A Short Presentation of Eukleides

Christian Obrecht

3, impasse Bellevue
F-89100 PARON
FRANCE
E-mail: christian.obrecht@wanadoo.fr

Many documents concerning mathematics, especially elementary mathematics, contain both text and geometric figures. To include such figures in a LaTeX document, a common solution is to create an EPS file using a "point and click" geometry software. In my opinion, this is not satisfying: I believe it's often more efficient to describe a figure in a script than to try to draw it on a computer screen.

Another way to illustrate a LaTeX document with a geometric figure is to use *PSTricks*. Unfortunately, this excellent package is not designed for geometry: in many cases one has to compute oneself a large number of numerical values (like coordinates), which is not efficient either.

This situation gave me the idea to create Eukleides, a Euclidean geometry drawing language. The main software related to this language is a compiler, named `eukleides` which can take a LaTeX source containing Eukleides code, and replace this code with *PSTricks* macros, producing a ready-to-TeX file. An X-Window front-end, named `xeukleides`, is also available. It allows to edit and to view Eukleides code, and provides additional interactive features.

Both softwares are released under GNU Public License. They have been developed on a GNU/Linux system and have been successfully ported on FreeBSD and Mac OS/X. Their source code is available on CTAN[1] or the Eukleides home page[2].

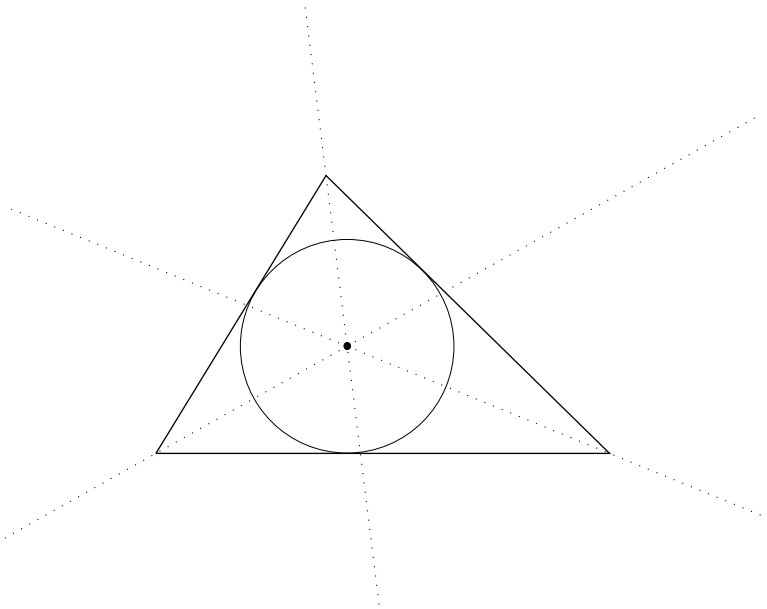
1. A simple example

The code below:

1 In `/tex-archive/support/eukleides/`.
2 At `http://perso.wanadoo.fr/obrecht/`.

```
A B C triangle
I = incircle(A,B,C)
draw(A,B,C) ; draw(center(I))
style(dotted)
draw(bisector(A,B,C))
draw(bisector(B,C,A))
draw(bisector(C,A,B))
style(full) ; thickness(.5)
draw(I)
```

leads to the following graphical result:



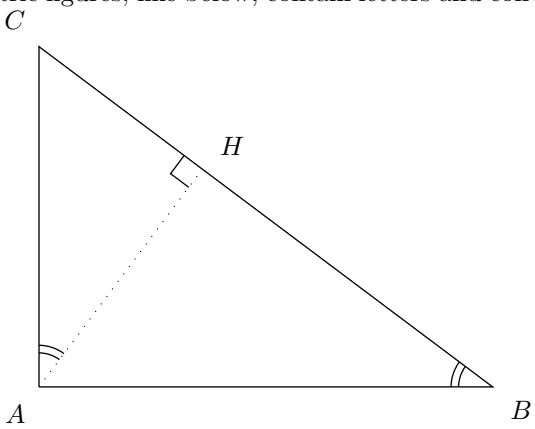
I believe that the meaning of the corresponding script is rather obvious. In fact, Eukleides has been designed in order to be close to the traditional language of geometry. In many situations, it allows to use very few coordinates to describe a diagram (or even none like in our example).

On the first line is a multiple assignment. With Eukleides, this is the usual way to define the vertices of a polygon. In our example, the obtained triangle is an optimal scalene triangle (such that $AB = 6$ cm). The '`triangle`' command

may be followed by various optional parameters. For instance, the statement `A B C triangle(6,4,5)` defines ABC such that $AB = 6$ cm, $BC = 4$ cm and $CA = 5$ cm. For specific kind of triangles, other commands are available: '`right`', '`isosceles`', '`equilateral`'.

2. Labels and marks

Often geometric figures, like below, contain letters and conventional marks.



Here is the corresponding code:

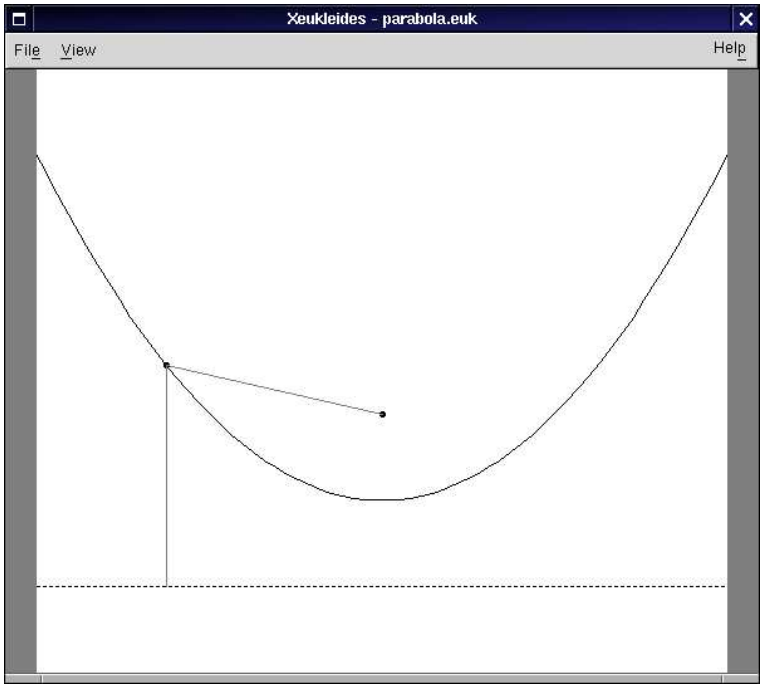
```
A B C right
H = projection(A,line(B,C))
draw(A,B,C) ; draw(segment(A,H),dotted)
draw("$A$",A,-130:)
draw("$B$",B,-40:)
draw("$C$",C,130:)
draw("$H$",H,40:)
mark(C,B,A,double)
mark(H,A,C,double)
mark(C,H,A,right)
```

In fact, with `eukleides` a label may be any valid TEX code. Hence it allows to insert formulas within figures. The statement `draw("$A$",A,-130:)` prints the label with an argument of $-130°$ in respect of A (and at a default distance of 3 mm).[3] Concerning angle marks, other types are of course available. Furthermore, it's also possible to mark segments.

[3] With Eukleides, '`:`' means "degrees" and '`<`' means "radian". These are necessary in order to distinguish angular and metric parameters.

3. An interactive figure

Here is a screen-shot of `xeukleides` in viewing mode.



The displayed figure is interactive, that is: when the right or left arrow keys are pressed the point moves along the parabola. It makes obvious that every point on a parabola is equidistant from its locus and directrix. At start, `xeukleides` looks like a simple text editor. So, to obtain the former figure one can type in the following script:

```
box(-4,-1,4,6)
t interactive(-2.5,.05,"A",right)
F = point(0,2)
D = line(point(0,0),0:)
P = parabola(F,D)
M = point(P,t) ; H = projection(M,D)
draw(P) ; draw(F) ; draw(M)
draw(D,dashed)
color(gray)
draw(segment(M,H)) ; draw(segment(M,F))
```

Then, to switch in viewing mode, the simplest way is to hit the escape key. The first line tells `xeukleides` that the lower left corner of the displayed picture has coordinates $(-4, -1)$ and the upper right $(4, 6)$.[4] The program uses the largest scale such that the figure fits into the window.

The second line has a rather complex syntax. Mainly, it defines t as an interactive variable with initial value -2.5. Every time the right (left) arrow key is pressed 0.05 is added to (subtracted from) t and the figure is updated.[5]

The statement on the fourth line defines a line given a point and an argument and the one on the fifth line defines a parabola given a focus and a directrix. Furthermore, M is defined as a point on the parabola with parameter t in respect of its internal parametric representation.

Conclusion

Several important features are not mentioned in this short presentation. One of these is the snapshot function of `xeukleides`, which allows to save several states of an interactive figure. An other is the '`trace`' command which is useful to draw various kind of curves and especially locuses.

I believe Eukleides is now a rather comprehensive system to typeset plane geometry diagrams. Since I want to keep it as small and simple and possible, I probably won't add any important feature. In the future, I'll just try to improve the existing code.

4 Defaults are $(-2, -2)$ and $(8, 6)$.

5 The same code compiled with `eukleides` leads to the figure in its initial state.