Abstract

This document brings together some notes about tkz-euclide, a tool for creating geometric figures. The two most important Euclidean tools used by early Greeks to construct different geometrical shapes and angles were a compass and a straightedge. My idea is to allow you to follow step by step a construction that would be done by hand (with compass and straightedge) as naturally as possible.
Example I

**Book I, proposition I _Euclide_**

*To construct an equilateral triangle on a given finite straight line.*

Explanation

The fourth tutorial of the *PgfManual* is about geometric constructions. T. Tantau proposes to get the drawing with its beautiful tool TikZ. Here I propose the same construction with *tkz-euclide*. The color of the TikZ code is green and that of *tkz-euclide* is red.

\begin{verbatim}
\usepackage{tikz}
\usetikzlibrary{calc,intersections,through,backgrounds}
\usepackage{tkz-euclide}
\end{verbatim}

How to get the line AB? To get this line, we use two fixed points.

\begin{verbatim}
\coordinate [label=left:$A$] (A) at (0,0);
\coordinate [label=right:$B$] (B) at (1.25,0.25);
\draw (A) -- (B);
\tkzDefPoint(0,0){A}
\tkzDefPoint(1.25,0.25){B}
\tkzDrawSegment(A,B)
\tkzLabelPoint[left](A){$A$}
\tkzLabelPoint[right](B){$B$}
\end{verbatim}

We want to draw a circle around the points A and B whose radius is given by the length of the line AB.

\begin{verbatim}
\draw let \p1 = ($ (B) - (A) $), \n2 = {veclen(\x1,\y1)} in
(A) circle (\n2)
(B) circle (\n2);
\tkzDrawCircles(A,B B,A)
\end{verbatim}

The intersection of the circles \(D\) and \(E\)

\begin{verbatim}
\draw [name path=A--B] (A) -- (B);
node (D) [name path=D,draw,circle through=(B),label=left:$D$] at (A) {};
node (E) [name path=E,draw,circle through=(A),label=right:$E$] at (B) {};
path [name intersections={of=D and E, by={\[label=above:$C$\]C, \[label=below:$C'$\]C'}}];
\draw [name path=C--'C',red] (C) -- (C');
path [name intersections={of=A--B and C--'C',by=F}];
node [fill=red,inner sep=1pt,label=-45:$F$] at (F) {};
\tkzInterCC(A,B)(B,A) \tkzGetPoints{C}{X}
\end{verbatim}

How to draw points:

\begin{verbatim}
\foreach \point in {A,B,C}
\fill [black,opacity=.5] (\point) circle (2pt);
\tkzDrawPoints[fill=gray,opacity=.5](A,B,C)
\end{verbatim}
Proposition I

To construct an equilateral triangle on a given finite straight line.

Let \(AB\) be the given finite straight line. ...

\begin{tikzpicture}[thick,help lines/.style={thin,draw=black!50}, old paper]
\tkzDefPoint(0,0){A}
\tkzDefPoint(1.25+rand(),0.25+rand()){B}
\tkzInterCC(A,B)(B,A) \tkzGetPoints{C}{X}
\tkzFillPolygon[triangle](A,B,C)
\tkzDrawSegment[input](A,B)
\tkzDrawSegments[red](A,C B,C)
\tkzDrawCircles[help lines](A,B B,A)
\tkzLabelPoints(A,B)
\tkzLabelCircle[below=12pt](A,B)(180){\mathcal{D}}
\tkzLabelCircle[above=12pt](B,A)(180){\mathcal{E}}
\tkzLabelPoint[above,red](C){$C$}
\tkzDrawPoints[fill=gray,opacity=.5](A,B,C)
\tkzText[text width=8cm,align=justify](0,-4){% \\emph{To construct an \textcolor{triangle}{equilateral triangle} on a given \textcolor{input}{finite straight line}.}} \end{tikzpicture}
Example II

**Book I, Proposition II, Euclid’s Elements**

To place a straight line equal to a given straight line with one end at a given point.

**Explanation**

In the first part, we need to find the midpoint of the straight line $AB$. With **TikZ** we can use the calc library

\begin{verbatim}
\coordinate [label=left:$A$] (A) at (0,0);
\coordinate [label=right:$B$] (B) at (1.25,0.25);
\draw (A) -- (B);
\node [fill=red,inner sep=1pt,label=below:$X$] (X) at ($ (A)!.5!(B) $) {};
\end{verbatim}

With **tkz-euclide** we have a macro `\tkzDefMidPoint`, we get the point $X$ with `\tkzGetPoint` but we don’t need this point to get the next step.

\begin{verbatim}
\tkzDefPoints{0/0/A,0.75/0.25/B,1/1.5/C}
\tkzDefMidPoint(A,B) \tkzGetPoint{X}
\end{verbatim}

Then we need to construct a triangle equilateral. It’s easy with **tkz-euclide**. With TikZ you need some effort because you need to use the midpoint $X$ to get the point $D$ with trigonometry calculation.

\begin{verbatim}
\node [fill=red,inner sep=1pt,label=below:$X$] (X) at ($ (A)!.5!(B) $) {};
\node [fill=red,inner sep=1pt,label=above:$D$] (D) at ($ (X) ! {sin(60)*2} ! 90:(B) $) {};
\draw (A) -- (D) -- (B);
\tkzDefTriangle[equil](A,B) \tkzGetPoint{D}
\end{verbatim}

We can draw the triangle at the end of the picture with

\begin{verbatim}
\tkzDrawPolygon{A,B,C}
\end{verbatim}

We know how to draw the circle $\mathcal{H}$ around $B$ through $C$ and how to place the points $E$ and $F$

\begin{verbatim}
\node (H) [label=135:$H$,draw,circle through=(C)] at (B) {};
\draw (D) -- ($ (D) ! 3.5 ! (B) $) coordinate [label=below:$F$] (F);
\draw (D) -- ($ (D) ! 2.5 ! (A) $) coordinate [label=below:$E$] (E);
\end{verbatim}

\begin{verbatim}
\tkzDrawCircle(B,C)
\tkzDrawLines[add=0 and 2](D,A D,B)
\end{verbatim}

We can place the points $E$ and $F$ at the end of the picture. We don’t need them now.

Intersecting a Line and a Circle: here we search the intersection of the circle around $B$ through $C$ and the line $DB$. The infinite straight line $DB$ intercepts the circle but with **TikZ** we need to extend the lines $DB$ and that can be done using partway calculations. We get the point $F$ and $BF$ or $DF$ intercepts the circle

\begin{verbatim}
\node (H) [label=135:$H$,draw,circle through=(C)] at (B) {};
\path let \p1 = ($ (B) - (C) $) in 
  coordinate [label=left:$G$] (G) at ($ (B) ! veclen(\x1,\y1) ! (F) $);
\fill[red,opacity=.5] (G) circle (2pt);
\end{verbatim}
Like the intersection of two circles, it’s easy to find the intersection of a line and a circle with \texttt{tkz-euclide}.

We don’t need F

\begin{verbatim}
\tkzInterLC(B,D)(B,C)\tkzGetFirstPoint{G}
\end{verbatim}

There are no more difficulties. Here the final code with some simplifications. Nous traçons le cercle $\mathcal{K}$ de centre D et passant par G. Il coupe la droite AD au point L. $AL = BC$.

\begin{verbatim}
\tkzDrawCircle(D,G)
\tkzInterLC(D,A)(D,G)\tkzGetSecondPoint{L}
\end{verbatim}

\begin{proposition}
To place a straight line equal to a given straight line with one end at a given point.
Let A be the given point, and BC the given straight line
\end{proposition}
\textbf{Proposition II} \linebreak \textit{To place a straight line equal to a given straight line with one end at a given point.}\linebreak Let $\text{A}$ be the given point, and $\text{B C}$ the given straight line \dots

\textbf{Example III} \linebreak

\textit{To place a straight line equal to a given straight line with one end at a given point.}

\textbf{Explanation} \linebreak
From Wikipedia: \textit{Apollonius showed that a circle can be defined as the set of points in a plane that have a specified ratio of distances to two fixed points, known as foci. This Apollonian circle is the basis of the Apollonius pursuit problem. The solutions to this problem are sometimes called the circles of Apollonius. A circle is the set of points in a plane that are equidistant from a given point $O$. The distance $r$ from the center is called the radius, and the point $O$ is called the center. It is the simplest definition but it is not the only one. Apollonius of Perga gives another definition: The set of all points whose distances from two fixed points are in a constant ratio is a circle. With \texttt{tkz-euclide} is easy to show you the last definition.}

\textbf{The code and the analyse} \linebreak
\documentclass{standalone} \linebreak
% Excellent class to show the result and to verify the bounding box.
The document contains a LaTeX code for a TikZ picture that demonstrates the use of tkz-euclide for creating geometric diagrams. The code defines points, circles, and segments, and labels them accordingly. The TikZ picture is centered around a fixed point and an apollonius circle with a given ratio, followed by drawing segments and labeling the points involved.
The result
Example IV

The Apollonius circle of a triangle – Apollonius

To place a straight line equal to a given straight line with one end at a given point.

Explanation

The purpose of the first examples was to show the simplicity with which we could recreate these propositions. With TikZ you need to do calculations and use trigonometry while with \texttt{tkz-euclide} you only need to build simple objects.

But don’t forget that behind or far above \texttt{tkz-euclide} there is TikZ. I’m only creating an interface between TikZ and the user of my package.

The last example is very complex and it is to show you all that we can do with \texttt{tkz-euclide}.

The code and the analyse

\begin{verbatim}
%!TEX TS-program = lualatex
\documentclass{standalone}
\usepackage{tkz-euclide}
\begin{document}
\begin{tikzpicture} \[scale=.6,old paper\]
\tkzDefPoints{0/0/A,6/0/B,0.8/4/C}
\tkzDefTriangleCenter[euler](A,B,C) \tkzGetPoint{N}
\tkzDefTriangleCenter[circum](A,B,C) \tkzGetPoint{O}
\tkzDefTriangleCenter[lemoine](A,B,C) \tkzGetPoint{K}
\tkzDefTriangleCenter[ortho](A,B,C) \tkzGetPoint{H}
\tkzDefSpcTriangle[excentral,name=J](A,B,C){a,b,c}
\tkzDefSpcTriangle[centroid,name=M](A,B,C){a,b,c}
\tkzDefCircle[in](Ma,Mb,Mc) \tkzGetPoint{Sp} \% Sp Spieker center
\tkzDefProjExcenter[name=J](A,B,C)(a,b,c){Y,Z,X}
\tkzDefLine[parallel=through Za](A,B) \tkzGetPoint{Xc}
\tkzDefLine[parallel=through Zc](B,C) \tkzGetPoint{Ya}
\tkzDefLine[parallel=through Za](B,C) \tkzGetPoint{Yc}
\tkzInterLL(K,O)(N,Sp) \tkzGetPoint{Q}
\tkzInterLC(A,B)(Q,Cb) \tkzGetFirstPoint{Ba}
\tkzInterLC(A,C)(Q,Cb) \tkzGetPoints{Ac}{Ca}
\tkzInterLC(B,C')(Q,Cb) \tkzGetFirstPoint{Bc}
\tkzInterLC[next to=Ja](Ja,Q)(Q,Cb) \tkzGetFirstPoint{F'a}
\tkzInterLC[next to=Jc](Jc,Q)(Q,Cb) \tkzGetFirstPoint{F'c}
\tkzInterLC[next to=Jb](Jb,Q)(Q,Cb) \tkzGetFirstPoint{F'b}
\tkzInterLC[common=F'a](Sp,F'a)(Ja,F'a) \tkzGetFirstPoint{Fa}
\tkzInterLC[common=F'b](Sp,F'b)(Jb,F'b) \tkzGetFirstPoint{Fb}
\tkzInterLC[common=F'c](Sp,F'c)(Jc,F'c) \tkzGetFirstPoint{Fc}
\tkzInterLC(Mc,Sp)(Q,Cb) \tkzGetFirstPoint{A''}
\end{tikzpicture}
\end{document}
\end{verbatim}

9
The result