tile stroke
0 15 translate}for
}def
gsave
frame clip
1 setgray stroke %white clipped circle
10 setlinewidth %broad strokes
dotiling
grestore %showpage

```

5.2 Jackowski's METAFONT code

The code makes use of `expand_stroke` and `remove_overlap` from his ROEX package, and are beyond my experience as yet.⁶

Jacko is not happy with me calling this an alternative solution. Whatever in the name, I've shown that 80% (or more) of the effect can be done by 20% (or less) of the energy. Moreover, it's just an example.

6 Acknowledgements

Thank you Jackowski for paying us a visit and sharing your experience. As usual Jos Winnink proofed the script and lend a helping hand in procrusting the BLUE script into MAPS submission.

7 Conclusions

It's great that Jackowski showed us how METAFONT can be used to solve and visualize Analytic Geometry exer-

cises, next to sharing his METAFONT, PostScript, and associated tools experience with us.

Our tools become fancier and fancier, but we should not forget to do a few a priori backside of the envelope calculations, whenever feasible. This together with PostScript allows us to include nice graphics at the expense of a little knowledge of PostScript and the merging macros.

My added value is that I've shown that PostScript can be used equally well for visualizing Analytic Geometry exercises, modulo some lumps of syntactic sugar. For me the choice of using PostScript for these problems was necessary for illustrating this note, simply because on my Mac I don't have MFTOEPS running. Next to that I prefer to include pedagogical and intelligible PostScript code as well. They are so concise.

Important is a clear analysis, a minimal algorithmic setup, followed by coding in the language at hand, be it METAFONT, PostScript, or you name it.

I have no access to color printers as yet, and hope that the clipping issues and approaches of the last example have been demonstrated convincingly by grey scales.

A major advantage of Jacko's approach and experiments is that he easily can variate and discover beautiful results. Demonstrating, stimulating creativity. My approach is more constructivistic. I know what I'm after and try to achieve that as simple as possible, in the spirit of Gabo.

My case rest.

Have fun, and all the best.