# Euclidean Geometry with tkz-euclide

Alain Matthes

January 23, 2020

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.

## Book I, proposition I

### Proposition I

*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 orange and that of tkz-euclide is red.

```
\usepackage{tikz}
\usetikzlibrary{calc,intersections,through,backgrounds}

\usepackage{tkz-euclide}
```

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

```
\coordinate (A) at (0,0);
\coordinate (B) at (1.25,0.25); \draw[red] (A) -- (B);

\tkzDefPoint(0,0){A}
\tkzDefPoint(1.25,0.25){B}
\tkzDrawSegment(A,B)
```

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

```
\coordinate [label=left:$A$] (A) at (0,0);
\coordinate [label=right:$B$] (B) at (1.25,0.25);
\draw (A) -- (B);
\draw let \p1 = ($ (B) - (A) $),
\n2 = {veclen(\x1,\y1)} in
          (A) circle (\n2)
          (B) circle (\n2);
```

```
\tkzDefPoint(0,0){A}
\tkzDefPoint(1.25,0.25){B}
\tkzDrawSegment(A,B)
\tkzDrawCircles(A,B B,A)
\tkzLabelPoint[left](A){$A$}
\tkzLabelPoint[right](B){$B$}
```

The intersection of the circles

```
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) {};
```

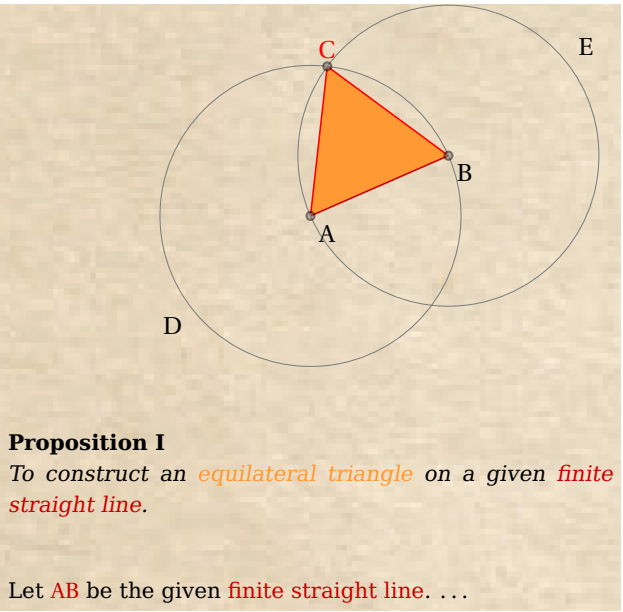
```
\tkzInterCC(A,B)(B,A) \tkzGetPoints{C}{X}
```

How to draw points :

```
\foreach \point in {A,B,C}
\fill [black,opacity=.5] (\point) circle (2pt);
```

```
 \tkzDrawPoints[fill=gray,opacity=.5](A,B,C)
```

**The complete code with tkz-euclide**




**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){$D$}
\tkzLabelCircle[above=12pt](B,A)(180){$E$}
\tkzLabelPoint[above,red](C){$C$}
\tkzDrawPoints[fill=gray,opacity=.5](A,B,C)

\tkzText[text width=8cm,align=justify](0,-4){%
   \small\textbf{Proposition I}\\
   \emph{To construct an \textcolor{triangle}{equilateral triangle}
   on a given \textcolor{input}{finite straight line}.}
   \\
     \vskip1em
   Let \A\B\ be the given \textcolor{input}{finite straight line}. \dots
}
\end{tikzpicture}
```

## Book I, Proposition II

### Proposition II

*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

```
\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) $) {};
```

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.

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

The 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.

```
\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[equilateral](A,B) \tkzGetPoint{D}
```

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

```
\tkzDrawPolygon{A,B,C}
```

We know how to draw the circle around B through C and how to place the points E and F

```
\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);
```

```
\tkzDrawCircle(B,C)
\tkzDrawLines[add=0 and 2](D,A D,B)
```

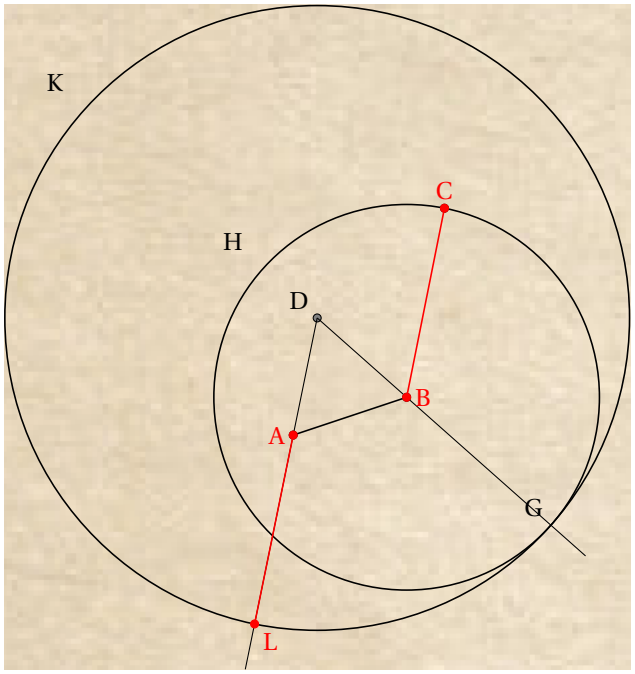
We can place the points E and F at the end of the picture. We don't need them now.
Intersecting a Line and a Circle : here we search the intersection of the circle around B through C and the line DB. The infinite straight line DB intercepts the circle but with TikZ we need to extend the lines DB and that can be done using partway calculations. We get the point F and BF or DF intercepts the circle

```
 \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);
```

Like the intersection of two circles, it's easy to find the intersection of a line and a circle with tkz-euclide. We don't need F

```
 \tkzInterLC(B,D)(B,C)\tkzGetFirstPoint{G}
```

there are no more difficulties



```
\begin{tikzpicture}[scale=2,old paper] % remove old paper style !
\tkzDefPoint(0,0){A}
\tkzDefPoint(0.75,0.25){B}
\tkzDefPoint(1,1.5){C}
\tkzDefMidPoint(A,B) \tkzGetPoint{X}
\tkzDefTriangle[equilateral](A,B) \tkzGetPoint{D}
\tkzDrawSegment(A,B)
\tkzDrawPoints[fill=gray](A,B,D,C)
\tkzLabelPoints[left,red](A)
\tkzDrawCircle(B,C)
\tkzDrawLines[add=0 and 2](D,A D,B)
\tkzInterLC(B,D)(B,C)\tkzGetFirstPoint{G}
\tkzLabelPoints[above left](D,G)
\tkzLabelPoints[above,red](C)\tkzLabelPoints[right,red](B)
\tkzDrawCircle(D,G)
\tkzInterLC(D,A)(D,G)\tkzGetSecondPoint{L}
\tkzLabelPoints[below right,red](L)
\tkzDrawPoints[red](B,C)
\tkzDrawPoints[red](A,L)
\tkzDrawSegment[red](A,L)
\tkzDrawSegment[red](B,C)
\tkzLabelCircle[above left=6pt](B,G)(180){$H$}
\tkzLabelCircle[above left=6pt](D,G)(180){$K$}
\end{tikzpicture}
```

## Apollonius' definition of a circle

From Wikipedia : *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 tkz-euclide is easy to show you the last definition

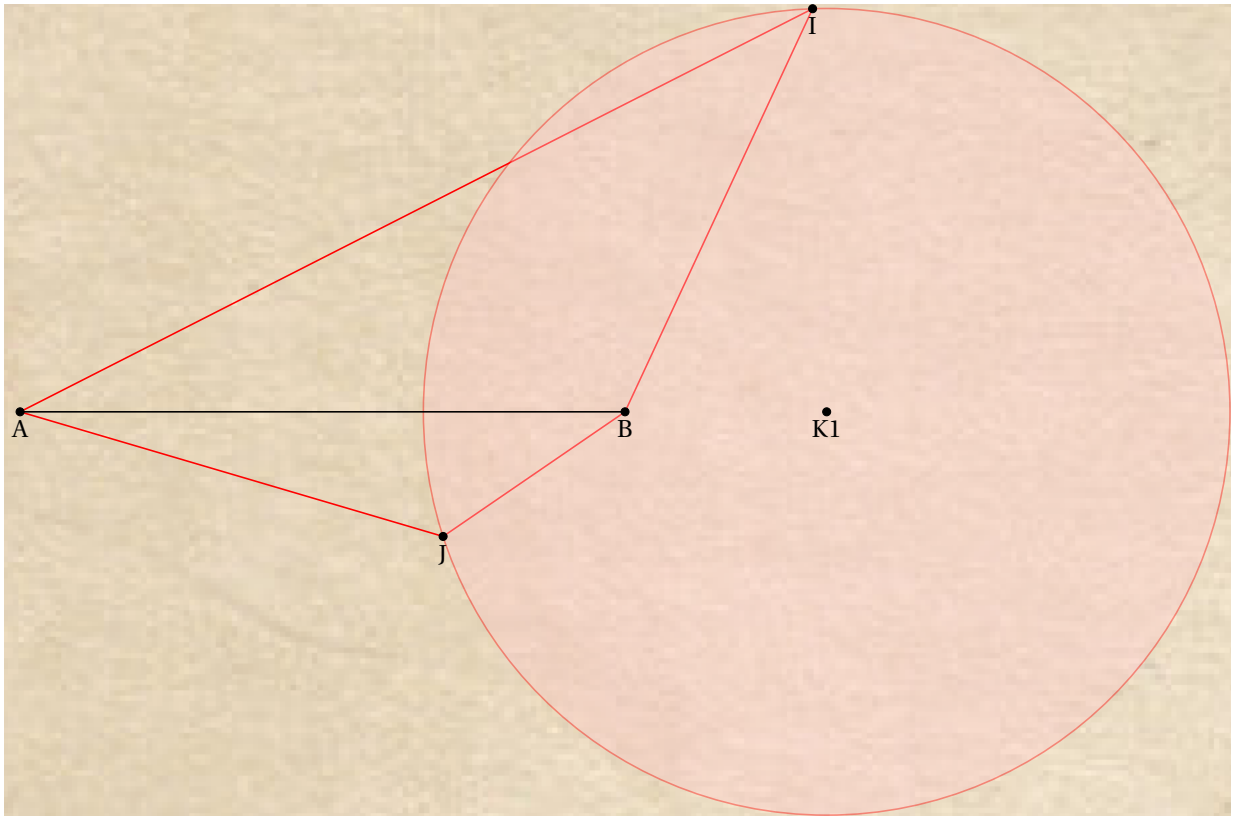
### The code and the analyse

```
\documentclass{standalone}
% Excellent class to show the result and to verify the bounding box.
\usepackage{tkz-euclide}
% no need to use \usetkzobj !
\begin{document}

\begin{tikzpicture}
  % no need to use \tkzInit and \tkzClip
  % Firstly we defined two fixed point. The figure depends of these points and the ratio K
  \tkzDefPoint(0,0){A}
  \tkzDefPoint(4,0){B}
  % tkz-euclide knows about the apollonius's circle
  % with K=2 we search some points like  I such as IA=2 IB
  \tkzDefCircle[apollonius,K=2](A,B) \tkzGetPoint{K1}
  % K1 is the center of the circle
```

```
  \tkzGetLength{rAp} % we get also the radius
  % We can find random points of the circle
  \tkzDefRandPointOn[circle = center K1 radius \rAp pt] \tkzGetPoint{I}
  \tkzDrawSegments[red](A,I I,B)% drawing of some segments
  \tkzDefRandPointOn[circle = center K1 radius \rAp pt] \tkzGetPoint{J}
  \tkzDrawSegments[red](A,J J,B)
  \tkzDrawCircle[R,color = red,fill=red!20,opacity=.4](K1,\rAp pt)
  \tkzLabelPoints[below,font=\scriptsize](A,B,K1,I,J)
  \tkzDrawPoints(A,B,K1,I,J)
  \tkzDrawSegment(A,B)
\end{tikzpicture}
\end{document}
```

**The result**



## The Apollonius circle of a triangle

*The Apollonius circle of a triangle is the circle tangent internally to each of the three excircles.*

The purpose of the first two 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 tkz-euclide you only need to build simple objects
But don't forget that behind or far above 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 tkz-euclide. Unfortunately the only limitation is the accuracy of the calculations.

### The code and the analyse

```
% !TEX TS-program = lualatex-dev
\documentclass{standalone}
\usepackage{tkz-euclide}
\begin{document}

\begin{tikzpicture}[scale=1]
 \tkzDefPoints{0/0/A,6/0/B,0.8/4/C}
 % we need some special points if the triangle,  tkz-euclide knows about them

 \tkzDefTriangleCenter[euler](A,B,C)    \tkzGetPoint{N} % or   \tkzEulerCenter(A,B,C)
 \tkzDefTriangleCenter[circum](A,B,C)   \tkzGetPoint{O} %    \tkzCircumCenter(A,B,C)
 \tkzDefTriangleCenter[lemoine](A,B,C)  \tkzGetPoint{K}
 \tkzDefTriangleCenter[ortho](A,B,C)    \tkzGetPoint{H}

  % \tkzDefSpcTriangle new macro to define new triangle in relation wth ABC
 \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

 % here I used the definition but tkz-euclide knows this point
 % \tkzDefTriangleCenter[spieker](A,B,C)    \tkzGetPoint{Sp}
 % each center has three projections on the sides of the triangle ABC
 % We can do this with one macro

 \tkzDefProjExcenter[name=J](A,B,C)(a,b,c){Y,Z,X}

 % but possible is
 % \tkzDefPointBy[projection=onto A-C ](Ja)  \tkzGetPoint{Za}
 % etc...
 \tkzDefLine[parallel=through Za](A,B)      \tkzGetPoint{Xc}
 \tkzInterLL(Za,Xc)(C,B)                    \tkzGetPoint{C'}
 \tkzDefLine[parallel=through Zc](B,C)      \tkzGetPoint{Ya}
 \tkzInterLL(Zc,Ya)(A,B)                    \tkzGetPoint{A'}
 \tkzDefPointBy[reflection= over Ja-Jc](C')\tkzGetPoint{Ab}
 \tkzDefPointBy[reflection= over Ja-Jc](A')\tkzGetPoint{Cb}

 % Now we can get the center of THE CIRCLE : Q
 % BUT we need to find the radius or a point on the circle

 \tkzInterLL(K,O)(N,Sp)                     \tkzGetPoint{Q}
 \tkzInterLC(A,B)(Q,Cb)                     \tkzGetSecondPoint{Ba}
```

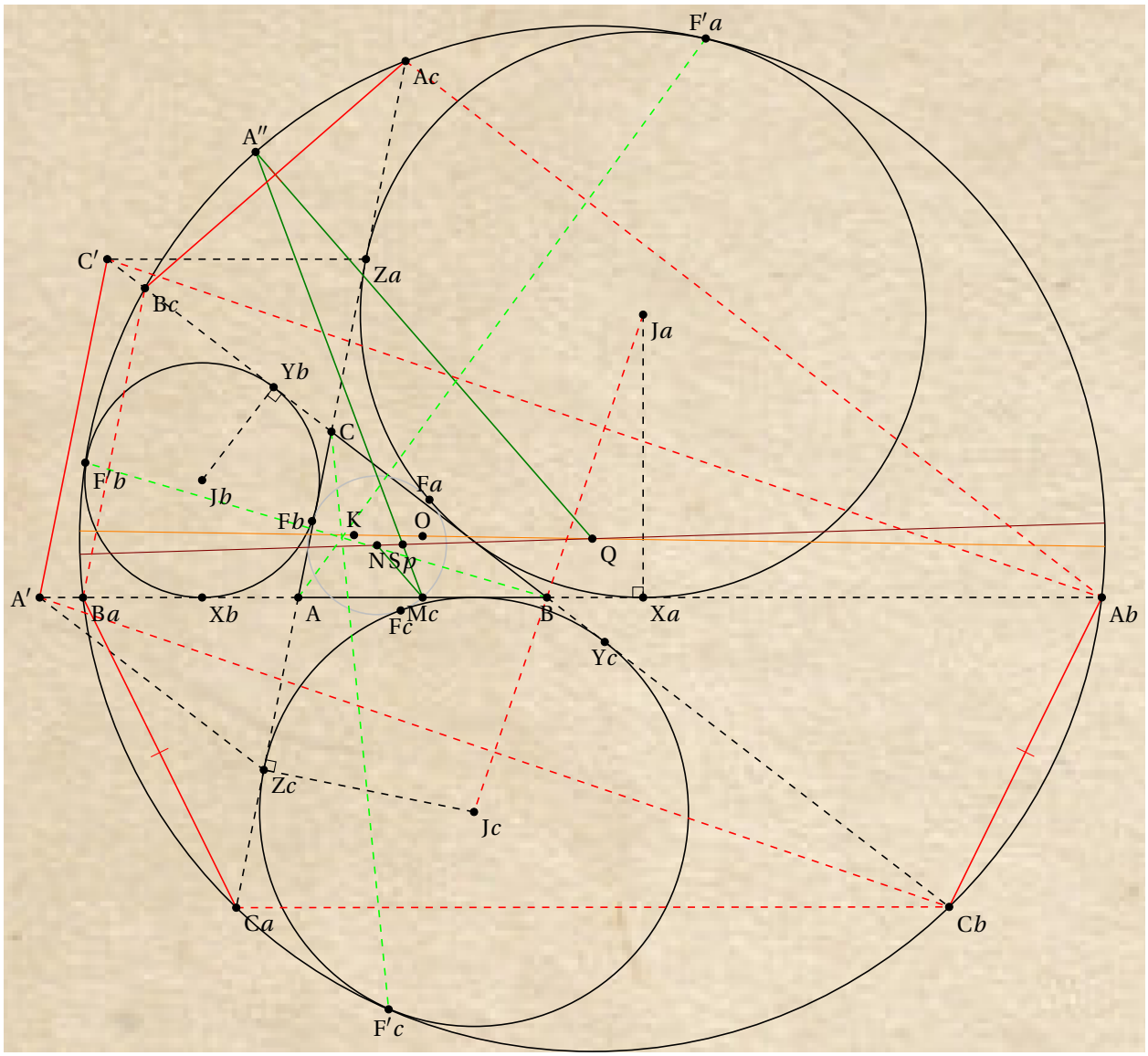
```
 \tkzInterLC(A,C)(Q,Cb)                    \tkzGetPoints{Ca}{Ac}
 \tkzInterLC(B,C')(Q,Cb)                   \tkzGetSecondPoint{Bc}
 \tkzInterLC(Q,Ja)(Q,Cb)                   \tkzGetSecondPoint{F'a}
 \tkzInterLC(Q,Jc)(Q,Cb)                   \tkzGetSecondPoint{F'c}
 \tkzInterLC(Q,Jb)(Q,Cb)                   \tkzGetSecondPoint{F'b}
 \tkzInterLC(Sp,F'a)(Ja,Za)                \tkzGetFirstPoint{Fa}
 \tkzInterLC(Sp,F'b)(Jb,Yb)                \tkzGetFirstPoint{Fb}
 \tkzInterLC(Sp,F'c)(Jc,Yc)                \tkzGetSecondPoint{Fc}
 \tkzInterLC(Mc,Sp)(Q,Cb)                  \tkzGetSecondPoint{A"}
 \tkzDefLine[parallel=through A"](N,Mc)    \tkzGetPoint{q}

 % Calculations are done, now you can draw, mark and label

 \tkzDrawPolygon(A,B,C)
 \tkzDrawCircle(Q,Bc)%
 \tkzDrawCircle[euler,lightgray](A,B,C)
 \tkzDrawCircles[ex](A,B,C B,C,A C,A,B)
 \tkzDrawSegments[dashed](A,A'  C,C' A',Zc Za,C'
                    B,Cb B,Ab A,Ca C,Ac
                    Ja,Xa Jb,Yb Jc,Zc)
 \begin{scope}
    \tkzClipCircle(Q,Cb) % We limit the drawing of the lines
    \tkzDrawLine[add=5 and 12,orange](K,O)
    \tkzDrawLine[add=12 and 28,red!50!black](N,Sp)
 \end{scope}
  \tkzDrawPoints(A,B,C,K,Ja,Jb,Jc,Q,N,O,Sp,Mc,Xa,Xb,Yb,Yc,Za,Zc,
                 A',C',A",Ab,Cb,Bc,Ca,Ac,Ba,Fa,Fb,Fc,F'a,F'b,F'c)
 \tkzLabelPoints(Ja,Jb,Jc,Q,Xa,Xb,Za,Zc,Ab,Cb,
                 Bc, Ca, Ac, Ba, F'b)
 \tkzLabelPoints[above](O, K, F'a, Fa, A")
 \tkzLabelPoints[below](B, F'c, Yc, N, Sp, Fc, Mc)
 \tkzLabelPoints[left](A', C', Fb)
 \tkzLabelPoints[right](C)
 \tkzLabelPoints[below right](A)
 \tkzLabelPoints[above right](Yb)
 \tkzDrawSegments[color=green!50!black](Mc,N Mc,A" A",Q)
 \tkzDrawSegments[color=red,dashed](Ac,Ab Ca,Cb Ba,Bc Ja,Jc A',Cb C',Ab)
 \tkzDrawSegments[color=red](Cb,Ab Bc,Ac Ba,Ca A',C')
 \tkzMarkSegments[color=red,mark=|](Cb,Ab Bc,Ac Ba,Ca)
 \tkzMarkRightAngles(Jc,Zc,A Ja,Xa,B Jb,Yb,C)
 \tkzDrawSegments[green,dashed](A,F'a B,F'b C,F'c)
\end{tikzpicture}

\end{document}
```

**The result**


F′a
Ac
A″
C′
Za
Bc
Ja
Yb
C
F′b
Jb
Fa
Fb
K
O
Q
N
S
p
A′
Ba
Xb
A
Mc
Fc
B
Xa
Ab
Yc
Zc
Jc
Ca
Cb
F′c