

The `pst-euclide` Package

Version 1.46

Dominique Rodriguez¹

jan 9th 2013

¹domino.rodriguez@laposte.net

Abstract

The `pst-eucl` package allow the drawing of Euclidean geometric figures using \LaTeX macros for specifying mathematical constraints. It is thus possible to build point using common transformations or intersections. The use of coordinates is limited to points which controlled the figure.

Acknowledgements

I would like to thanks the following persons for the help they gave me for development of this package:

- Denis Girou pour ses critiques pertinentes et ses encouragement lors de la découverte de l’embryon initial et pour sa relecture du présent manuel ;
- Michael Vulis for his fast testing of the documentation using \LaTeX which leads to the correction of a bug in the `PostScript` code;
- Manuel Luque and Olivier Reboux for their remarks and their examples.
- Alain Delplanque for its modification propositions on automatic placing of points name and the ability of giving a list of points in `\pstGeonode`.

WARNING

This is the first release put on CTAN archives.

LICENSE

This program and its documentation can be redistributed and/or modified under the terms of the “ \LaTeX Project Public License” Distributed from CTAN archives in directory `macros/latex/base/lppl.txt`. However, you may send me an Email with a small commentary. Then you should consider making a donation¹:

1. directly to the \LaTeX 3 team;
2. and/or to me for the support of this package².

A donation of time depending of competences is possible : correction of the documentation (especially this one), test of fonctionnalités, propositions of extensions, ...

¹especially if you use a purchased operating system!. Furthermore, do not forget that \LaTeX is freely usable and that many users buy several hundreds of euros (dollars, pounds) softwares of lower quality

²1 €, £1 ou \$1 is OK, but I accept more.

Contents

1	User's manual	2			
1.1	Special specifications	2	1.5.4	Function–function	17
1.1.1	PSTricks Options	2	1.5.5	Function–line	18
1.1.2	Conventions	2	1.5.6	Function–circle	18
1.2	Basic Objects	2	2	Examples gallery	19
1.2.1	Points	2	2.1	Basic geometry	19
1.2.2	Segment mark	4	2.1.1	Drawing of the bissector	19
1.2.3	Triangles	5	2.1.2	Transformation de poly- gones et courbes	19
1.2.4	Angles	6	2.1.3	Triangle lines	21
1.2.5	Lines, half-lines and segments	7	2.1.4	Euler circle	22
1.2.6	Circles	7	2.1.5	Orthocenter and hyper- bola	23
1.2.7	Circle arcs	8	2.1.6	17 sides regular polygon	24
1.2.8	Curved abscissa	8	2.1.7	Circles & tangents	25
1.2.9	Généric curve	9	2.1.8	Fermat's point	25
1.3	Geometric Transformations	9	2.1.9	Escribed and inscribed circles of a triangle	27
1.3.1	Central symmetry	10	2.2	Some locus points	28
1.3.2	Orthogonal (or axial) symmetry	10	2.2.1	Parabola	28
1.3.3	Rotation	10	2.2.2	Hyperbola	29
1.3.4	Translation	11	2.2.3	Cycloid	30
1.3.5	Homothetie	11	2.2.4	Hypocycloids (Astroid and Deltoid)	31
1.3.6	Orthogonal projection	12	2.3	Lines and circles envelope	32
1.4	Special object	12	2.3.1	Conics	32
1.4.1	Midpoint	12	2.3.2	Cardioid	33
1.4.2	Triangle center of gravity	13	2.4	Homotethy and fractals	34
1.4.3	Centre of the circumcir- cle of a triangle	13	2.5	hyperbolic geometry: a triangle and its altitudes	35
1.4.4	Perpendicular bisector of a segment	13	A	Glossaire des commandes	36
1.4.5	Bisectors of angles	14	B	Compatibilité ascendantes de pst-eucl	38
1.5	Intersections	15			
1.5.1	Line-Line	15			
1.5.2	Circle-Line	16			
1.5.3	Circle-Circle	16			

Chapter 1

User's manual

1.1 Special specifications

1.1.1 PSTricks Options

The package activates the `\SpecialCoor` mode. This mode extend the coordinates specification. Furthermore the plotting type is set to `dimen=middle`, which indicates that the position of the drawing is done according to the middle of the line. Please look at the user manual for more information about these setting.

At last, the working axes are supposed to be (ortho)normed.

1.1.2 Conventions

For this manual, I used the geometric French conventions for naming the points:

- O is a centre (circle, axes, symmetry, homothety, rotation);
- I defined the unity of the abscissa axe, or a midpoint;
- J defined the unity of the ordinate axe;
- A, B, C, D are points ;
- M' is the image of M by a transformation ;

At last, although these are nodes in `PSTricks`, I treat them intentionally as points.

1.2 Basic Objects

1.2.1 Points

default axes

`\pstGeonode[$\langle par \rangle$](x_1, y_1){ $\langle A_1 \rangle$ }(x_2, y_2){ $\langle A_2 \rangle$ }\dots(x_n, y_n){ $\langle A_n \rangle$ }`

This command defines one or more geometrical points associated with a node. Each point has a node name $\langle A_i \rangle$ which defines the default label put on the picture. This label is managed by default in mathematical mode, the boolean parameter `PtNameMath` (by default `true`) can modify this behavior and let manage the label in normal mode. It is placed at a distance of `PointNameSep` (by default `1em`) of the center of the node with a angle of `PosAngle` (by default `0`). It is possible to specify another label using the parameter `PointName` (by default `default`), and an empty label can be specified by selecting the value `none`, in that case the point will have no name on the picture.

The point symbol is given by the parameter `PointSymbol` (by default `*`). The symbol is the same as used by the macro `\pstdot`. This parameter can be set to `none`, which means that the point will not be drawn on the picture.

Here are the possible values for this parameter:

- | | | |
|-----------------------------|------------------------------|------------------------------|
| • <code>*</code> : ● | • <code>otimes</code> : ⊗ | • <code>diamond*</code> : ◆ |
| • <code>o</code> : ○ | • <code>triangle</code> : △ | |
| • <code>+</code> : + | • <code>triangle*</code> : ▲ | • <code>pentagon</code> : ⬠ |
| • <code>x</code> : × | • <code>square</code> : □ | • <code>pentagon*</code> : ⬡ |
| • <code>asterisk</code> : * | • <code>square*</code> : ■ | |
| • <code>oplus</code> : ⊕ | • <code>diamond</code> : ◇ | • <code> </code> : |

Furthermore, these symbols can be controlled with some others `PSTricks`, several of these are :

- their scale with `dotscale`, the value of whom is either two numbers defining the horizontal and vertical scale factor, or one single value being the same for both,
- their angle with parameter `dotangle`.

Please consult the `PSTricks` documentation for further details.

The parameters are specified explicitly in the $\langle par \rangle$ part. The parameters `PosAngle`, `PointSymbol`, `PointName` and `PointNameSep` can be set to :

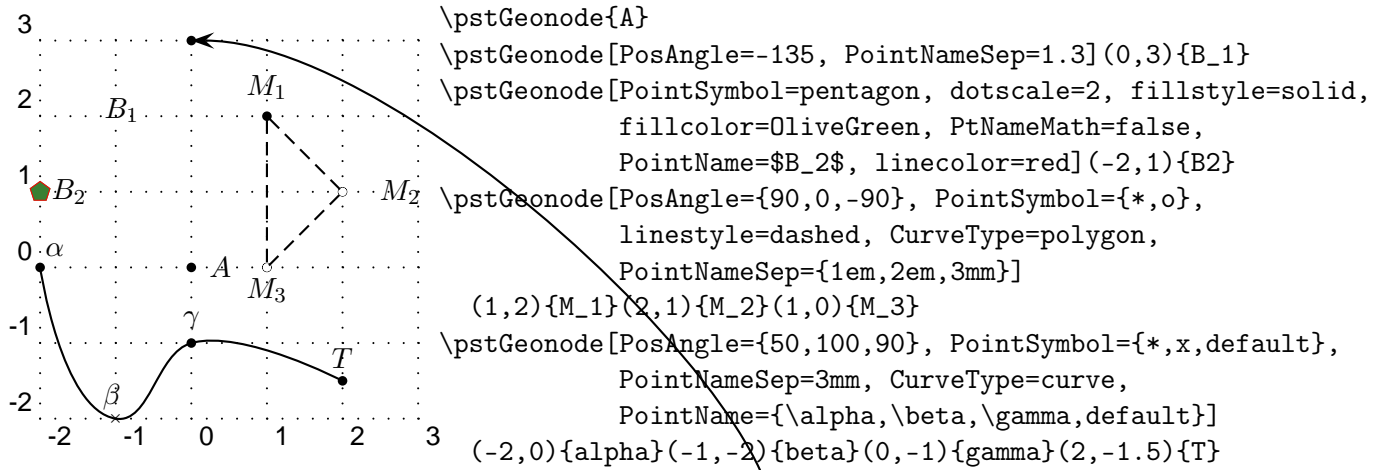
- either a single value, the same for all points ;
- or a list of values delimited by accolads `{ ... }` and separated with comma *without any blanks*, allowing to differentiate the value for each point.

In the later case, the list can have less values than point which means that the last value is used for all the remaining points.

At least, the parameter `CurveType` (by default `none`) can be used to draw a line between the points:

- opened `polyline` ;
- closed `polygon` ;

- open and curved curve.

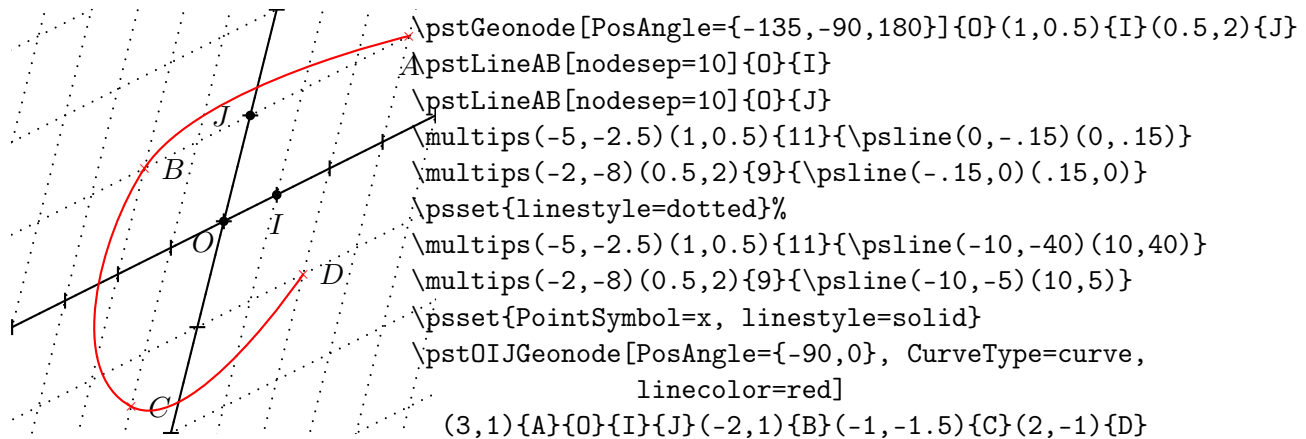


Obviously, the nodes appearing in the picture can be used as normal PSTricks nodes. Thus, it is possible to reference a point from here:

User defined axes

```
\pstOIJGeonode[⟨par⟩](x1,y1){⟨A1⟩}{⟨O⟩}{⟨I⟩}{⟨J⟩}(x2,y2){⟨A2⟩}... (xn,yn){⟨An⟩}
```

This command allows the placement of points in any landmark(?) defined by the three points (O; I; J).



1.2.2 Segment mark

A segment can be drawn using the `\ncline` command. However, for marking a segment there is the following command:

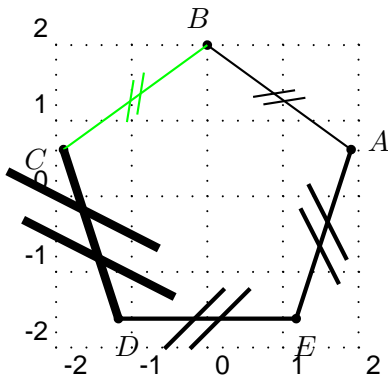
```
\pstMarkSegment[⟨par⟩]{⟨A⟩}{⟨B⟩}
```

The symbol drawn on the segment is given by the parameter `SegmentSymbol`. Its value can be any valid command which can be used in math mode. Its default value is `pstslashh`, which produced two slashes on the segment. The segment is drawn.

Several commands are predefined for marking the segment:

• <code>\pstslash</code> :		• <code>\MarkHash</code> :		• <code>\MarkCros</code> :	
• <code>\pstslashh</code> :		• <code>\MarkHashh</code> :		• <code>\MarkCross</code> :	
• <code>\pstslashhh</code> :		• <code>\MarkHashhh</code> :			

The three commands of the family `MarkHash` draw a line whose inclination is controlled by the parameter `MarkAngle` (by default 45). Their width and colour depends of the width and color of the line when the drawing is done, as shown is the next example.



```
\rput{18}{%
  \pstGeonode[PosAngle={0,90,180,-90}](2,0){A}(2;72){B}
  (2;144){C}(2;216){D}(2;288){E}}
\pstSegmentMark{A}{B}
\pstSegmentMark[linecolor=green]{B}{C}
\psset{linewidth=2\pslinewidth}
\pstSegmentMark[linewidth=2\pslinewidth]{C}{D}
\pstSegmentMark{D}{E}
\pstSegmentMark{E}{A}
```

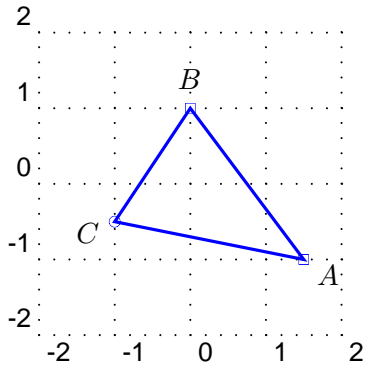
1.2.3 Triangles

The more classical figure, it has its own macro for a quick definition:

`\pstTriangle[par]($x_A; y_A$){ $\langle A \rangle$ }($x_B; y_B$){ $\langle B \rangle$ }($x_C; y_C$){ $\langle C \rangle$ }`

In order to accurately put the name of the points, there are three parameters `PosAngleA`, `PosAngleB` and `PosAngleC`, which are associated respectively to the nodes $\langle A \rangle$, $\langle B \rangle$ et $\langle C \rangle$. Obviously they have the same meaning as the parameter `PosAngle`. If one or more of such parameters is omitted, the value of `PosAngle` is taken. If no angle is specified, points name are placed on the bissector line.

In the same way there are parameters for controlling the symbol used for each points: `PointSymbolA`, `PointSymbolB` and `PointSymbolC`. They are equivalent to the parameter `PointSymbol`. The management of the default value followed the same rule.



```
\pstTriangle[PointSymbol=square, PointSymbolC=o,
  linecolor=blue, linewidth=1.5\pslinewidth]
(1.5,-1){A}(0,1){B}(-1,-.5){C}
```

1.2.4 Angles

Each angle is defined with three points. The vertex is the second point. Their order is important because it is assumed that the angle is specified in the direct order. The first command is the marking of a right angle:

```
\pstRightAngle[⟨par⟩]{⟨A⟩}{⟨B⟩}{⟨C⟩}
```

The symbol used is controlled by the parameter `RightAngleType` (by default `default`). Its possible values are :

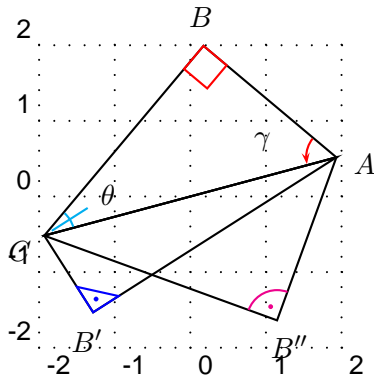
- `default` : standard symbol ;
- `german` : german symbol (given by U. Dirr) ;
- `suissromand` : swiss romand symbol (given P. Schneulin).

The only parameter controlling this command, excepting the ones which controlled the line, is `RightAngleSize` which defines the size of the symbol (by default 0.28 unit).

For other angles, there is the command:

```
\pstMarkAngle[⟨par⟩]{⟨A⟩}{⟨B⟩}{⟨C⟩}
```

The `label` can be any valid \TeX box, it is put at `LabelSep` (by default 1 unit) of the node in the direction of the bisector of the angle modified by `LabelAngleOffset` (by default 0) and positioned using `LabelRefPt` (by default `c`). Furthermore the arc used for marking has a radius of `MarkAngleRadius` (by default .4 unit). At least, it is possible to place an arrow using the parameter `arrows`. Finally, it is possible to mark the angle by specifying a \TeX command as argument of parameter `Mark`.



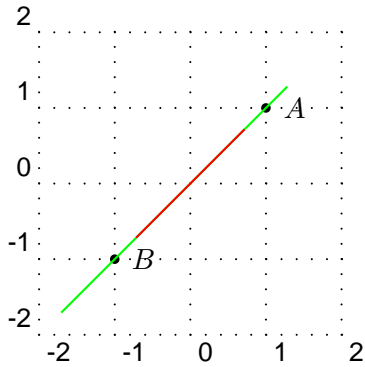
```
\psset{PointSymbol=none}
\pstTriangle(2;15){A}(2;85){B}(2;195){C}
\psset{PointName=none}
\pstTriangle[PointNameA=default](2;-130){B'}(2;15){A'}(2;195){C'}
\pstTriangle[PointNameA=default](2;-55){B''}(2;15){A''}(2;195){C''}
\pstRightAngle[linecolor=red]{C}{B}{A}
\pstRightAngle[linecolor=blue, RightAngleType=suissromand]{A}{B'}{C}
\pstRightAngle[linecolor=magenta, RightAngleType=german]{A}{B''}{C}
\psset{arcsep=\pslinewidth}
\pstMarkAngle[linecolor=cyan, Mark=MarkHash]{A}{C}{B}{\theta}
\pstMarkAngle[linecolor=red, arrows=->]{B}{A}{C}{\gamma}
```

1.2.5 Lines, half-lines and segments

The classical line!

$\backslash\text{pstLineAB}[\langle par \rangle]\{\langle A \rangle\}\{\langle B \rangle\}$

In order to control its length¹, the two parameters `nodesepA` et `nodesepB` specify the abscissa of the extremity of the drawing part of the line. A negative abscissa specify an outside point, while a positive abscissa specify an internal point. If these parameters have to be equal, `nodesep` can be used instead. The default value of these parameters is equal to 0.



```
\pstGeonode(1,1){A}(-1,-1){B}
\pstLineAB[nodesepA=-.4, nodesepB=-1, linecolor=green]{A}{B}
\pstLineAB[nodesep=.4, linecolor=red]{A}{B}
```

1.2.6 Circles

A circle can be defined either with its center and a point of its circumference, or with two diametrically opposed points. There is two commands :

$\backslash\text{pstCircleOA}[\langle par \rangle]\{\langle O \rangle\}\{\langle A \rangle\}$

$\backslash\text{pstCircleAB}[\langle par \rangle]\{\langle A \rangle\}\{\langle B \rangle\}$

For the first macro, it is possible to omit the second point and then to specify a radius or a diameter using the parameters `Radius` and `Diameter`. The values of these parameters must be specified with one of the two following macros :

$\backslash\text{pstDistAB}[\langle par \rangle]\{\langle A \rangle\}\{\langle B \rangle\}$

$\backslash\text{pstDistVal}[\langle par \rangle]\{\langle x \rangle\}$

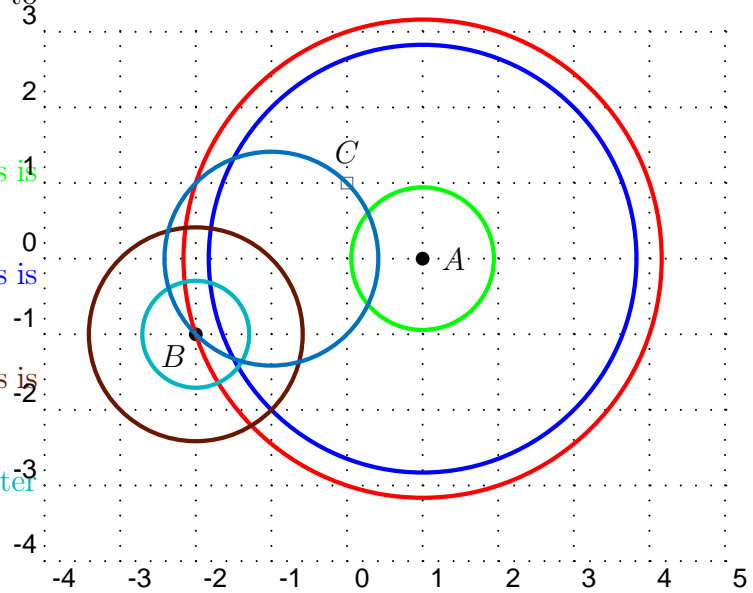
The first specifies a distance between two points. The parameter `DistCoef` can be used to specify a coefficient to reduce or enlarge this distance. To be taken into account this last parameter must be specified before the distance. The second macro can be used to specify an explicit numeric value.

We will see later how to draw the circle crossing three points.

¹which is the comble for a line!

With this package, it becomes possible to draw:

- the circle of center A crossing B ;
- the circle of center A whose radius is AC ;
- the circle of center A whose radius is BC ;
- the circle of center B whose radius is AC ;
- the circle of center B of diameter AC ;
- the circle whose diameter is BC .



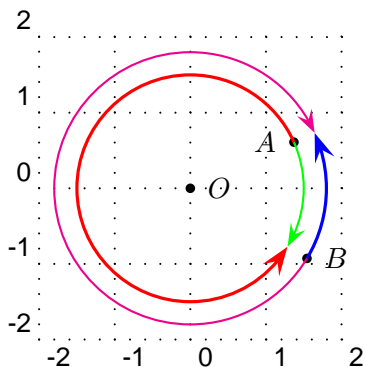
```
\psset{linewidth=2\pslinewidth}
\pstGeonode[PosAngle={0,-135,90},PointSymbol={*,*,square}](1,0){A}(-2,-1){B}(0,1){C}
\pstCircleOA[linecolor=red]{A}{B}
\pstCircleOA[linecolor=green,DistCoef=2 3 div,Radius=\pstDistAB{A}{C}]{A}{C}
\pstCircleOA[linecolor=blue,Radius=\pstDistAB{B}{C}]{A}{B}
\pstCircleOA[linecolor=Sepia,Radius=\pstDistAB{A}{C}]{B}{C}
\pstCircleOA[linecolor=Aquamarine,Diameter=\pstDistAB{A}{C}]{B}{A}
\pstCircleAB[linecolor=RoyalBlue]{B}{C}
```

1.2.7 Circle arcs

$$\text{\texttt{\textbackslash pstArcOAB}[\langle par \rangle]\{\langle O \rangle\}\{\langle A \rangle\}\{\langle B \rangle\}}$$

$$\text{\texttt{\textbackslash pstArcnOAB}[\langle par \rangle]\{\langle O \rangle\}\{\langle A \rangle\}\{\langle B \rangle\}}$$

These two macros draw circle arcs, O is the center, the radius defined by OA , the beginning angle given by A and the final angle by B . Finally, the first macro draws the arc in the direct way, whereas the second in the indirect way.



```
\pstGeonode[PosAngle={180,0}](1.5;24){A}(1.8;-31){B}
\pstGeonode{O}
\psset{arrows=->,arrowscale=2}
\pstArcOAB[linecolor=red,linewidth=1.5\pslinewidth]{O}{A}{B}
\pstArcOAB[linecolor=blue,linewidth=1.5\pslinewidth]{O}{B}{A}
\pstArcnOAB[linecolor=green]{O}{A}{B}
\pstArcnOAB[linecolor=magenta]{O}{B}{A}
```

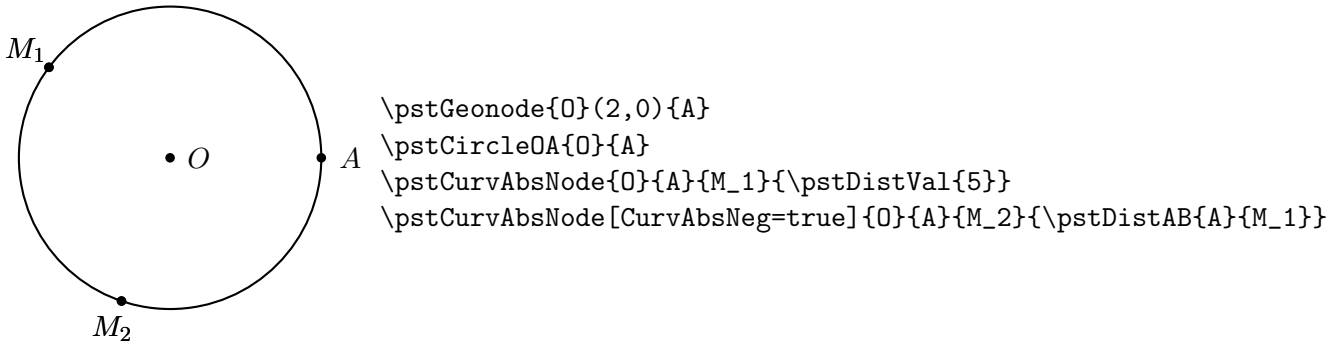
1.2.8 Curved abscissa

A point can be positioned on a circle using its curved abscissa.

`\pstCurvAbsNode[⟨par⟩]{⟨O⟩}{⟨A⟩}{⟨B⟩}{⟨Abs⟩}`

The point $\langle B \rangle$ is positioned on the circle of center $\langle O \rangle$ crossing $\langle A \rangle$, with the curved abscissa $\langle Abs \rangle$. The origin is $\langle A \rangle$ and the direction is anti-clockwise by default. The parameter `CurvAbsNeg` (by default `false`) can change this behavior.

If the parameter `PosAngle` is not specified, the point label is put automatically in order to be aligned with the circle center and the point.

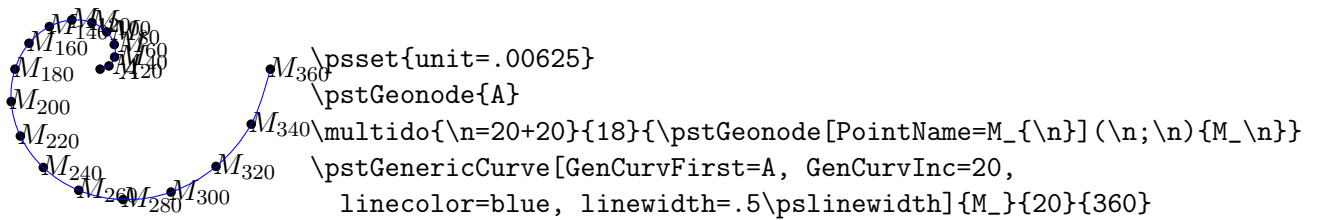


1.2.9 Généric curve

It is possible to generate a set of points using a loop, and to give them a generic name defined by a radical and a number. The following command can draw a interpolated curve crossing all such kind of points.

`\pstGenericCurve[⟨par⟩]{⟨Radical⟩}{⟨n1⟩}{⟨n2⟩}`

The curve is drawn on the points whose name is defined using the radical $\langle Radical \rangle$ followed by a number from $\langle n_1 \rangle$ to $\langle n_2 \rangle$. In order to manage side effect, the parameters `GenCurvFirst` et `GenCurvLast` can be used to specified special first or last point. The parameter `GenCurvInc` can be used to modify the increment from a point to the next one (by default 1).



1.3 Geometric Transformations

The geometric transformations are the ideal tools to construct geometric figures. All the classical transformations are available with the following macros which share the same syntactic scheme end two parameters.

The common syntax put at the end two point lists whose second is optional or with a cardinal at least equal. These two lists contain the antecedent points and their respective images. In the case no image is given for some points the a default name is build appending a ' to the antecedent name.

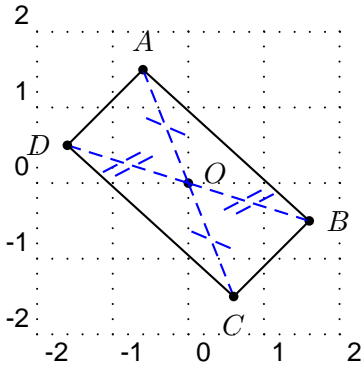
The first shared parameter is `CodeFig` which draws the specific constructions lines. Its default value is `false`, and a `true` value activates this optional drawing. The drawing is done using the line style `CodeFigStyle` (by default `dashed`), with the color `CodeFigColor` (by default `cyan`).

Their second shared parameter is `CurveType` which controls the drawing of a line crossing all images, and thus allow a quick description of a transformed figure.

1.3.1 Central symmetry

$$\backslash\text{pstSym0}[\langle par \rangle][\langle O \rangle][\langle M_1, M_2, \dots, M_n \rangle][\langle M'_1, M'_2, \dots, M'_p \rangle]$$

Draw the symmetric point in relation to point O . The classical parameter of point creation are usable here, and also for all the following functions.

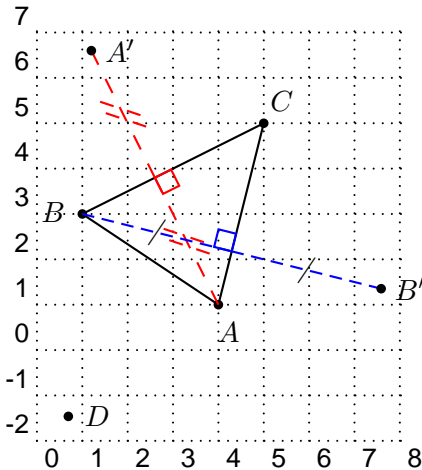


```
\psset{CodeFig=true}
\pstGeonode[PosAngle={20,90,0}]{O}{-0.6,1.5}{A}{1.6,-0.5}{B}
\pstSym0[CodeFigColor=blue, PosAngle={-90,180}]{O}{A, B}{C, D}
%\pstSym0[SegmentSymbol=pstslash, PosAngle=180]
% {O}{B}{D}
\pstLineAB{A}{B}\pstLineAB{C}{D}
\pstLineAB{A}{D}\pstLineAB{C}{B}
```

1.3.2 Orthogonal (or axial) symmetry

$$\backslash\text{pstOrtSym}[\langle par \rangle][\langle A \rangle][\langle B \rangle][\langle M_1, M_2, \dots, M_n \rangle][\langle M'_1, M'_2, \dots, M'_p \rangle]$$

Draws the symmetric point in relation to line (AB) .



```
\pstTriangle(1,3){B}(5,5){C}(4,1){A}
\pstOrtSym{A}{B}{C}{D}
\psset{CodeFig=true}
\pstOrtSym[dash=2mm 2mm, CodeFigColor=red]
{C}{B}{A}
\pstOrtSym[SegmentSymbol=pstslash, linestyle=dotted,
dotsep=3mm, CodeFigColor=blue]
{C}{A}{B}
```

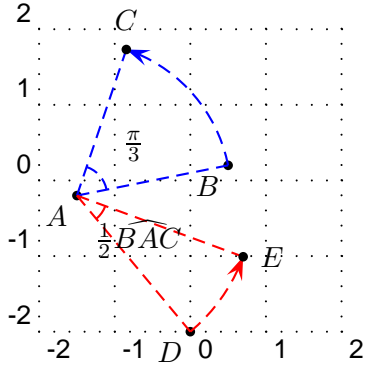
1.3.3 Rotation

$$\backslash\text{pstRotation}[\langle par \rangle][\langle O \rangle][\langle M_1, M_2, \dots, M_n \rangle][\langle M'_1, M'_2, \dots, M'_p \rangle]$$

Draw the image of M_i by the rotation of center O and angle given by the parameter `RotAngle`. This later can be an angle specified by three points. In such a case, the following function must be used:

`\pstAngleABC{<A>}{}{<C>}`

Never forget to use the rotation for drawing a square or an equilateral triangle. The parameter `CodeFig` puts a bow with an arrow between the point and its image, and if `TransformLabel` (by default `none`) contain some text, it is put on the corresponding angle in mathematical mode.

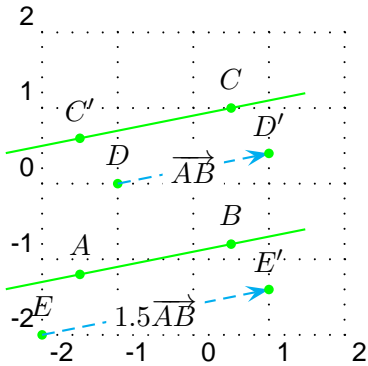


```
\psset{arrowscale=2}
\pstGeonode[PosAngle=-135](-1.5,-.2){A}(.5,.2){B}(0,-2){D}
\pstRotation[PosAngle=90, RotAngle=60, CodeFig=true,
  CodeFigColor=blue,
  TransformLabel=\frac{\pi}{3}{A}{B}{C}]
\pstRotation[AngleCoef=.5, RotAngle=\pstAngleAOB{B}{A}{C},
  CodeFigColor=red, CodeFig=true,
  TransformLabel=\frac{1}{2}\widehat{BAC}]{A}{D}{E}
```

1.3.4 Translation

`\pstTranslation[<par>]{<A>}{}{<M1, M2, ..., Mn>}{<M'1, M'2, ..., M'p>}`

Draws the translated M'_i of M_i using the vector \overrightarrow{AB} . Useful for drawing a parallel line.



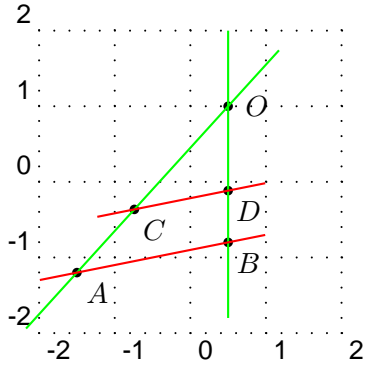
```
\psset{linecolor=green, nodesep=-1, PosAngle=90,arrowscale=2}
\pstGeonode(-1.5,-1.2){A}(.5,-.8){B}(.5,1){C}(-1,0){D}(-2,-2){E}
\pstTranslation{B}{A}{C}
\psset{CodeFig=true, TransformLabel=default}
\pstTranslation{A}{B}{D}
\pstTranslation[DistCoef=1.5]{A}{B}{E}
\pstLineAB{A}{B}\pstLineAB{C}{C'}
```

The parameter `DistCoef` can be used as a multiplicand coefficient to modify the translation vector. The parameter `CodeFig` draws the translation vector le vecteur de translation between the point and its image, labeled in its middle defaultly with the vector name or by the text specified with `TransformLabel` (by default `none`).

1.3.5 Homothetie

`\pstHomO[<par>]{<O>}{<M1, M2, ..., Mn>}{<M'1, M'2, ..., M'p>}`

Draws M'_i the image of M_i by the homotethy of center O and coefficient specified with the parameter `HomCoef`.

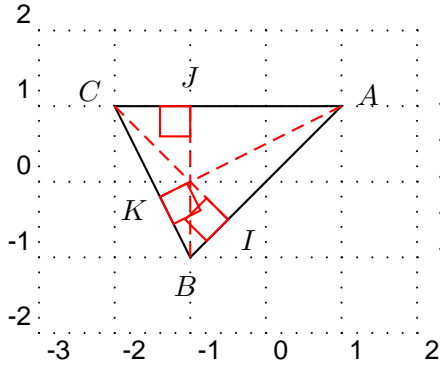


```
\pstGeonode[PosAngle={0,-45}](.5,1){O}(-1.5,-1.2){A}(0.5,-.8){B}
\pstHomO[HomCoef=.62, PosAngle=-45]{O}{A,B}[C,D]
\psset{linecolor=green, nodesep=-1}
\pstLineAB{A}{O}\pstLineAB{B}{O}
\psset{linecolor=red, nodesep=-.5}
\pstLineAB{A}{B}\pstLineAB{C}{D}
```

1.3.6 Orthogonal projection

`\pstProjection[⟨par⟩]{⟨A⟩}{⟨B⟩}{⟨M1, M2, ..., Mn⟩}{⟨M'1, M'2, ..., M'p⟩}`

Projects orthogonally the point M_i on the line (AB) . Useful for the altitude of a triangle. The name is aligned with the point and the projected point as shown in the exemple.



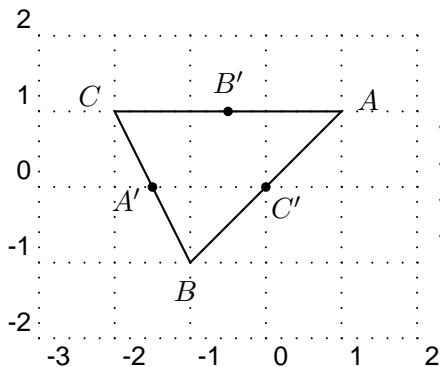
```
\psset{PointSymbol=none, CodeFig=true, CodeFigColor=red}
\pstTriangle(1,1){A}(-2,1){C}(-1,-1){B}
\pstProjection{A}{B}{C}[I]
\pstProjection{A}{C}{B}[J]
\pstProjection{C}{B}{A}[K]
```

1.4 Special object

1.4.1 Midpoint

`\pstMiddleAB[⟨par⟩]{⟨A⟩}{⟨B⟩}{⟨I⟩}`

Draw the midpoint I of segment $[AB]$. By default, the point name is automatically put below the segment.

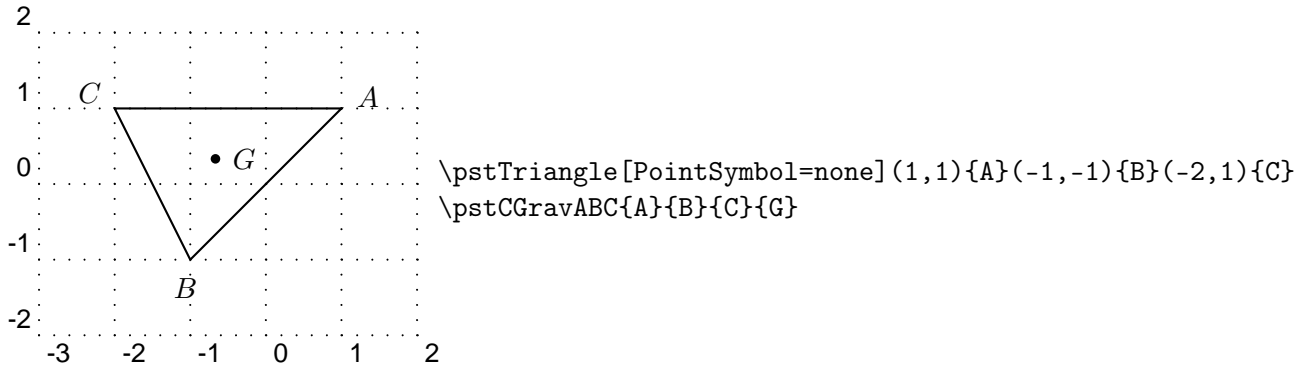


```
\pstTriangle[PointSymbol=none](1,1){A}(-1,-1){B}(-2,1){C}
\pstMiddleAB{A}{B}{C'}
\pstMiddleAB{C}{A}{B'}
\pstMiddleAB{B}{C}{A'}
```

1.4.2 Triangle center of gravity

`\pstCGravABC[⟨par⟩]{⟨A⟩}{⟨B⟩}{⟨C⟩}{⟨G⟩}`

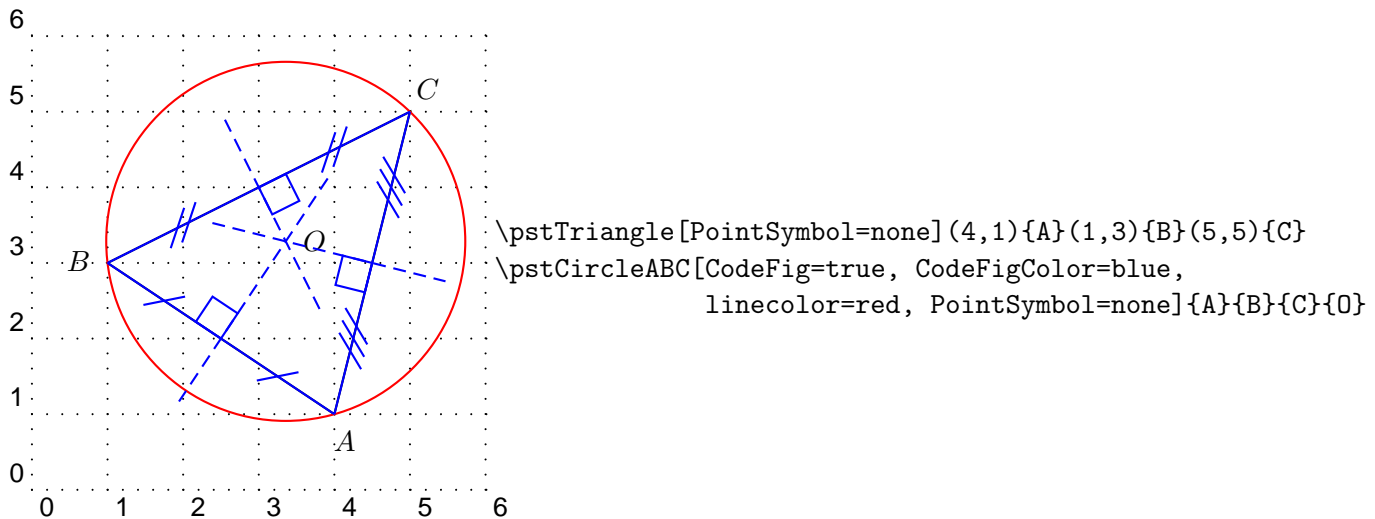
Draw the ABC triangle centre of gravity G .



1.4.3 Centre of the circumcircle of a triangle

`\pstCircleABC[⟨par⟩]{⟨A⟩}{⟨B⟩}{⟨C⟩}{⟨O⟩}`

Draws the circle crossing three points (the circum circle) and put its center O . The effective drawing is controlled by the boolean parameter `DrawCirABC` (by default `true`). Moreover the intermediate constructs (mediator lines) can be drawn by setting the boolean parameter `CodeFig`. In that case the middle points are marked on the segments using three different marks given by the parameters `SegmentSymbolA`, `SegmentSymbolB` et `SegmentSymbolC`.

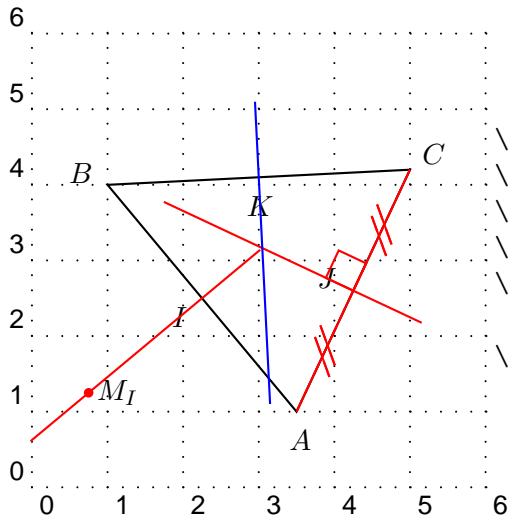


1.4.4 Perpendicular bisector of a segment

`\pstMediatorAB[⟨par⟩]{⟨A⟩}{⟨B⟩}{⟨I⟩}{⟨M⟩}`

The perpendicular bisector of a segment is a line perpendicular to this segment in its midpoint. The segment is $[AB]$, the midpoint I , and M is a point belonging to the perpendicular bisector line. It is built by a rotation of B of 90 degrees around I . This means that the order of A and B is important, it controls the position of M . The command creates the two points M and I . The construction is controlled by the following parameters:

- `CodeFig`, `CodeFigColor` et `SegmentSymbol` for marking the right angle ;
- `PointSymbol` et `PointName` for controlling the drawing of the two points, each of them can be specified separately with the parameters `...A` et `...B` ;
- parameters controlling the line drawing.



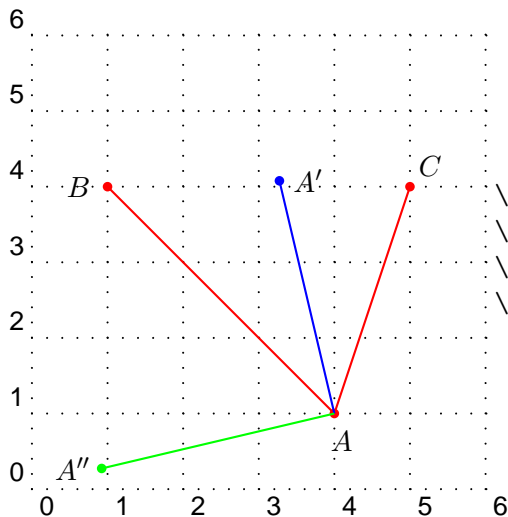
```
\pstTriangle[PointSymbol=none](3.5,1){A}(1,4){B}(5,4.2){C}
\psset{linecolor=red, CodeFigColor=red, nodesep=-1}
\pstMediatorAB[PointSymbolA=none]{A}{B}{I}{M_I}
\psset{PointSymbol=none, PointNameB=none}
\pstMediatorAB[CodeFig=true]{A}{C}{J}{M_J}
\pstMediatorAB[PosAngleA=45, linecolor=blue]{C}{B}{K}{M_K}
```

1.4.5 Bisectors of angles

```
\pstBissectBAC[⟨par⟩]{⟨B⟩}{⟨A⟩}{⟨C⟩}{⟨N⟩}
```

```
\pstOutBissectBAC[⟨par⟩]{⟨B⟩}{⟨A⟩}{⟨C⟩}{⟨N⟩}
```

there are two bisectors for a given geometric angle: the inside one and the outside one; this is why there is two commands. The angle is specified by three points specified in the trigonometric direction (anti-clockwise). The result of the commands is the specific line and a point belonging to this line. This point is built by a rotation of point B .



```
\psset{CurveType=polyline, linecolor=red}
\pstGeonode[PosAngle={180, -75, 45}](1,4){B}(4,1){A}(5,4){C}
\pstBissectBAC[linecolor=blue]{C}{A}{B}{A'}
\pstOutBissectBAC[linecolor=green, PosAngle=180]{C}{A}{B}{A''}
```

1.5 Intersections

Points can be defined by intersections. Six intersection types are managed:

- line-line;
- line-circle;
- circle-circle;
- function-function;
- function-line;
- function-circle.

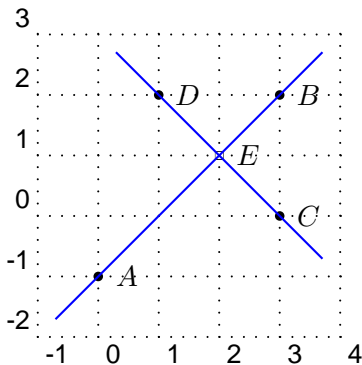
An intersection can not exist: case of parallel lines. In such a case, the point(s) are positioned at the origin. In fact, the user has to manage the existence of these points.

1.5.1 Line-Line

`\pstInterLL[$\langle par \rangle$]{ $\langle A \rangle$ }{ $\langle B \rangle$ }{ $\langle C \rangle$ }{ $\langle D \rangle$ }{ $\langle M \rangle$ }`

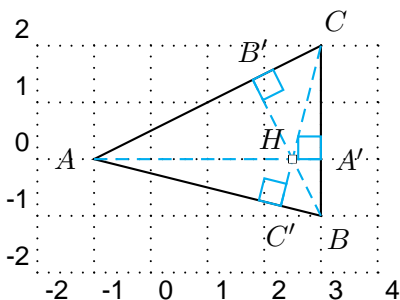
Draw the intersection point between lines (AB) and (CD) .

basique



```
\pstGeonode(0,-1){A}(3,2){B}(3,0){C}(1,2){D}
\pstInterLL[PointSymbol=square]{A}{B}{C}{D}{E}
\psset{linecolor=blue, nodesep=-1}
\pstLineAB{A}{B}\pstLineAB{C}{D}
```

Horthocentre



```
\psset{CodeFig=true, PointSymbol=none}
\pstTriangle[PosAngleA=180](-1,0){A}(3,-1){B}(3,2){C}
\pstProjection[PosAngle=-90]{B}{A}{C}
\pstProjection{B}{C}{A}
\pstProjection[PosAngle=90]{A}{C}{B}
\pstInterLL[PosAngle=135, PointSymbol=square]
{A}{A'}{B}{B'}{H}
```

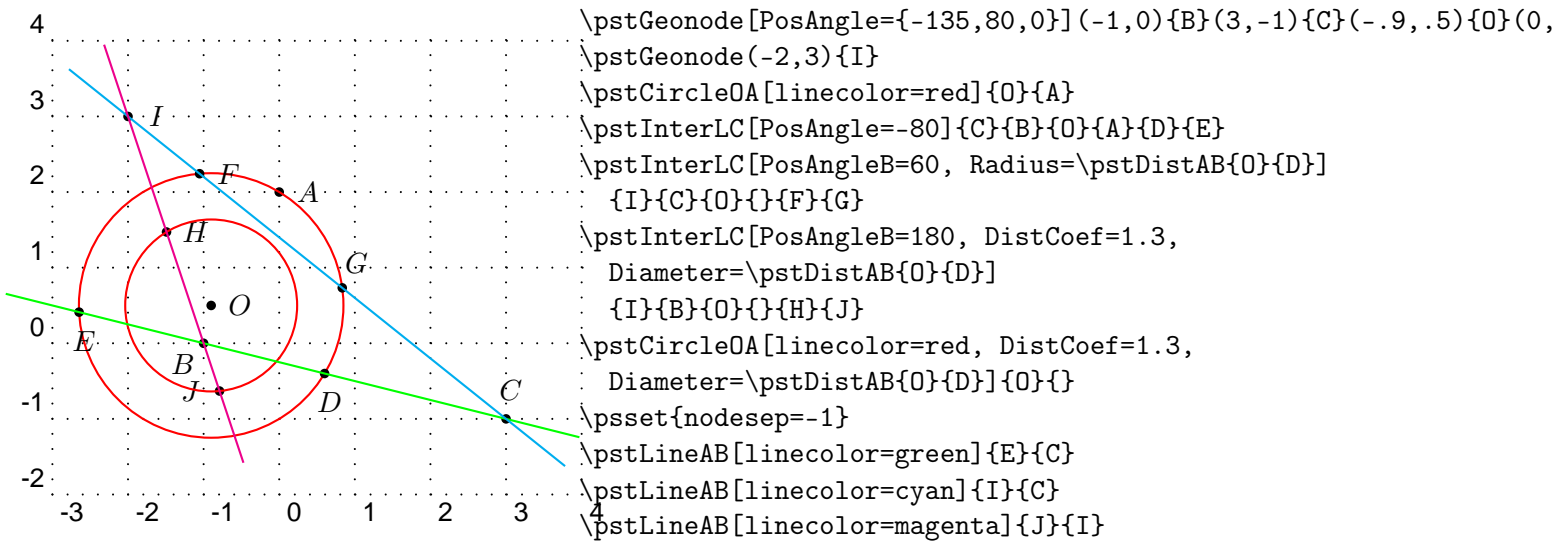
1.5.2 Circle–Line

`\pstInterLC[⟨par⟩]{⟨A⟩}{⟨B⟩}{⟨O⟩}{⟨C⟩}{⟨M1⟩}{⟨M2⟩}`

Draw the one or two intersection point(s) between the line (AB) and the circle of centre O and with radius OC .

The circle is specified with its center and either a point of its circumference or with a radius specified with parameter **radius** or its diameter specified with parameter **Diameter**. These two parameters can be modify by coefficient **DistCoef**.

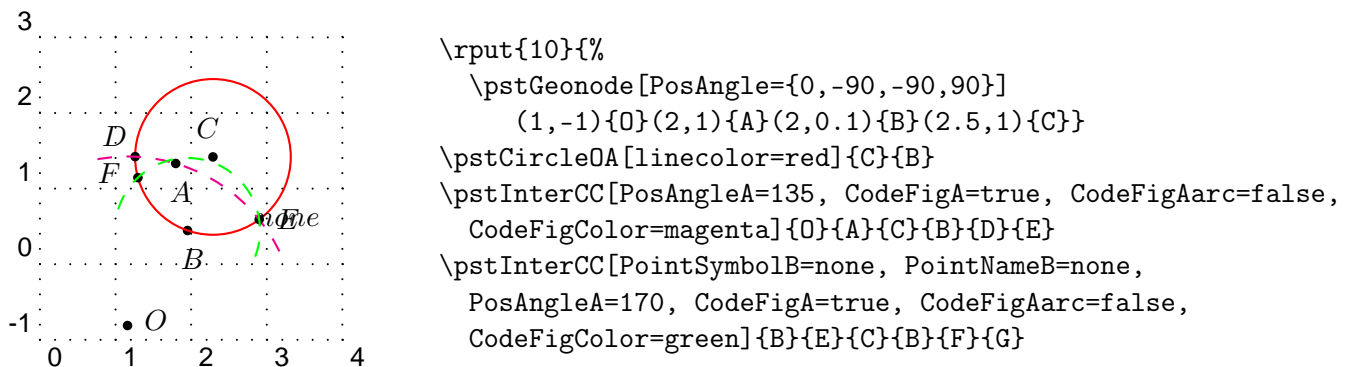
The position of the two points is such that the vectors \overrightarrow{AB} and $\overrightarrow{M_1M_2}$ are in the same direction. Thus, if the points defining the line are switch, then the resulting points will be also switched. If the intersection is void, then the points are positionned at the center of the circle.



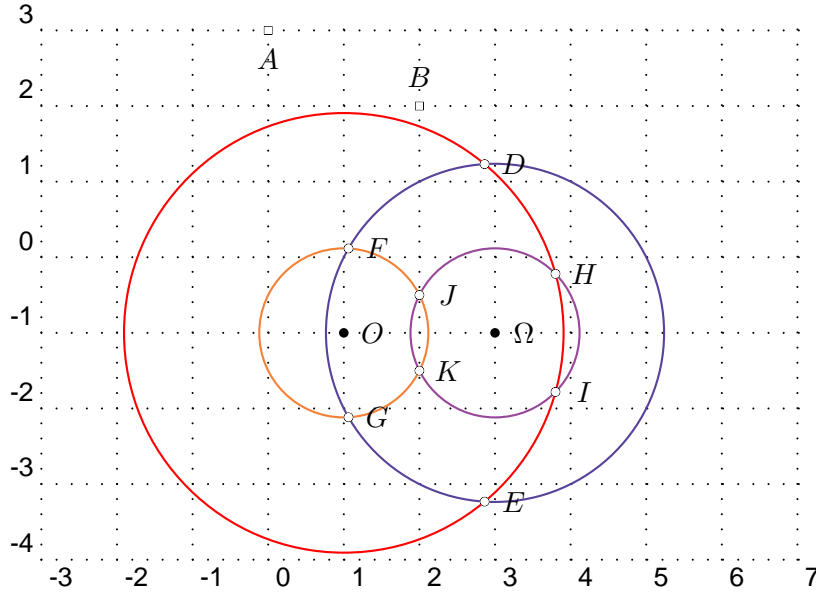
1.5.3 Circle–Circle

`\pstInterCC[⟨par⟩]{⟨O1⟩}{⟨B⟩}{⟨O2⟩}{⟨C⟩}{⟨M1⟩}{⟨M2⟩}`

This function is similar to the last one. The boolean parameters **CodeFigA** et **CodeFigB** allow the drawing of the arcs at the intersection. In order to get a coherence **CodeFig** allow the drawing of both arcs. The boolean parameters **CodeFigArcA** and **CodeFigArcB** specified the direction of these optional arcs: trigonometric (by default) or clockwise. Here is a first example.



And a more complete one, which includes the special circle specification using radius and diameter. For such specifications it exists the parameters `RadiusA`, `RadiusB`, `DiameterA` and `DiameterB`.



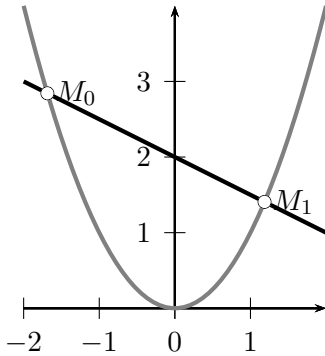
```
\pstGeonode[PointName={\Omega,0}](3,-1){\Omega}(1,-1){O}
\pstGeonode[PointSymbol=square, PosAngle={-90,90}](0,3){A}(2,2){B}
\psset{PointSymbol=o}
\pstCircleOA[linecolor=red, DistCoef=1 3 10 div add, Radius=\pstDistAB{A}{B}]{O}{\Omega}
\pstCircleOA[linecolor=Orange, Diameter=\pstDistAB{A}{B}]{O}{\Omega}
\pstCircleOA[linecolor=Violet, Radius=\pstDistAB{A}{B}]{\Omega}{\Omega}
\pstCircleOA[linecolor=Purple, Diameter=\pstDistAB{A}{B}]{\Omega}{\Omega}
\pstInterCC[DistCoef=1 3 10 div add, RadiusA=\pstDistAB{A}{B},
  DistCoef=none, RadiusB=\pstDistAB{A}{B}]{O}{\Omega}{D}{E}
\pstInterCC[DiameterA=\pstDistAB{A}{B}, RadiusB=\pstDistAB{A}{B}]{O}{\Omega}{F}{G}
\pstInterCC[DistCoef=1 3 10 div add, RadiusA=\pstDistAB{A}{B},
  DistCoef=none, DiameterB=\pstDistAB{A}{B}]{O}{\Omega}{H}{I}
\pstInterCC[DiameterA=\pstDistAB{A}{B}, DiameterB=\pstDistAB{A}{B}]{O}{\Omega}{J}{K}
```

1.5.4 Function–function

`\pstInterFF[$\langle par \rangle$]{ $\langle f \rangle$ }{ $\langle g \rangle$ }{ $\langle x_0 \rangle$ }{ $\langle M \rangle$ }`

This function put a point at the intersection between two curves defined by a function. x_0 is an intersection approximated value of the abscissa. It is obviously possible to use this function several times if more than one intersection is present. Each function is described in `PostScript` in the same way as the description used by the `\psplot` macro of `PSTricks`. A constant function can be specified, and then searching function root is possible.

The Newton algorithm is used for the research, and the intersection may not be found. In such a case the point is positioned at the origin. On the other hand, the research can be trapped (in a local extremum near zero).

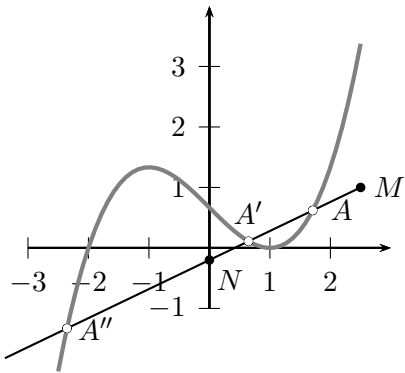


```
\psaxes{->}(0,0)(-2,0)(2,4)
\psset{linewidth=2\pslinewidth}
\psplot[linecolor=gray]{-2}{2}{x 2 exp}
\psplot{-2}{2}{2 x 2 div sub}
\psset{PointSymbol=o}
\pstInterFF{2 x 2 div sub}{x 2 exp}{1}{M_1}
\pstInterFF{2 x 2 div sub}{x 2 exp}{-2}{M_0}
```

1.5.5 Function–line

`\pstInterFL[⟨par⟩]{⟨f⟩}{⟨A⟩}{⟨B⟩}{⟨x0⟩}{⟨M⟩}`

Puts a point at the intersection between the function f and the line (AB) .

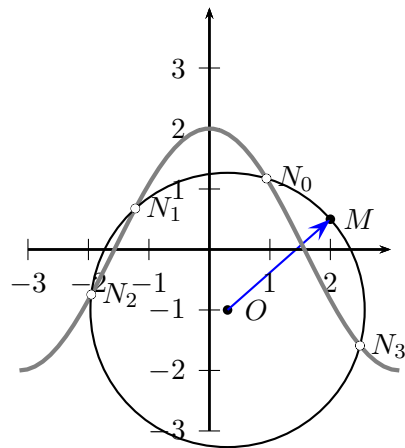


```
\def\F{x 3 exp 3 div x sub 2 3 div add .0001 add}
\psaxes{->}(0,0)(-3,-1)(3,4)
\psplot[linecolor=gray]{-2.5}{2.5}{\F}
\psset{PointSymbol=*}
\pstGeonode[PosAngle=-45,0](0,-.2){N}(2.5,1){M}
\pstLineAB[nodesepA=-3cm]{N}{M}
\psset{PointSymbol=o}
\pstInterFL{\F}{N}{M}{2}{A}
\pstInterFL[PosAngle=90]{\F}{N}{M}{0}{A'}
\pstInterFL{\F}{N}{M}{-2}{A''}
```

1.5.6 Function–circle

`\pstInterFC[⟨par⟩]{⟨f⟩}{⟨O⟩}{⟨A⟩}{⟨x0⟩}{⟨M⟩}`

Puts a point at the intersection between the function f and the circle of centre O and radius OA .



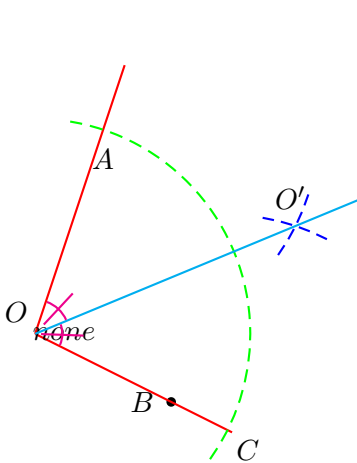
```
\def\F{x 180 mul 3.1415926 div cos 2 mul}
\pstGeonode(0.3,-1){O}(2,.5){M}
\ncline[linecolor=blue, arrowscale=2]{->}{O}{M}
\psaxes{->}(0,0)(-3,-3)(3,4)
\psplot[linecolor=gray]{-3.14}{3.14}{\F}
\psset{PointSymbol=*}
\pstCircleOA{O}{M}
\psset{PointSymbol=o}
\pstInterFC{\F}{O}{M}{1}{N_0}
\pstInterFC{\F}{O}{M}{-1}{N_1}
\pstInterFC{\F}{O}{M}{-2}{N_2}
\pstInterFC{\F}{O}{M}{2}{N_3}
```

Chapter 2

Examples gallery

2.1 Basic geometry

2.1.1 Drawing of the bissector



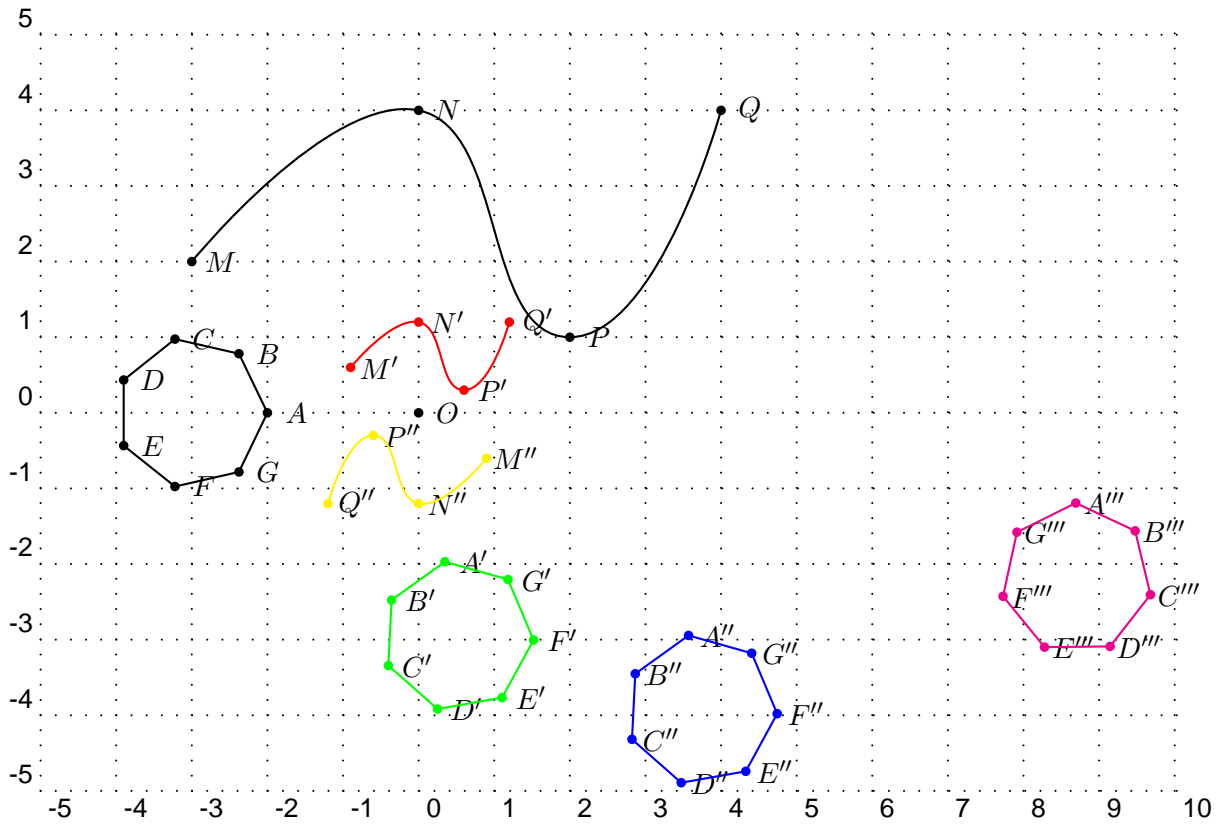
```

\psset{PointSymbol=none}
\pstGeonode[PosAngle={180,130,-90}, PointSymbol={default,none}]
    (2,0){B}(0,1){O}(1,4){A}
\pstLineAB[nodesepB=-1, linecolor=red]{O}{A}
\pstLineAB[nodesepB=-1, linecolor=red]{O}{B}
\pstInterLC[PointSymbolA=none, PosAngleB=-45]{O}{B}{O}{A}{G}{C}
\psset{arcsepA=-1, arcsepB=-1}
\pstArcOAB[linecolor=green, linestyle=dashed]{O}{C}{A}
\pstInterCC[PointSymbolB=none, PointNameB=none, PosAngleA=100]{A}{O}{C}{O'}
\pstArcOAB[linecolor=blue, linestyle=dashed]{A}{O'}{O'}
\pstArcOAB[linecolor=blue, linestyle=dashed]{C}{O'}{O'}
\pstLineAB[nodesepB=-1, linecolor=cyan]{O}{O'}
\psset{arcsep=\pslinewidth, linecolor=magenta, Mark=MarkHash}
\pstMarkAngle{C}{O}{O'}{}
\pstMarkAngle[MarkAngleRadius=.5]{O'}{O}{A}{}

```

2.1.2 Transformation de polygones et courbes

Here is an example of the use of CurveType with transformation.

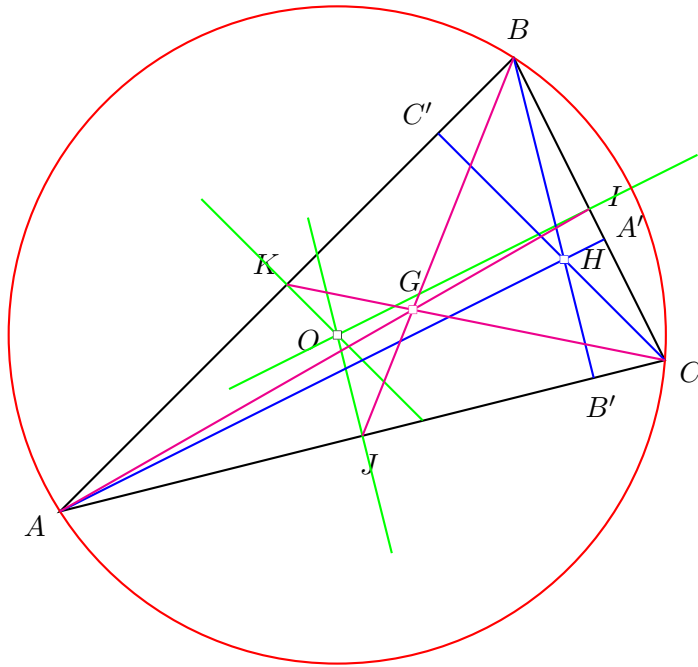


```

\pstGeonode{O}
\rput(-3,0){\pstGeonode[CurveType=polygon](1,0){A}(1;51.43){B}(1;102.86){C}
(1;154.29){D}(1;205.71){E}(1;257.14){F}(1;308.57){G}}
\rput(-4,-1){\pstGeonode[CurveType=curve](1,3){M}(4,5){N}(6,2){P}(8,5){Q}}
\pstRotation[linecolor=green, RotAngle=100, CurveType=polygon]{O}{A, B, C, D, E, F, G}
\pstHomO[linecolor=red, HomCoef=.3, CurveType=curve]{O}{M,N,P,Q}
\pstTranslation[linecolor=blue, CurveType=polygon]{C}{O}{A', B', C', D', E', F', G'}
\pstSymO[linecolor=yellow, CurveType=curve]{O}{M',N',P',Q'}
\pstOrtSym[linecolor=magenta, CurveType=polygon]{Q}{F''}
{A', B', C', D', E', F', G'}[A''', B''', C''', D''', E''', F''', G''']

```

2.1.3 Triangle lines



```

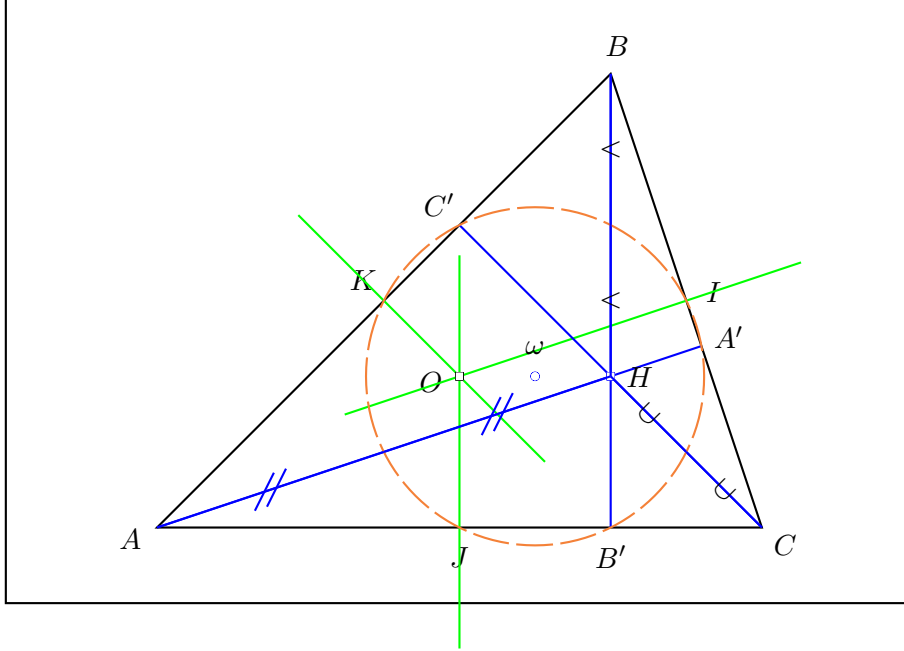
\psset{PointSymbol=none}
\pstTriangle[PointSymbol=none](-2,-1){A}(1,2){B}(2,0){C}
% les médiatrices
{% encapsulation de modif paramètres
  \psset{linestyle=none, PointNameB=none}
  \pstMediatorAB{A}{B}{K}{KP}
  \pstMediatorAB[PosAngleA=-40]{C}{A}{J}{JP}
  \pstMediatorAB[PosAngleA=75]{B}{C}{I}{IP}
}% fin
\pstInterLL[PointSymbol=square, PosAngle=-170]{I}{IP}{J}{JP}{O}
{% encapsulation de modif paramètres
  \psset{nodesep=-.8, linecolor=green}
  \pstLineAB{O}{I}\pstLineAB{O}{J}\pstLineAB{O}{K}
}% fin
\pstCircleOA[linecolor=red]{O}{A}
% pour que le symbol de O soit sur et non sous les droites
\psdot[dotstyle=square](O)
% les hauteurs et l'orthocentre
\pstProjection{B}{A}{C}
\pstProjection{B}{C}{A}
\pstProjection{A}{C}{B}
\psset{linecolor=blue}\ncline{A}{A'}\ncline{C}{C'}\ncline{B}{B'}
\pstInterLL[PointSymbol=square]{A}{A'}{B}{B'}{H}
% les médianes et le centre de gravité
\psset{linecolor=magenta}\ncline{A}{I}\ncline{C}{K}\ncline{B}{J}

```



```
\pstCGravABC[PointSymbol=square, PosAngle=95]{A}{B}{C}{G}
```

2.1.4 Euler circle

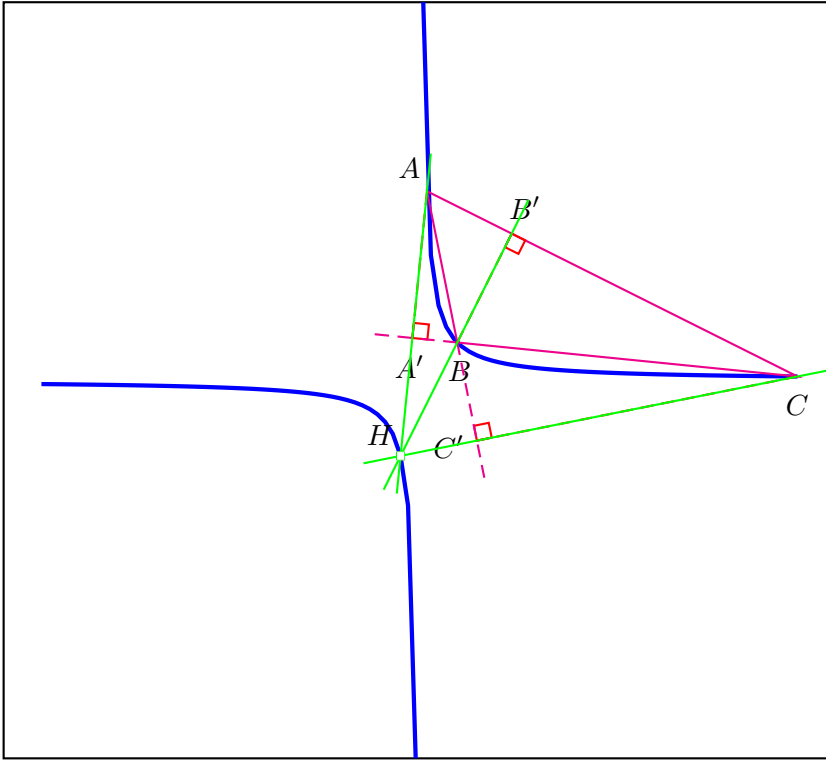


```
\psset{PointSymbol=none}
\pstTriangle(-2,-1){A}(1,2){B}(2,-1){C}
{% encapsulation de modif paramètres
  \psset{linestyle=none, PointSymbolB=none, PointNameB=none}
  \pstMediatorAB{A}{B}{K}{KP}
  \pstMediatorAB{C}{A}{J}{JP}
  \pstMediatorAB{B}{C}{I}{IP}
}% fin
\pstInterLL[PointSymbol=square, PosAngle=-170]{I}{IP}{J}{JP}{O}
{% encapsulation de modif paramètres
  \psset{nodesep=-.8, linecolor=green}
  \pstLineAB{O}{I}\pstLineAB{O}{J}\pstLineAB{O}{K}
}% fin
\psdot[dotstyle=square](O)
\pstProjection{B}{A}{C}
\pstProjection{B}{C}{A}
\pstProjection{A}{C}{B}
\psset{linecolor=blue}\ncline{A}{A'}\ncline{C}{C'}\ncline{B}{B'}
\pstInterLL[PointSymbol=square]{A}{A'}{B}{B'}{H}
% le cercle d'Euler (centre au milieu de [OH])
\pstMiddleAB[PointSymbol=o, PointName=\omega]{O}{H}{\omega}
\pstCircleOA[linecolor=Orange, linestyle=dashed, dash=5mm 1mm]{\omega}{B'}
\psset{PointName=none}
% il passe par le milieu des segments joignant l'orthocentre et les sommets
\pstMiddleAB{H}{A}{AH}\pstMiddleAB{H}{B}{BH}\pstMiddleAB{H}{C}{CH}
\pstSegmentMark{H}{AH}\pstSegmentMark{AH}{A}
\psset{SegmentSymbol=wedge}\pstSegmentMark{H}{BH}\pstSegmentMark{BH}{B}
```

```
\psset{SegmentSymbol=cup}\pstSegmentMark{H}{CH}\pstSegmentMark{CH}{C}
```

2.1.5 Orthocenter and hyperbola

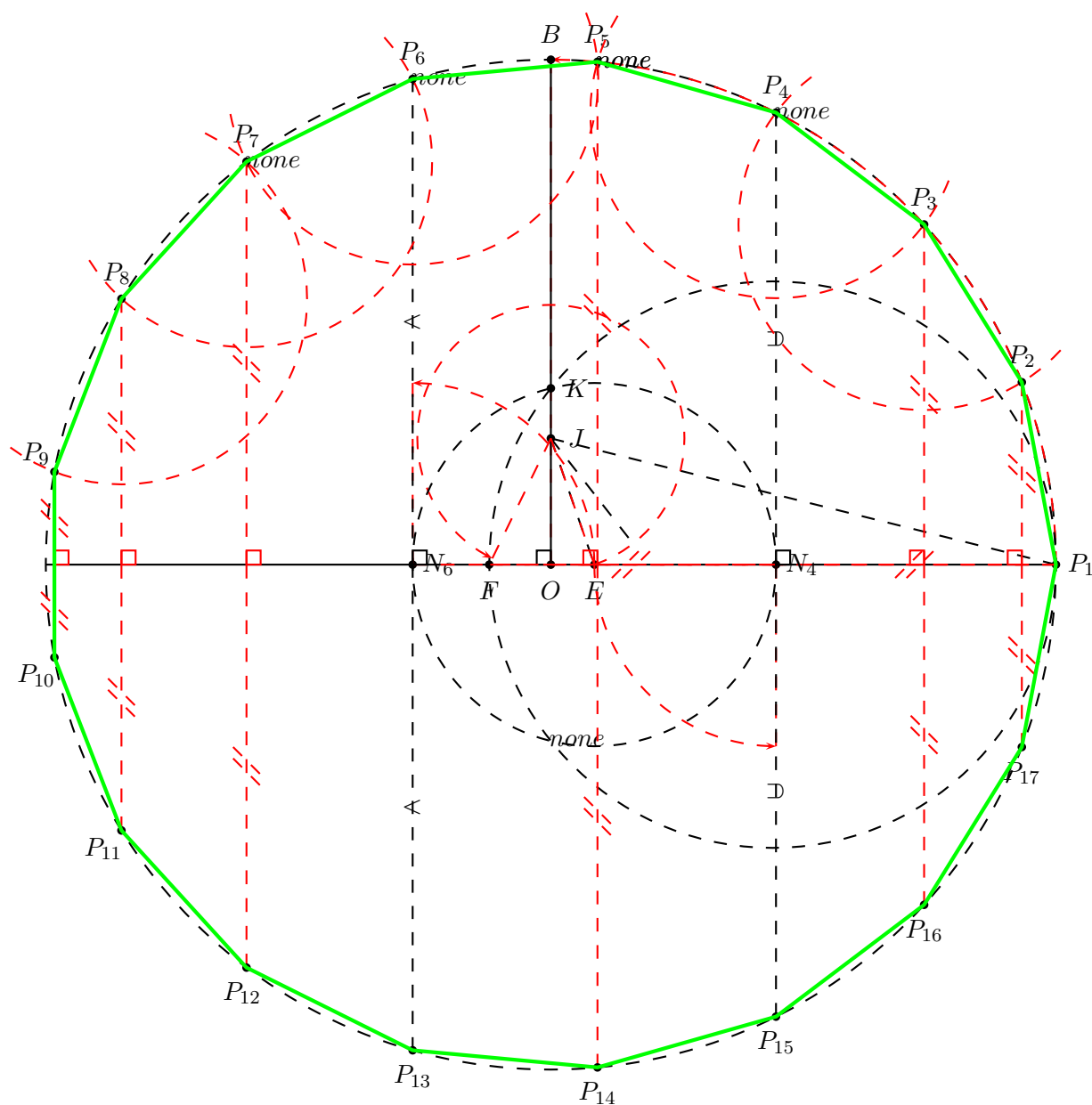
The orthocenter of a triangle whose points are on the branches of the hyperbola $\mathcal{H} : y = a/x$ belong to this hyperbola.



```
\psset{linecolor=blue, linewidth=2\pslinewidth}
\psplot{-10}{-.1}{1 x div}
\psplot{.1}{10}{1 x div}
\psset{PointSymbol=none, linewidth=.5\pslinewidth}
\pstTriangle[linecolor=magenta, PosAngleB=-85, PosAngleC=-90](.2,5){A}(1,1){B}(10,.1){C}
%\pstTriangle[linecolor=magenta, PosAngleB=-135](.2, 5){A}(-1,-1){B}(10,.1){C}
\psset{linecolor=magenta, CodeFig=true, CodeFigColor=red}
\pstProjection{B}{A}{C}
\ncline[nodesepA=-1, linestyle=dashed, linecolor=magenta]{C'}{B}
\pstProjection{B}{C}{A}
\ncline[nodesepA=-1, linestyle=dashed, linecolor=magenta]{A'}{B}
\pstProjection{A}{C}{B}
\pstInterLL[PosAngle=135, PointSymbol=square]{A}{A'}{B}{B'}{H}
\psset{linecolor=green, nodesep=-1}
\pstLineAB{A}{H}\pstLineAB{B'}{H}\pstLineAB{C}{H}
\psdot[dotstyle=square](H)
```

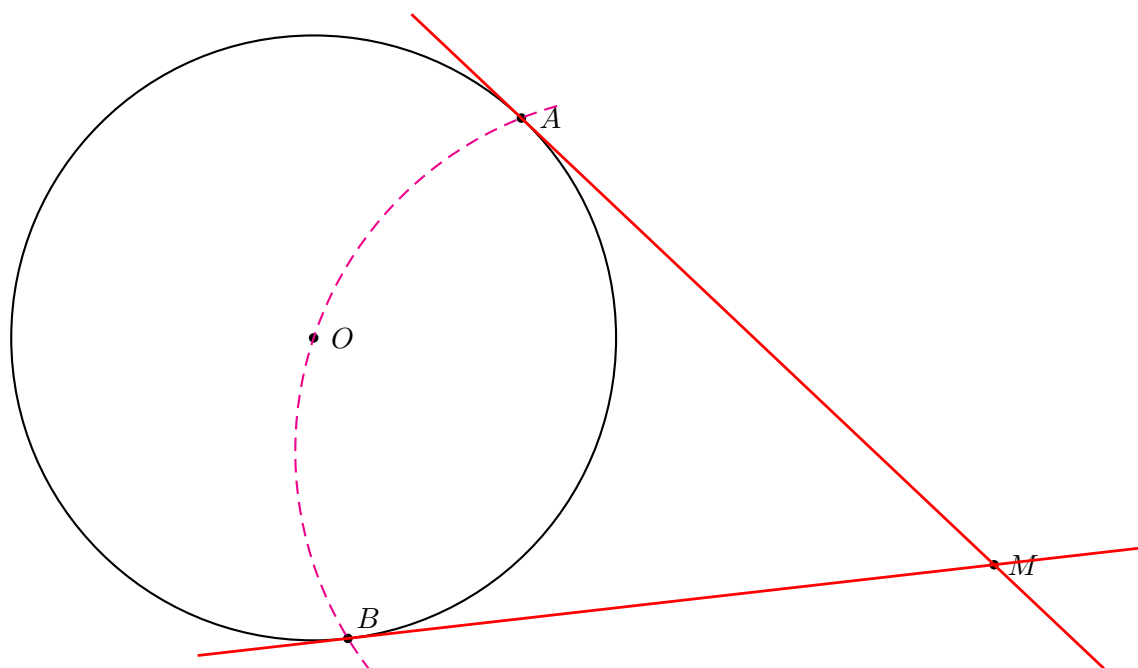
2.1.6 17 sides regular polygon

Striking picture created by K. F. Gauss. he also proved that it is possible to build the regular polygons which have $2^{2^p} + 1$ sides, the following one has 257 sides!

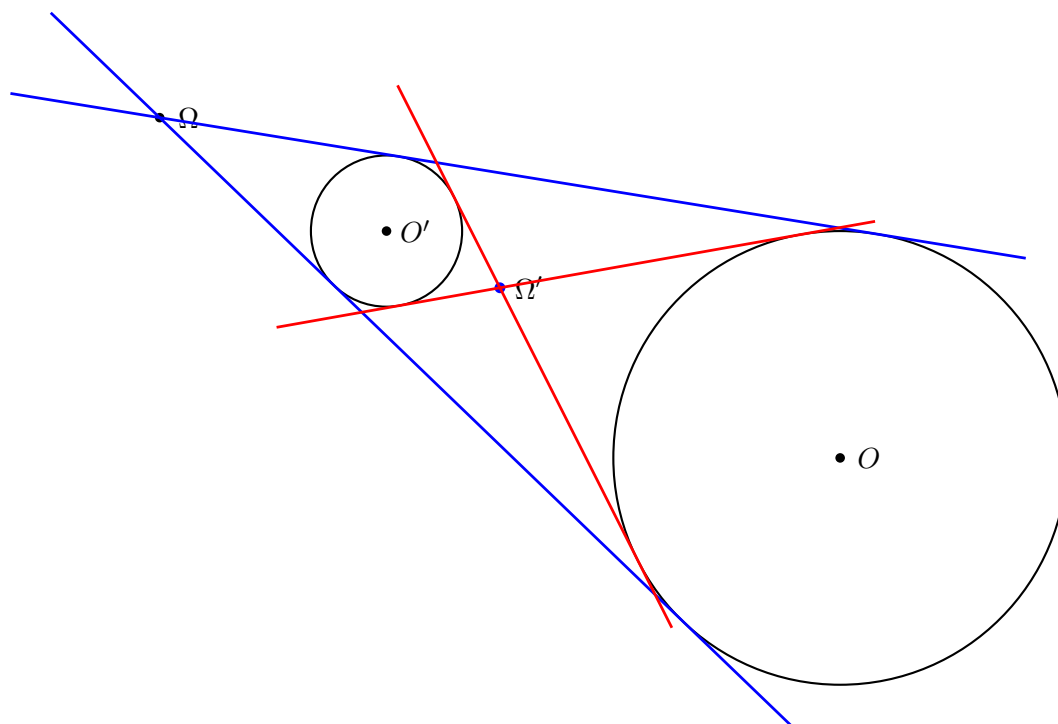


2.1.7 Circles & tangents

The drawing of the circle tangents which crosses a given point.

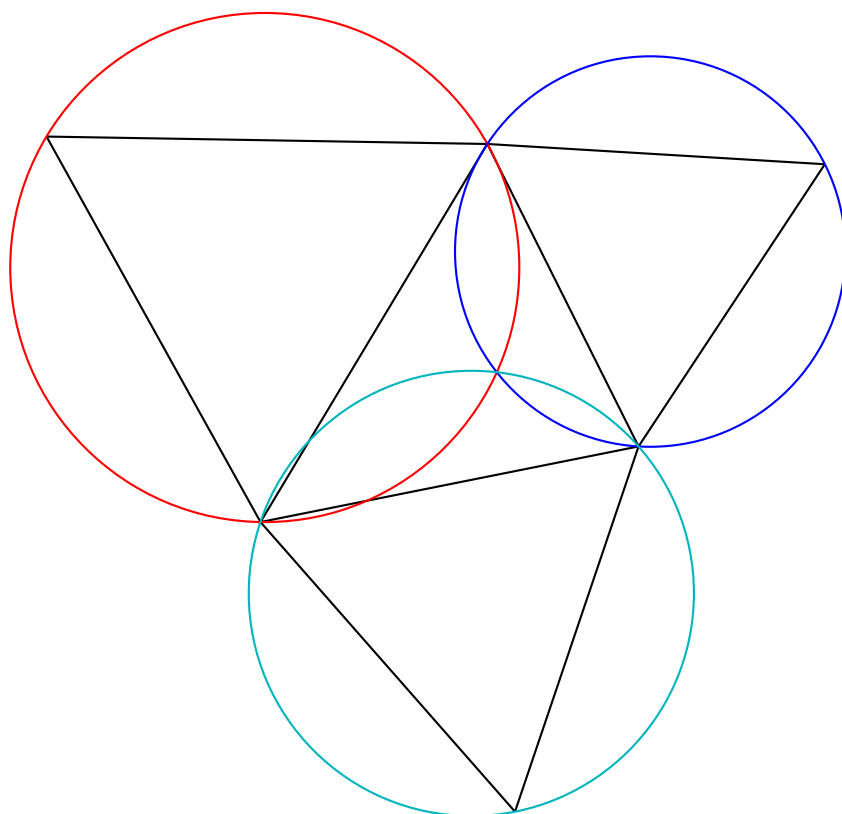


The drawing of the common tangent of two circles.

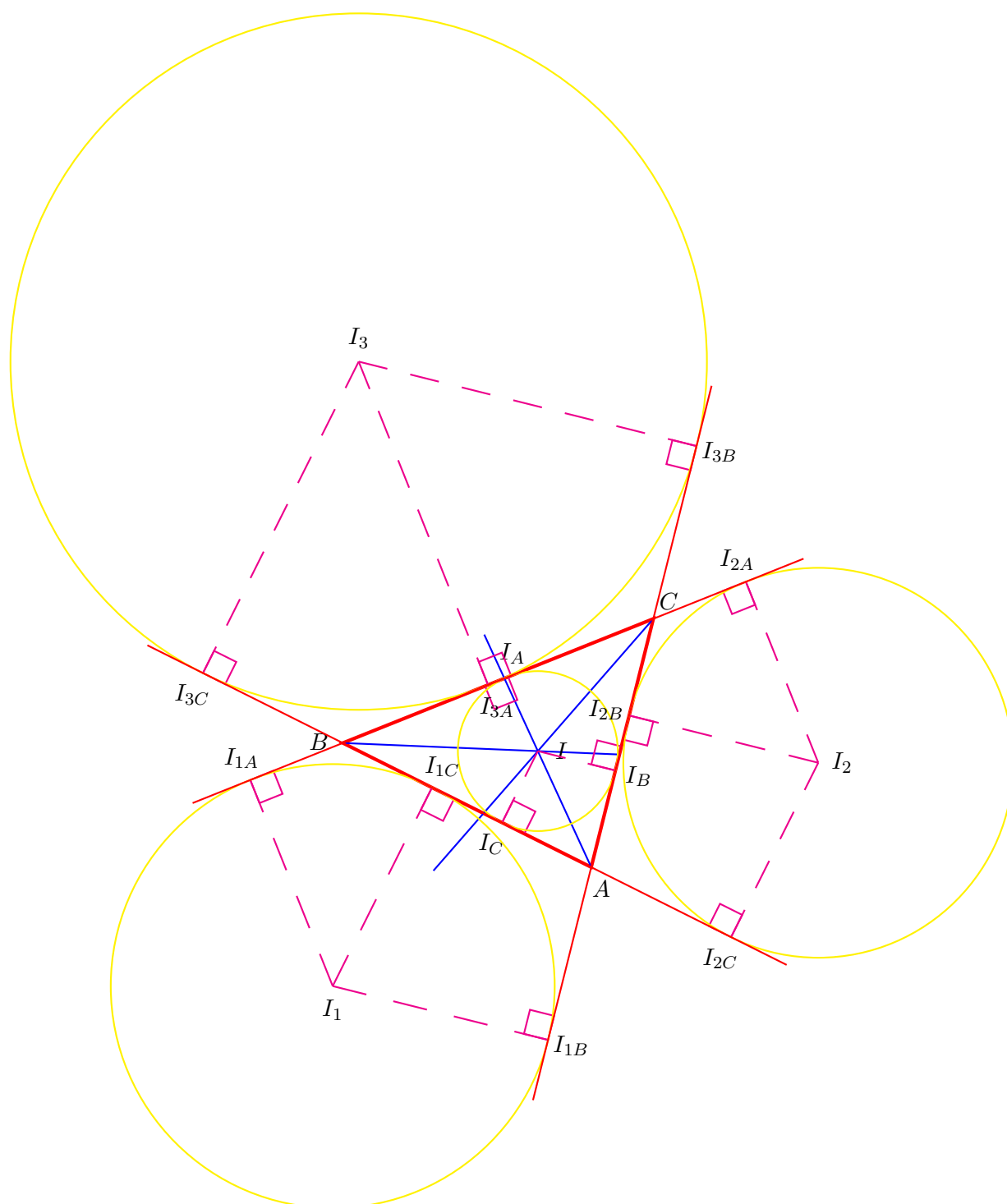


2.1.8 Fermat's point

Drawing of Manuel Luque.



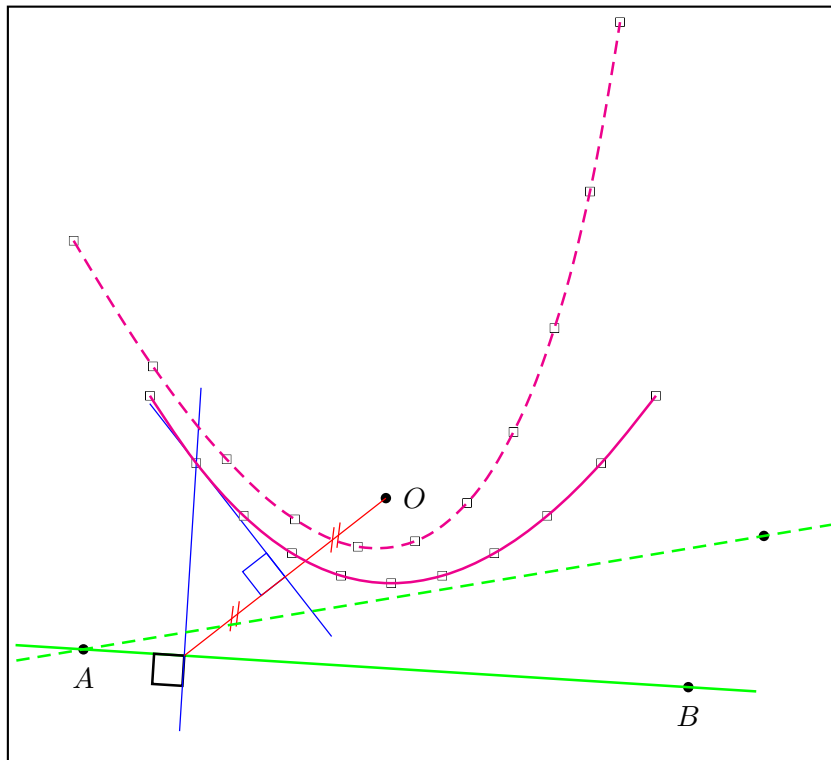
2.1.9 Escribed and inscribed circles of a triangle



2.2 Some locus points

2.2.1 Parabola

The parabola is the set of points which are at the same distance between a point and a line.

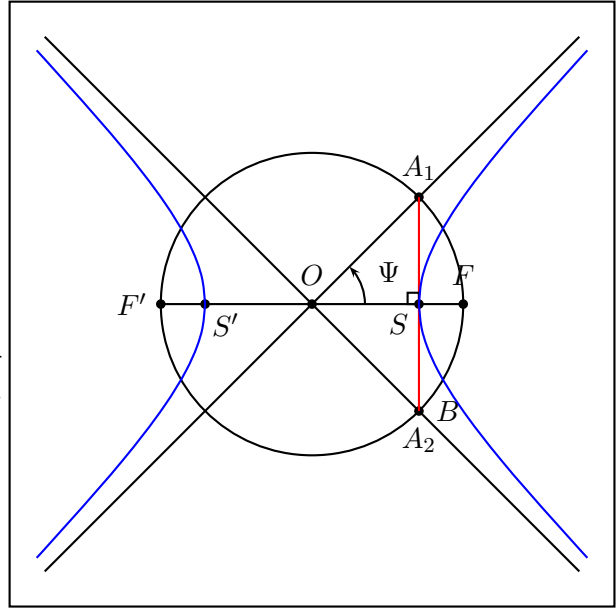


```
\psset{linewidth=1.2\pslinewidth}\renewcommand{\NbPt}{11}
\pstGeonode[PosAngle={0,-90}](5,4){O}(1,2){A}(9,1.5){B}
\newcommand{\Parabole}[1][100]{%
  \pstLineAB[nodesep=-.9, linecolor=green]{A}{B}
  \psset{RotAngle=90, PointSymbol=none, PointName=none}
  \multido{\n=1+1}{\NbPt}{%
    \pstHomO[HomCoef=\n\space \NbPt\space 1 add div]{A}{B}[M\n]
    \pstMediatorAB[linestyle=none]{M\n}{O}{M\n_I}{M\n_IP}
    \pstRotation{M\n}{A}[M\n_P]
    \pstInterLL[PointSymbol=square, PointName=none]{M\n_I}{M\n_IP}{M\n}{M\n_P}{P\n}
    \ifnum\n=#1
      \bgroup
      \pstRightAngle{A}{M\n}{M\n_P}
      \psset{linewidth=.5\pslinewidth, nodesep=-1, linecolor=blue}
      \pstLineAB{M\n_I}{P\n}\pstLineAB{M\n}{P\n}
      \pstRightAngle{P\n}{M\n_I}{M\n}
      \psset{linecolor=red}\pstSegmentMark{M\n}{M\n_I}\pstSegmentMark{M\n_I}{O}
      \egroup
    \fi}}%fin multido-newcommand
\Parabole[2]\pstGenericCurve[linecolor=magenta]{P_}{1}{\NbPt}
%% Nouvelle parabole avec un nouveau point B
\pstGeonode[PointSymbol=*, PosAngle=-90](10,3.5){B}
\psset{linestyle=dashed}
\Parabole\pstGenericCurve[linecolor=magenta]{P_}{1}{\NbPt}
```

2.2.2 Hyperbola

The hyperbola is the set of points whose difference between their distance of two points (the focus) is constant.

```
%% QQ RAPPELS : a=\Sommet, c=\PosFoyer,
%% b^2=c^2-a^2, e=c/a
%% pour une hyperbole -> e>1, donc c>a,
%% ici on choisi a=\sqrt{2}, c=2, e=\sqrt{2}
%% M est sur H <=> |MF-MF'|=2a
```



```
\newcommand{\Sommet}{1.4142135623730951}
\newcommand{\PosFoyer}{2}
\newcommand{\HypAngle}{0}
\setcounter{i}{1}
\newcounter{CoefDiv}\setcounter{CoefDiv}{20}
\newcounter{Inc}\setcounter{Inc}{2}
\newcounter{n}\setcounter{n}{2}
%% rayon des cercles successifs utilisés pour trouver les points de H
%% on choisit \Rii=\Ri+2\Sommet (définition de l'hyperbole)
\newcommand{\Ri}{% c'est du postscript
  \PosFoyer\space\Sommet\space sub \arabic{i}\space\arabic{CoefDiv}\space div add}
\newcommand{\Rii}{\Ri\space\Sommet\space 2 mul add}
\pstGeonode[PosAngle=90]{0}(\PosFoyer;\HypAngle){F}
\pstSym0[PosAngle=180]{0}{F}\pstLineAB{F}{F'}
%% TRACÉ DES ASYMPTOTES
\pstCircleOA{0}{F}
%% positionnement des deux sommets de H
\pstGeonode[PosAngle=-135](\Sommet;\HypAngle){S}
\pstGeonode[PosAngle=-45](\Sommet;\HypAngle){S'}
%% l'intersection de la droite perpendiculaire à (FF') passant par S,
%% coupe les asymptotes sur le cercle de diamètre [FF'] (cette droite est une tangente)
\pstRotation[RotAngle=90, PointSymbol=none]{S}{0}[B]
\pstInterLC[PosAngleA=90, PosAngleB=-90]{S}{B}{0}{F}{A_1}{A_2}
\pstLineAB[nodesepA=-3,nodesepB=-5]{A_1}{0}
\pstLineAB[nodesepA=-3,nodesepB=-5]{A_2}{0}
%% cos(\Psi)=OS/OF (c-a-d \Sommet/\PosFoyer)
%% ici \sqrt{2}/2, donc \Psi=45 => hyperbole équilatère
\pstMarkAngle[LabelSep=.8, MarkAngleRadius=.7, arrows=->,
  LabelSep=1.1]{F}{0}{A_1}{\Psi}
```



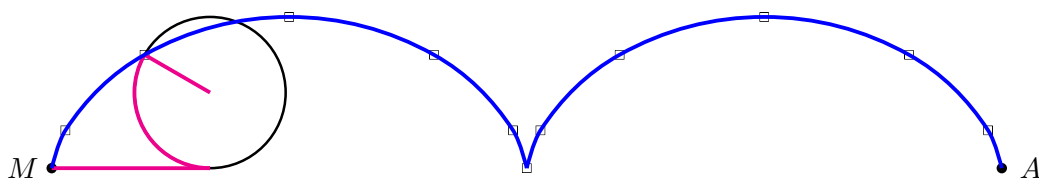
```

\ncircle[linewidth=red]{A_1}{A_2}
\pstRightAngle[RightAngleSize=.15]{A_1}{S}{O}
\psset{PointName=none}
\whiledo{\value{n}<8}{%
  \psset{RadiusA=\pstDistVal{\Ri},RadiusB=\pstDistVal{\Rii},PointSymbol=none}
  \pstInterCC{F'}{F'}{M\arabic{n}}{P\arabic{n}}
  \pstInterCC{F'}{F'}{M'\arabic{n}}{P'\arabic{n}}
  %% bcp de points au début, moins ensuite
  %% n -> numéro du point, i -> taille des cercles
  %% Inc -> incrément variable de i (2^n)
  \stepcounter{n}\addtocounter{i}{\value{Inc}}
  \addtocounter{Inc}{\value{Inc}}}% fin de whiledo
\psset{linewidth=blue}
%% tracé des quatres 1/2 branches de l'hyperbole
\pstGenericCurve[GenCurvFirst=S]{M}{2}{7}
\pstGenericCurve[GenCurvFirst=S]{P}{2}{7}
\pstGenericCurve[GenCurvFirst=S']{M'}{2}{7}
\pstGenericCurve[GenCurvFirst=S']{P'}{2}{7}
%% pour vérif le trace paramétrique
%\parametricplot[linecolor=black, linewidth=.25\pslinewidth]{-1}{1}
% {t dup tx@EclDdict begin sh exch ch end \Sommet\space mul exch
%   \PosFoyer\space dup mul \Sommet\space dup mul sub sqrt mul}
%\parametricplot[linecolor=black, linewidth=.25\pslinewidth]{-1}{1}
% {t dup tx@EclDdict begin sh exch ch end neg \Sommet\space mul exch
%   \PosFoyer\space dup mul \Sommet\space dup mul sub sqrt mul}

```

2.2.3 Cycloid

The wheel rolls from M to A . The circle points are on a cycloid.



```

\providecommand\NbPt{11}
\psset{linewidth=1.2\pslinewidth}
\pstGeonode[PointSymbol={*,none}, PointName={default,none}, PosAngle=180]{M}{0,1}{O}
%% 4*pi=12.5663706144
\pstGeonode(12.5663706144,0){A}
\pstTranslation[PointSymbol=none, PointName=none]{M}{A}{0}{B}
\multido{\nA=1+1}{\NbPt}{%
  \pstHomO[HomCoef=\nA\space \NbPt\space 1 add div,
    PointSymbol=none, PointName=none]{O}{B}{O\nA}
  \pstProjection[PointSymbol=none, PointName=none]{M}{A}{O\nA}{P\nA}
  %\pstCircleOA[linestyle=dashed, linecolor=red]{O\nA}{P\nA}

```

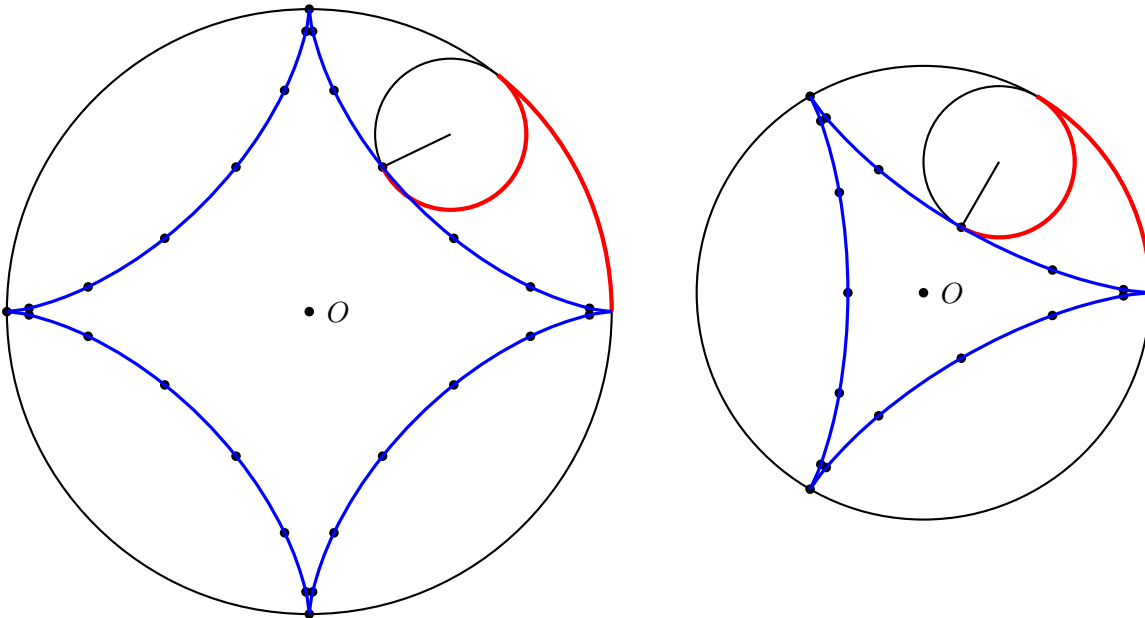
```

\pstCurvAbsNode[PointSymbol=square, PointName=none, CurvAbsNeg=true]%
  {0\nA}{P\nA}{M\nA}{\pstDistAB{0}{0\nA}}
\ifnum\nA=2 %affichage du second cercle
  \bgroup
  \pstCircleOA{0\nA}{M\nA}
  \psset{linecolor=magenta, linewidth=1.5\pslinewidth}
  \pstArcnOAB{0\nA}{P\nA}{M\nA}
  \ncline{0\nA}{M\nA}\ncline{P\nA}{M}
  \egroup
\fi
}% fin du multido
\psset{linecolor=blue, linewidth=1.5\pslinewidth}
\pstGenericCurve[GenCurvFirst=M]{M}{1}{6}
\pstGenericCurve[GenCurvLast=A]{M}{6}{\NbPt}
%% juste pour la vérification
%\parametricplot[linecolor=green, linewidth=.5\pslinewidth]{0}{12.5663706144}
% {t t 3.1415926 div 180 mul sin sub 1 t 3.1415926 div 180 mul cos sub}

```

2.2.4 Hypocycloids (Astroid and Deltoid)

A wheel rolls inside a circle, and depending of the radius ratio, it is an astroid, a deltoid and in the general case hypo-cycloids.



```

\newcommand{\HypoCyclo}[4][100]{%
  \def\R{#2}\def\petitR{#3}\def\NbPt{#4}% définitions locales pour lisibilité
  %% Selon la valeur de R sur petitR :
  %%      4 une astroïde -- 3 une deltoïde -- 2 un diamètre (!)
  \def\Anglen{\n\space 360 \NbPt\space 1 add div mul}
  \psset{PointSymbol=none,PointName=none}
  \pstGeonode[PointSymbol={*,none},PointName={default,none}, PosAngle=0]{0}{(\R;0)}{P}
  \pstCircleOA{0}{P}

```

```

\pstHomO[HomCoef=\petitR\space\R\space div]{P}{0}[M]
\multido{\n=1+1}{\NbPt}{%
  \pstRotation[RotAngle=\Anglen]{0}{M}[M\n]
  \rput(M\n){\pstGeonode(\petitR;0){Q}}
  \pstRotation[RotAngle=\Anglen]{M\n}{Q}[N]
  \pstRotation[RotAngle=\n\space -360 \NbPt\space 1 add div
mul \R\space\petitR\space div mul, PointSymbol=*, PointName=none]%
{M\n}{N}[N\n]
\ifnum\n=#1
  \pstCircleOA{M\n}{N\n}\ncircle{M\n}{N\n}%
  {\psset{linecolor=red, linewidth=2\pslinewidth}
  \pstArcOAB{M\n}{N\n}{N}\pstArcOAB{0}{P}{N}}
\fi}}%fin multido-newcommand

\HypoCyclo[4]{4}{1}{27}
\psset{linecolor=blue, linewidth=1.5\pslinewidth}
%% il est préférable de mettre les quatre branches séparément à cause
%% des points de rebroussement
\pstGenericCurve[GenCurvFirst=P]{N}{1}{7}
\pstGenericCurve{N}{7}{14}\pstGenericCurve{N}{14}{21}
\pstGenericCurve[GenCurvLast=P]{N}{21}{27}

```

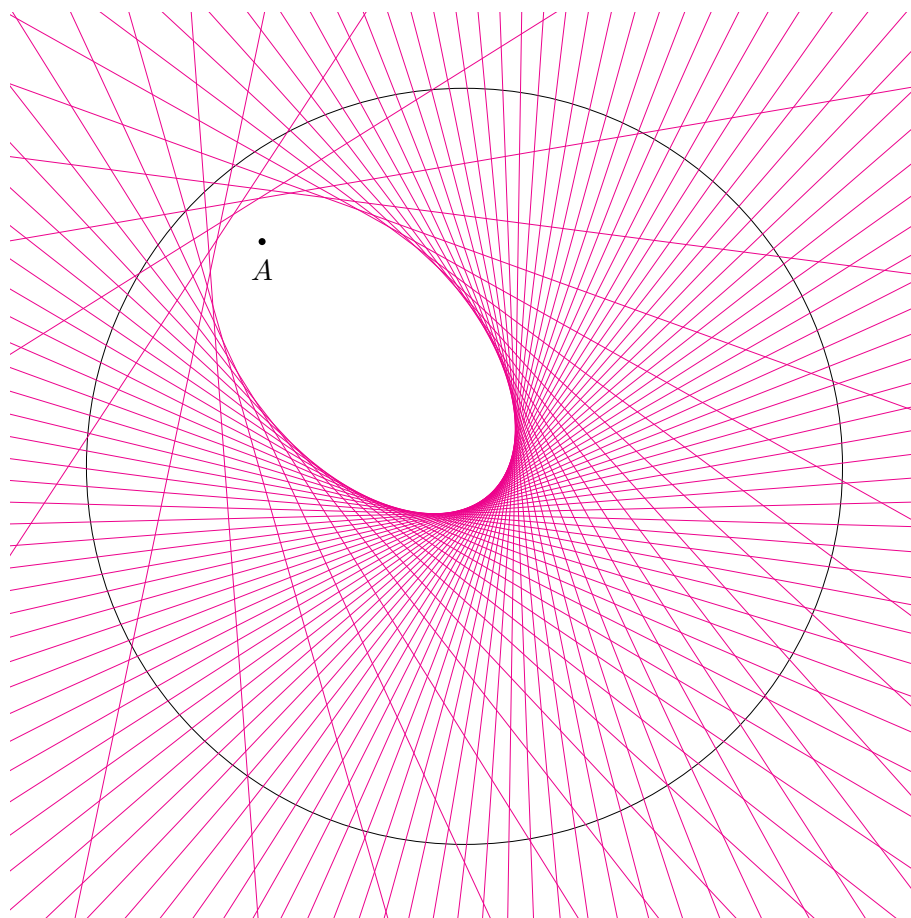
2.3 Lines and circles envelope

2.3.1 Conics

Let's consider a circle and a point A not on the circle. The set of all the mediator lines of segments defined by A and the circle points, create two conics depending of the position of A :

- inside the circle: an hyperbola;
- outside the circle: an ellipse.

(figure of O. Reboux).



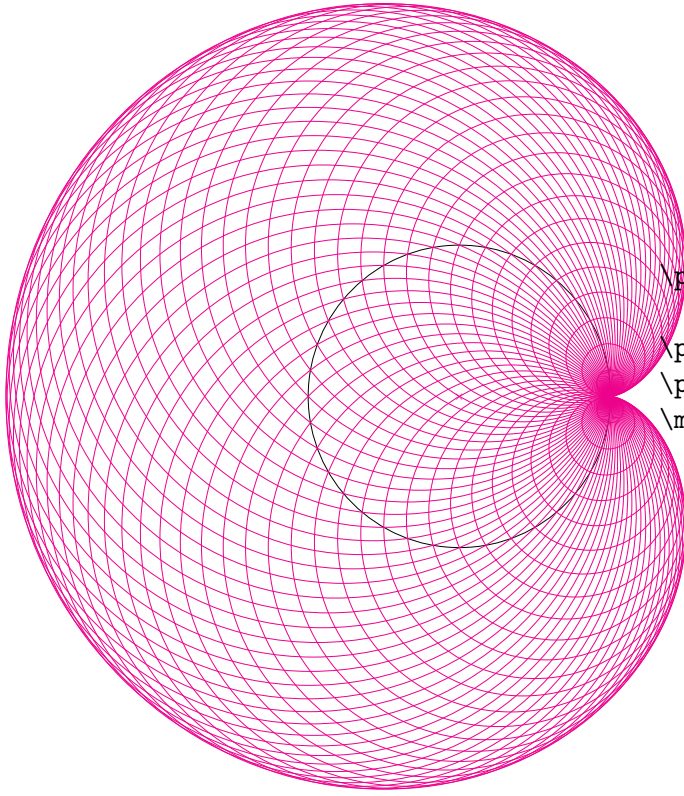
```

\psset{linewidth=0.4\pslinewidth,PointSymbol=none, PointName=none}
\pstGeonode[PosAngle=-90, PointSymbol={none,*,none}, PointName={none,default,none}]
  {0}(4;132){A}(5,0){O'}
\pstCircleOA{0}{O'}
\multido{\n=5+5}{72}{%
  \pstGeonode(5;\n){M_\n}
  \pstMediatorAB[nodesep=-15,linecolor=magenta]
    {A}{M_\n}{I}{J}}% fin multido

```

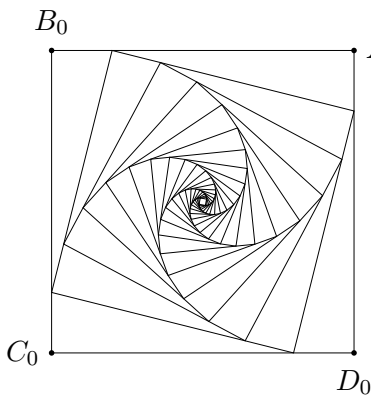
2.3.2 Cardioid

The cardioid is defined by the circles centered on a circle and crossing a given point.



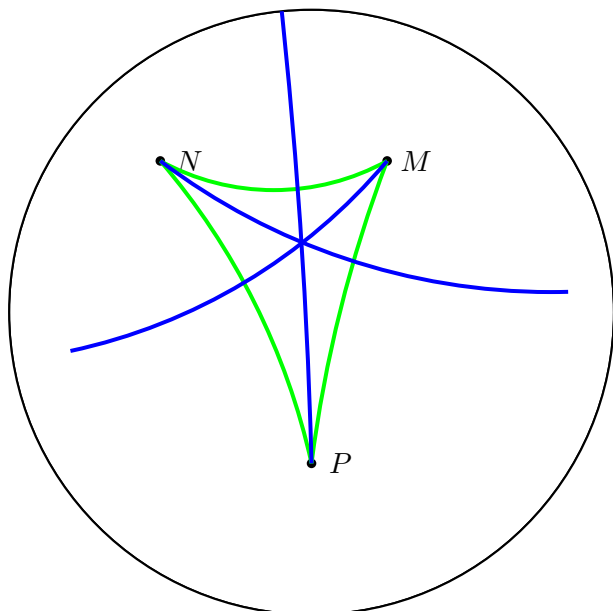
```
\psset{linewidth=0.4\pslinewidth,PointSymbol=x,
      nodesep=0,linecolor=magenta}
\pstGeonode[PointName=none]{0}(2,0){0'}
\pstCircleOA[linecolor=black]{0}{0'}
\multido{\n=5+5}{72}{%
  \pstGeonode[PointSymbol=none, PointName=none]
    (2;\n){M_\n}
  \pstCircleOA{M_\n}{0'}}
```

2.4 Homotethy and fractals



```
\pstGeonode[PosAngle={0,90}](2,2){A_0}(-2,2){B_0}%
A_0\psset{RotAngle=90}
\pstRotation[PosAngle=270]{A_0}{B_0}[D_0]
\pstRotation[PosAngle=180]{D_0}{A_0}[C_0]
\pspolygon(A_0)(B_0)(C_0)(D_0)%
\psset{PointSymbol=none, PointName=none, HomCoef=.2}
\multido{\n=1+1,\i=0+1}{20}{%
  \pstHom0[PosAngle=0]{B_\i}{A_\i}[A_\n]
  \pstHom0[PosAngle=90]{C_\i}{B_\i}[B_\n]
  \pstHom0[PosAngle=180]{D_\i}{C_\i}[C_\n]
  \pstHom0[PosAngle=270]{A_\i}{D_\i}[D_\n]
  \pspolygon(A_\n)(B_\n)(C_\n)(D_\n)}% fin multido
```

2.5 hyperbolic geometry: a triangle and its altitudes



Appendix A

Glossaire des commandes

Here is the complete macros list defined by `pst-eucl`. Each is shown with a short description and its parameters which control it. It is obvious that some over `PSTricks` parameters can be used, especially the ones which control the drawing of the line (width, style, color).

The parameters of `pst-eucl`

Paramètre	Défaut	Signification
<code>PointSymbol</code>	<code>default</code>	Symbol used for drawing a point.
<code>PointSymbolA</code>	<code>default</code>	idem for the first point of several.
<code>PointSymbolB</code>	<code>default</code>	for the second...
<code>PointSymbolC</code>	<code>default</code>	for the third...
<code>PointName</code>	<code>default</code>	Point's label.
<code>PointNameA</code>	<code>default</code>	idem for the first point of several.
<code>PointNameB</code>	<code>default</code>	for the second...
<code>PointNameC</code>	<code>default</code>	for the third...
<code>PtNameMath</code>	<code>true</code>	boolean parameter for (de)-activate the math style for the point name..
<code>SegmentSymbol</code>	<code>default</code>	Symbol used for marking a segment
<code>SegmentSymbolA</code>	<code>default</code>	idem for the first segment of a macro which marks several.
<code>SegmentSymbolB</code>	<code>default</code>	for the second...
<code>SegmentSymbolC</code>	<code>default</code>	for the third...
<code>Mark</code>	<code>default</code>	the mark symbol for an angle.
<code>MarkAngle</code>	<code>default</code>	angle for the precedent symbol.
<code>PointNameSep</code>	<code>1em</code>	Distance from the label and a point.
<code>PosAngle</code>	<code>undef</code>	Label position around the point.
<code>PosAngleA</code>	<code>undef</code>	idem for the first point.
<code>PosAngleB</code>	<code>undef</code>	for the second...
<code>PosAngleC</code>	<code>undef</code>	for the third...
<code>RightAngleSize</code>	<code>.4</code>	size for the right angle symbol
<code>RightAngleType</code>	<code>default</code>	Right angle type, possible value: <code>german</code> et <code>suisseromand</code>
<code>MarkAngleRadius</code>	<code>.4</code>	Radius of the angle mark.
<code>LabelAngleOffset</code>	<code>0</code>	Angular offset for the angle label.
<code>LabelSep</code>	<code>1</code>	Distance from the label and the angle top and its label.
<code>LabelRefPt</code>	<code>c</code>	Reference point <code>T_EX</code> used for the angle label.
<code>HomCoef</code>	<code>.5</code>	Homothetic angle.
... à suivre ...		

Paramètre	Défaut	Signification
RotAngle	60	Rotation angle.
DrawCirABC	true	Boolean parameter driving the drawing of the circumcircle.
CodeFig	false	Boolean parameter driving the coding of the construct.
CodeFigA	false	idem for the first...
CodeFigB	false	idem for the second...
CodeFigColor	cyan	Line color for the coding.
CodeFigStyle	dashed	Line style for the coding.
CodeFigArc	true	Boolean parameter driving the drawing of the bows around the first intersection.
CodeFigBarc	true	idem for the second...
Radius	none	Circle radius.
RadiusA	undef	For the first circle.
RadiusB	undef	For the second circle.
Diameter	none	Circle diameter.
DiameterA	undef	For the first circle.
DiameterB	undef	For the second circle.
DistCoef	none	Coefficient for modifying a distance/vector.
AngleCoef	none	Coefficient for modifying an angle.
CurvAbsNeg	false	Boolean parameter driving the direction of curved abscissa.
GenCurvFirst	none	Name of the first point of a generic curve (side effect).
GenCurvLast	none	Name of the last point of a generic curve (side effect).
GenCurvInc	none	Increment value for a generic curve.
CurveType	none	Drawing mode for a list of points.
TransformLabel	none	Label to be used for the rotation or the translation.

Appendix B

Compatibilité ascendantes de pst-eucl

Especially for this release, some macros have their syntax changed without changing their name, this mean that upward compatibility is not maintained. However, in order to help users it is possible to re-activate the old syntax by setting the option `old` when using the package `\usepackage[old]{pst-eucl}`. For this release this concern the macros for geometric transformations. You must refer to the latter manual for the syntax.