PSTricks

pst-euclide

A PSTricks package for drawing geometric pictures; v.1.52

October 19, 2015

Documentation by
Herbert Voß

Package author(s):
Dominique Rodriguez
Herbert Voß

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.

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 VTeX 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.

Thanks to: Manuel Luque; Thomas Söll.

Contents

I. The package 5

1. Special specifications 5
1.1. PSTricksOptions . . . 5
1.2. Conventions . . . 5

2. Basic Objects 5
2.1. Points . . . 5
2.2. Segment mark . . . 7
2.3. Triangles . . . 8
2.4. Angles . . . 8
2.5. Lines, half-lines and segments . . . 9
2.6. Circles . . . 10
2.7. Circle arcs . . . 11
2.8. Curved abscissa . . . 11
2.9. Generic curve . . . 12

3. Geometric Transformations 12
3.1. Central symmetry . . . 13
3.2. Orthogonal (or axial) symmetry . . . 13
3.3. Rotation . . . 14
3.4. Translation . . . 14
3.5. Homothetie . . . 15
3.6. Orthogonal projection . . . 15

4. Special object 15
4.1. Midpoint . . . 15
4.2. Triangle center of gravity . . . 16
4.3. Centre of the circumcircle of a triangle . . . 16
4.4. Perpendicular bisector of a segment . . . 17
4.5. Bisectors of angles . . . 18

5. Intersections 18
5.1. Line-Line . . . 18
5.2. Circle–Line . . . 19
5.3. Circle–Circle . . . 20
5.4. Function–function . . . 21
5.5. Function–line . . . 22
5.6. Function–Circle . . . 22

6. Helper Macros 23

II. Examples gallery 25

A. Basic geometry 25
A.1. Drawing of the bissector . . . 25
A.2. Transformation de polygones et courbes . . . 26
A.3. Triangle lines . . . 27
A.4. Euler circle . . . 28
A.5. Orthocenter and hyperbola . . . 29
A.6. 17 sides regular polygon . . . 30
A.7. Circles & tangents . . . 32
A.8. Fermat's point . . . 34
A.9. Escribed and inscribed circles of a triangle . . . 35

B. Some locus points 37
B.1. Parabola . . . 37
B.2. Hyperbola . . . 38
B.3. Cycloid . . . 39
B.4. Hypocycloids (Astroid and Deltoid) . . . 40

C. Lines and circles envelope 42
C.1. Conics . . . 42
C.2. Cardioid . . . 43

D. Homotethy and fractals 44

E. hyperbolic geometry: a triangle and its altitudes 45

F. List of all optional arguments for `pst-eucl` 46

References 47

Part I.
The package

1. Special specifications

1.1. PSTricksOptions

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.

1.2. 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.

2. Basic Objects

2.1. Points

`\pstGeonode[Options]`$(x_1,y_1)\{A_1\}(x_2,y_2)\{A_1\}\ldots(x,y)\{A_n\}$

This command defines one or more geometrical points associated with a node in the default cartesian coordinate system. Each point has a node name A_i which defines the default label put on the picture. This label is managed by default in mathematical mode, the boolean parameter `PtNameMath` (default `true`) can modify this behavior and let manage the label in normal mode. It is placed at a distance of `PointNameSep` (default 1em) of the center of the node with a angle of `PosAngle` (default 0). It is possible to specify another label using the parameter `PointName=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=`$*$. The symbol is the same as used by the macro `\psdot`. 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:

- *: ●
- o: ○
- +: +
- x: ×
- asterisk : ∗
- oplus: ⊕
- otimes: ⊗
- triangle: △
- triangle∗: ▲
- square: □
- square∗: ■
- diamond: ◊
- diamond∗: ⧫
- pentagon: ⬠
- pentagon∗: ⬟
- |: |

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

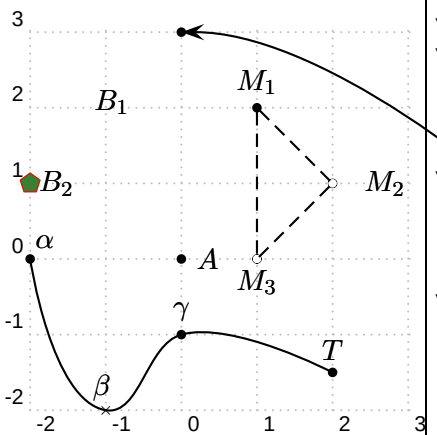
- their scale with dotscale, the value of whom is either two numbers defining the horizontal and vertical scale factor, or one single value being the same for both,
- their angle with parameter dotangle.

Please consult the PSTricksdocumentation for further details. 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 differenciate 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 setting CurveType=none can be used to draw a line between the points:

- opened polyline ;
- closed polygon ;
- open and curved curve.

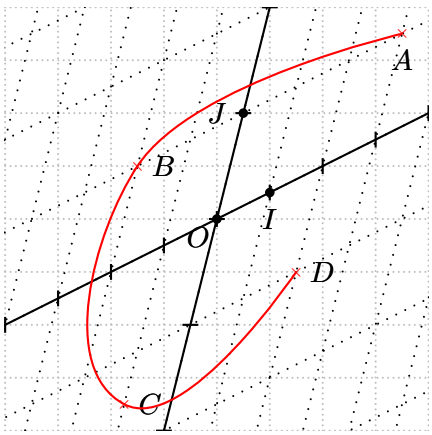



```
\begin{pspicture}[showgrid=true](-2,-2)(3,3)
\pstGeonode{A}
\pstGeonode[PosAngle=-135, PointNameSep=1.3](0,3){B_1}
\pstGeonode[PointSymbol=pentagon, dotscale=2, fillstyle=
    solid,
        fillcolor=OliveGreen, PtNameMath=false,
        PointName=$B_2$, linecolor=red](-2,1){B2}
\pstGeonode[PosAngle={90,0,-90}, PointSymbol={*,o},
        linestyle=dashed, CurveType=polygon,
        PointNameSep={1em,2em,3mm}]
 (1,2){M_1}(2,1){M_2}(1,0){M_3}
\pstGeonode[PosAngle={50,100,90}, PointSymbol={*,x,default
    },
        PointNameSep=3mm, CurveType=curve,
        PointName={\alpha,\beta,\gamma,default}]
 (-2,0){alpha}(-1,-2){beta}(0,-1){gamma}(2,-1.5){T}
\end{pspicture}
```

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

\pstOIJGeonode creates a list of points in the landmark $(O;I;J)$. Possible parameters are PointName, PointNameSep, PosAngle, PointSymbol, and PtNameMath.

\pstOIJGeonode [Options] $(x_1,y_1)\{A_1\}\{O\}\{I\}\{J\}$ $(x_2,y_2)\{A_2\}\ldots(x,y)\{A_n\}$




```
\psset{unit=.7}
\begin{pspicture*}[showgrid=true](-4,-4)(4,4)
  \pstGeonode[PosAngle={-135,-90,180}]{O}(1,0.5){I
      }(0.5,2){J}
  \pstLineAB[nodesep=10]{O}{I}
  \pstLineAB[nodesep=10]{O}{J}
  \multips(-5,-2.5)(1,0.5){11}{\psline(0,-.15)(0,.15)}
  \multips(-2,-8)(0.5,2){9}{\psline(-.15,0)(.15,0)}
  \psset{linestyle=dotted}%
  \multips(-5,-2.5)(1,0.5){11}{\psline(-10,-40)(10,40)}
  \multips(-2,-8)(0.5,2){9}{\psline(-10,-5)(10,5)}
  \psset{PointSymbol=x, linestyle=solid}
  \pstOIJGeonode[PosAngle={-90,0}, CurveType=curve,
    linecolor=red] (3,1){A}{O}{I}{J}(-2,1){B}(-1,-1.5){C
       }(2,-1){D}
\end{pspicture*}
```

2.2. Segment mark

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

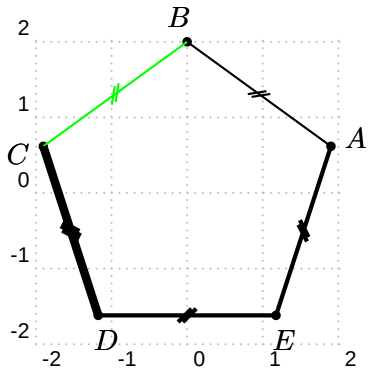
`\pstMarkSegment [Options] {A}{B}`

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:

- `\pstslash`
- `\pstslashh`
- `\pstslashhh`
- `\MarkHash`
- `\MarkHashh`
- `\MarkHashhh`
- `\MarkCros`
- `\MarkCross`

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




```
\begin{pspicture}[showgrid=true](-2,-2)(2,2)
  \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}
\end{pspicture}
```

The length and the separation of multiple hases can be set by `MarkHashLength` and `MarkHashSep`.

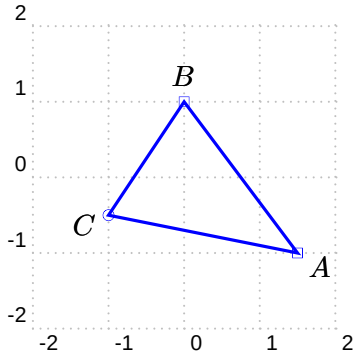
2.3. Triangles

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

\pstTriangle [Options] $(x_1,y_1)\{A\}(x_2,y_2)\{B\}(x_3,y_3)\{C\}$

Valid optional arguments are PointName, PointNameSep, PosAngle, PointSymbol, PointNameA, PosAngleA, PointSymbolA, PointNameB, PosAngleB, PointSymbolB, PointNameC, PosAngleC, and PointSymbolC. 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 $\langle A\rangle$, $\langle B\rangle$ and $\langle C\rangle$. 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.




```
\begin{pspicture}[showgrid](-2,-2)(2,2)
\pstTriangle[PointSymbol=square,PointSymbolC=o,
  linecolor=blue,linewidth=1.5\pslinewidth]
  (1.5,-1){A}(0,1){B}(-1,-.5){C}
\end{pspicture}
```

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 [Options] $\{A\}\{B\}\{C\}$

Valid optional arguments are RightAngleType, RightAngleSize, and RightAngleSize

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

- default : standard symbol ;
- german : german symbol (given by U. Dirr) ;
- suisseromand : swiss romand symbol (given P. Schnewlin).

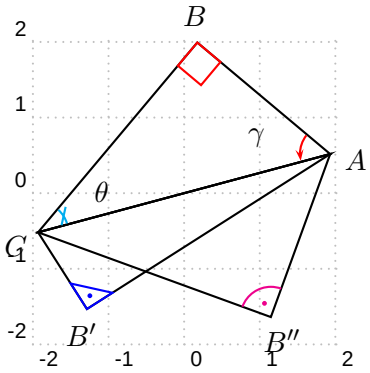
The only parameter controlling this command, excepting the ones which controlled the line, is RightAngleSize which defines the size of the symbol (by default 0.28 unit).

For other angles, there is the command:

\pstMarkAngle [Options] $\{A\}\{B\}\{C\}$

Valid optional arguments are MarkAngleRadius, LabelAngleOffset, and Mark 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`.



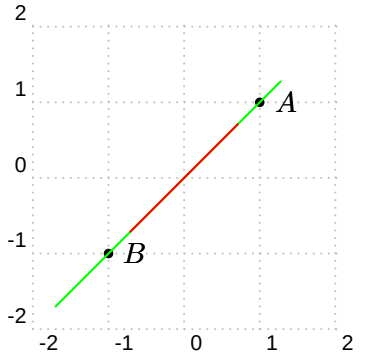

```
\begin{pspicture}[showgrid](-2,-2)(2,2)
\psset{PointSymbol=none}
\pstTriangle(2;15){A}(2;85){B}(2;195){C}
\psset{PointName=none}
\pstTriangle[PointNameA=default](2;-130){B'}(2;15){A
    '}(2;195){C'}
\pstTriangle[PointNameA=default](2;-55){B''}(2;15){A
    ''}(2;195){C''}
\pstRightAngle[linecolor=red]{C}{B}{A}
\pstRightAngle[linecolor=blue, RightAngleType=suisseromand
    ]{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}{$\gamma$}
\end{pspicture}
```

2.5. Lines, half-lines and segments

The classical line $(\overline{AB})$!

`\pstLineAB [Options] {A}{B}`

In order to control its length[1], 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.




```
\begin{pspicture}[showgrid](-2,-2)(2,2)
\pstGeonode(1,1){A}(-1,-1){B}
\pstLineAB[nodesepA=-.4,nodesepB=-1,
        linecolor=green]{A}{B}
\pstLineAB[nodesep=.4,linecolor=red]{A}{B}
\end{pspicture}
```

1 which is the comble for a line!

2.6. Circles

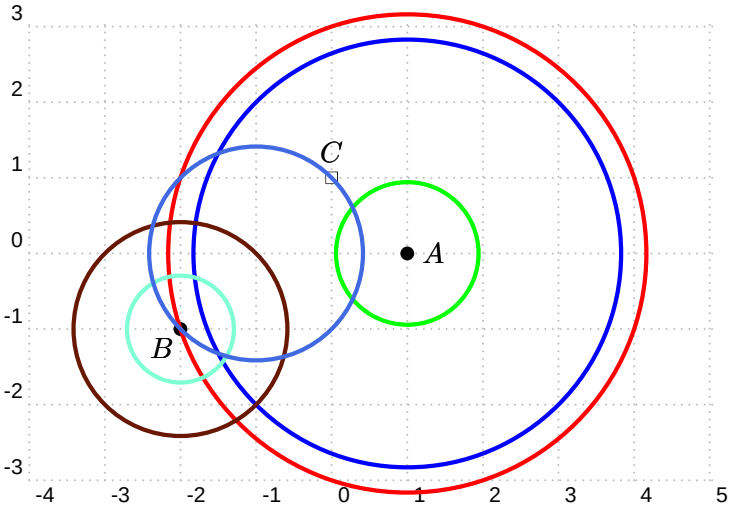
A circle can be defined either with its center and a point of its circumference, or with two diameterly opposed points. There is two commands :

```
\pstCircleOA [Options] {O}{A}
\pstCircleAB [Options] {O}{A}
\pstDistAB [Options] {A}{B}
\pstDistVal [Options] {x}
```

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 :

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. 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.

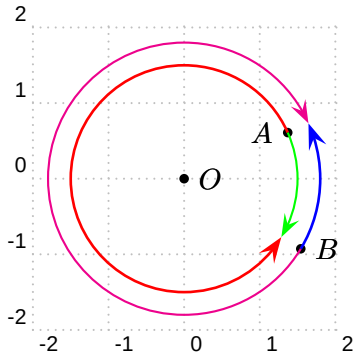


```
\begin{pspicture}[showgrid](-4,-4)(5,3)
\psset{linewidth=2\pslinewidth}
\pstGeonode[PosAngle={0,-135,90},PointSymbol={*,*,square}](1,0){A}(-2,-1){B}(0,1){C}
\pstCircleOA[linecolor=red]{A}{B}
\pstCircleOA[linecolor=green, DistCoef=2 3 div, Radius=\pstDistAB{A}{C}]{A}{}
\pstCircleOA[linecolor=blue, Radius=\pstDistAB{B}{C}]{A}{}
\pstCircleOA[linecolor=Sepia, Radius=\pstDistAB{A}{C}]{B}{}
\pstCircleOA[linecolor=Aquamarine, Diameter=\pstDistAB{A}{C}]{B}{}
\pstCircleAB[linecolor=RoyalBlue]{B}{C}
\end{pspicture}
```

2.7. Circle arcs

\pstArcOAB [Options] {O}{A}{B}
\pstArcnOAB [Options] {O}{A}{B}

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. It is not necessary that the two points are at the same distance of O.




```
\begin{pspicture}[showgrid](-2,-2)(2,2)
\pstGeonode[PosAngle={180,0}](1.5;24){A}(1.8;-31){B}
\pstGeonode{O}
\psset{arrows=->,arrowscale=2}
\pstArcOAB[linecolor=red,linewidth=1pt]{O}{A}{B}
\pstArcOAB[linecolor=blue,linewidth=1pt]{O}{B}{A}
\pstArcnOAB[linecolor=green]{O}{A}{B}
\pstArcnOAB[linecolor=magenta]{O}{B}{A}
\end{pspicture}
```

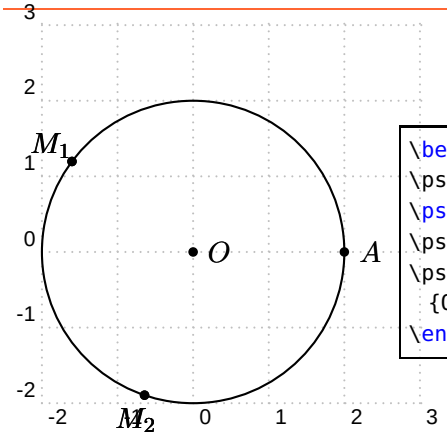
2.8. Curved abscissa

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

\pstCurvAbsNode [Options] {O}{A}{B}{Abs}

Possible optional arguments are `PointSymbol`, `PosAngle`, `PointName`, `PointNameSep`, `PtNameMath`, and `CurvAbsNeg`. 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 oirder to be alined with the circle center and the point.



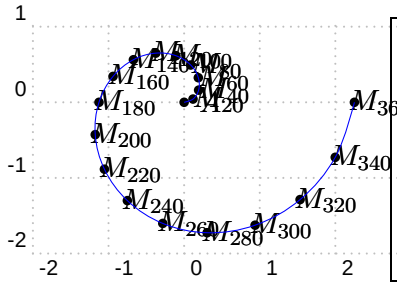

```
\begin{pspicture}[showgrid](-2.5,-2.5)(2.5,2.5)
\pstGeonode{O}(2,0){A}
\pstCircleOA{O}{A}
\pstCurvAbsNode{O}{A}{M_1}{\pstDistVal{5}}
\pstCurvAbsNode[CurvAbsNeg=true]%
 {O}{A}{M_2}{\pstDistAB{A}{M_1}}
\end{pspicture}
```

2.9. Generic 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 [Options] {`*Radical*`}{`n_1`}{`n_2`}`

Possible optional arguments are `GenCurvFirst`, `GenCurvInc`, and `GenCurvLast` The curve is drawn on the points whose name is defined using the radical ⟨Radical⟩ 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).




```
\begin{pspicture}[showgrid](-2.5,-2.5)(2.5,1)
\psset{unit=.00625}
\pstGeonode{A}
\multido{\n=20+20}{18}{%
 \pstGeonode[PointName=M_{\n}](\n;\n){M_\n}}
\pstGenericCurve[GenCurvFirst=A,GenCurvInc=20,
 linecolor=blue,linewidth=.5\pslinewidth]{M_}{20}{360}
\end{pspicture}
```

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 syntaxic 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.

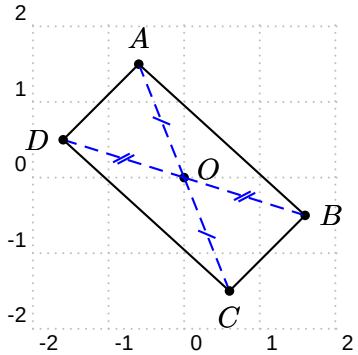
The first shared parameter is `CodeFig` which draws the specific constructions lines. Its default value is `false`, and a `true` value activates this optional drawing. The drawing is done using the line style `CodeFigStyle` (by default dashed), with the color `CodeFigColor` (by default cyan).

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.

3.1. Central symmetry

\pstSymO [Options] {O}{$M_1, M_2, \cdots, M_n$} [$M'_1, M'_2, \cdots, M'_p$]

Possible optional arguments are PointSymbol, PosAngle, PointName, PointNameSep, PtNameMath, CodeFig, CodeFigColor, and CodeFigStyle. 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.

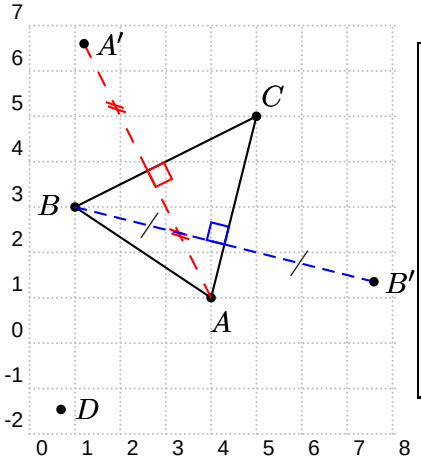



```
\begin{pspicture}[showgrid](-2,-2)(2,2)
\psset{CodeFig=true}
\pstGeonode[PosAngle={20,90,0}]{O}(-.6,1.5){A}(1.6,-.5){B}
\pstSymO[CodeFigColor=blue,
  PosAngle={-90,180}]{O}{A, B}[C, D]
\pstLineAB{A}{B}\pstLineAB{C}{D}
\pstLineAB{A}{D}\pstLineAB{C}{B}
\end{pspicture}
```

3.2. Orthogonal (or axial) symmetry

\pstOrtSym [Options] {A}{B}{$M_1, M_2, \cdots, M_n$} [$M'_1, M'_2, \cdots, M'_p$]

Possible optional arguments are PointSymbol, PosAngle, PointName, PointNameSep, PtNameMath, CodeFig, CodeFigColor, and CodeFigStyle. Draws the symmetric point in relation to line (AB).



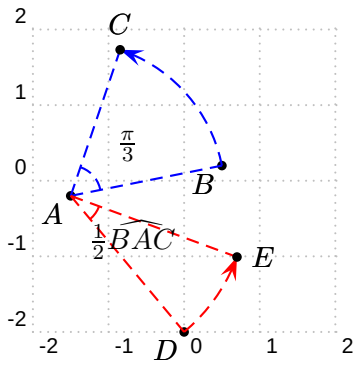

```
\psset{unit=0.6}
\begin{pspicture}[showgrid](0,-2)(8,7)
\pstTriangle(1,3){B}(5,5){C}(4,1){A}
\pstOrtSym{A}{B}{C}[D]
\psset{CodeFig=true}
\pstOrtSym[dash=2mm 2mm,CodeFigColor=red]%
  {C}{B}{A}
\pstOrtSym[SegmentSymbol=pstslash,
  linestyle=dotted,dotsep=3mm,CodeFigColor=blue]%
  {C}{A}{B}
\end{pspicture}
```

3.3. Rotation

\pstRotation [Options] {O}{$M_1, M_2, \cdots, M_n$} [$M'_1, M'_2, \cdots, M'_p$]
\pstAngleAOB{A}{O}{B}

Possible optional arguments are PointSymbol, PosAngle, PointName, PointNameSep, PtNameMath, and RotAngle for \pstRotation and AngleCoef, RotAngle for \pstAngleABC. 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:

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.



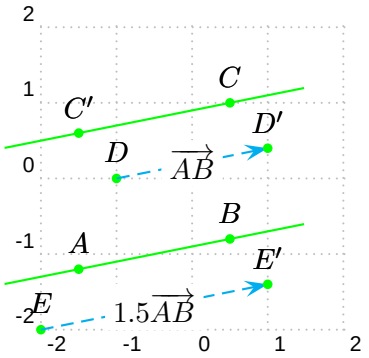

```
\begin{pspicture}[showgrid](-2,-2)(2,2)
\psset{arrowscale=2}
\pstGeonode[PosAngle=-135](-1.5,-.2){A}%
  (.5,.2){B}(0,-2){D}
\pstRotation[PosAngle=90,RotAngle=60,
  CodeFig,CodeFigColor=blue,
  TransformLabel=\frac{\pi}{3}]{A}{B}[C]
\pstRotation[AngleCoef=.5,
  RotAngle=\pstAngleAOB{B}{A}{C},
  CodeFigColor=red,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

3.4. Translation

\pstTranslation [Options] {A}{B}{$M_1, M_2, \cdots, M_n$} [$M'_1, M'_2, \cdots, M'_p$]

Possible optional arguments are PointSymbol, PosAngle, PointName, PointNameSep, PtNameMath, and DistCoef Draws the translated M'_i of M_i using the vector $\vec{AB}$. Useful for drawing a parallel line.

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).

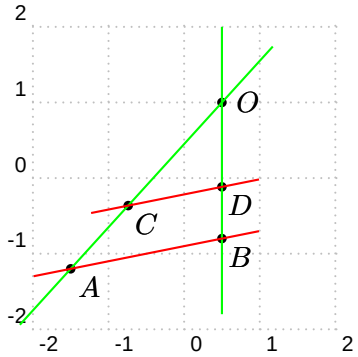



```
\begin{pspicture}[showgrid](-2,-2)(2,2)
\psset{linecolor=green,nodesep=-1,
  PosAngle=90,arrowscale=2}
\pstGeonode(-1.5,-1.2){A}(.5,-.8){B}(.5,1){C}(-1,0){D
    }(-2,-2){E}
\pstTranslation{B}{A}{C}
\psset{CodeFig,TransformLabel=default}
\pstTranslation{A}{B}{D}
\pstTranslation[DistCoef=1.5]{A}{B}{E}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

3.5. Homothetie

\pstHomO [Options] {O}{$M_1, M_2, \cdots, M_n$} [$M'_1, M'_2, \cdots, M'_p$]

Possible optional arguments are HomCoef, PointSymbol, PosAngle, PointName, PointNameSep, PtNameMath, and HomCoef. Draws M'_i the image of M_i by the homotethy of center O and coefficient specified with the parameter HomCoef.

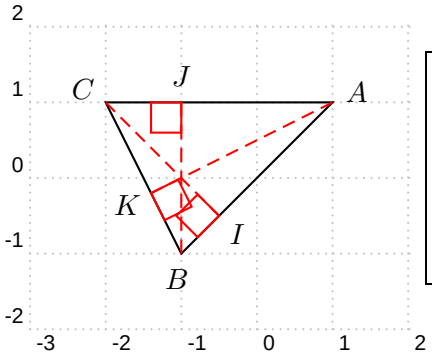



```
\begin{pspicture}[showgrid](-2,-2)(2,2)
\pstGeonode[PosAngle={0,-45}](.5,1){O}%
     (-1.5,-1.2){A}(.5,-.8){B}
\pstHomO[HomCoef=.62,PosAngle=-45]{O}{A,B}[C,D]
\psset{linecolor=green,nodesep=-1}
\pstLineAB{A}{O}\pstLineAB{B}{O}
\psset{linecolor=red,nodesep=-.5}
\pstLineAB{A}{B}\pstLineAB{C}{D}
\end{pspicture}
```

3.6. Orthogonal projection

\pstProjection [Options] {A}{B}{$M_1, M_2, \cdots, M_n$} [$M'_1, M'_2, \cdots, M'_p$]

Possible optional arguments are PointSymbol, PosAngle, PointName, PointNameSep, PtNameMath, CodeFig, CodeFigColor, andCodeFigStyle 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.



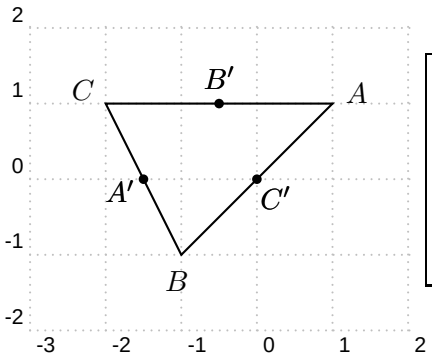

```
\begin{pspicture}[showgrid](-3,-2)(2,2)
\psset{PointSymbol=none,CodeFig,CodeFigColor=red}
\pstTriangle(1,1){A}(-2,1){C}(-1,-1){B}
\pstProjection{A}{B}{C}[I]
\pstProjection{A}{C}{B}[J]
\pstProjection{C}{B}{A}[K]
\end{pspicture}
```

4. Special object

4.1. Midpoint

\pstMiddleAB [Options] {A}{B}{I}

PointSymbol, PosAngle, PointName, PointNameSep, PtNameMath, SegmentSymbol, CodeFig, CodeFigColor, and CodeFigStyle Draw the midpoint I of segment $[AB]$. By default, the point name is automatically put below the segment.

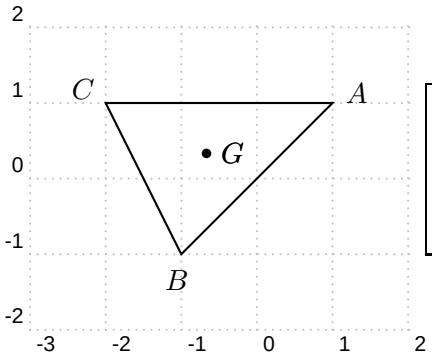



```
\begin{pspicture}[showgrid](-3,-2)(2,2)
\pstTriangle[PointSymbol=none]%
  (1,1){A}(-1,-1){B}(-2,1){C}
\pstMiddleAB{A}{B}{C'}
\pstMiddleAB{C}{A}{B'}
\pstMiddleAB{B}{C}{A'}
\end{pspicture}
```

4.2. Triangle center of gravity

\pstCGravABC [Options] {A}{B}{C}{G}

Possible optional arguments are PointName, PointNameSep, PosAngle, PointSymbol, and PtNameMath Draw the ABC triangle centre of gravity G.

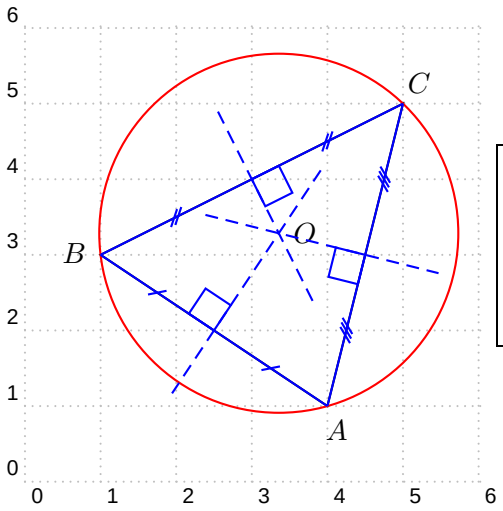



```
\begin{pspicture}[showgrid](-3,-2)(2,2)
\pstTriangle[PointSymbol=none]%
  (1,1){A}(-1,-1){B}(-2,1){C}
\pstCGravABC{A}{B}{C}{G}
\end{pspicture}
```

4.3. Centre of the circumcircle of a triangle

\pstCircleABC [Options] {A}{B}{C}{O}

Possible optional arguments are PointName, PointNameSep, PosAngle, PointSymbol, PtNameMath, DrawCirABC, CodeFig, CodeFigColor, CodeFigStyle, SegmentSymbolA, SegmentSymbolB, and SegmentSymbolC. 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 segemnts using three different marks given by the parameters SegmentSymbolA, SegmentSymbolB et SegmentSymbolC.



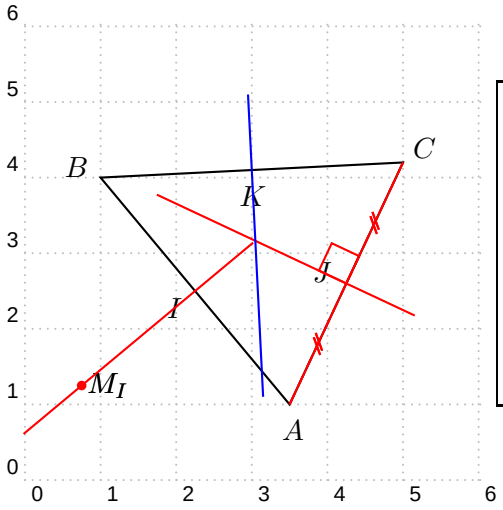

```
\begin{pspicture}[showgrid](6,6)
\pstTriangle[PointSymbol=none]%
  (4,1){A}(1,3){B}(5,5){C}
\pstCircleABC[CodeFig,CodeFigColor=blue,
  linecolor=red,PointSymbol=none]{A}{B}{C}{O}
\end{pspicture}
```

4.4. Perpendicular bisector of a segment

`\pstMediatorAB [Options] {`A`}{`B`}{`I`}{`M`}`

Possible optional arguments are `PointName`, `PointNameSep`, `PosAngle`, `PointSymbol`, `PtNameMath`, `CodeFig`, `CodeFigColor`, `CodeFigStyle`, and `SegmentSymbol`. 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 build by a rotation of B of 90 degrees around I. This mean that the order of A and B is important, it controls the position of M. The command creates the two points M end I. The construction is controlled by the following parameters:

- `CodeFig`, `CodeFigColor` and `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` and `...B` ;
- parameters controlling the line drawing.

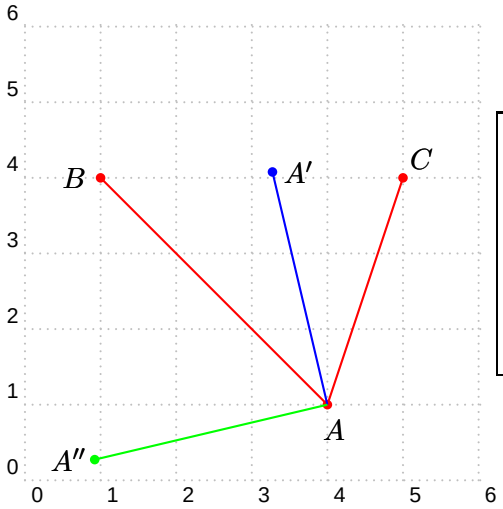



```
\begin{pspicture}[showgrid](6,6)
\pstTriangle[PointSymbol=none](3.5,1){A}(1,4){B
    }(5,4.2){C}
\psset{linecolor=red,CodeFigColor=red,nodesep=-1}
\pstMediatorAB[PointSymbolA=none]{A}{B}{I}{M_I}
\psset{PointSymbol=none,PointNameB=none}
\pstMediatorAB[CodeFig=true]{A}{C}{J}{M_J}
\pstMediatorAB[PosAngleA=45,linecolor=blue]
      {C}{B}{K}{M_K}
\end{pspicture}
```

4.5. Bisectors of angles

```
\pstBissectBAC [Options] {B}{A}{C}{N}
\pstOutBissectBAC [Options] {B}{A}{C}{N}
```

Possible optional arguments are PointSymbol, PosAngle, PointName, PointNameSep, and PtNameMath. 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.




```
\begin{pspicture}[showgrid](6,6)
\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''}
\end{pspicture}
```

5. Intersections

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

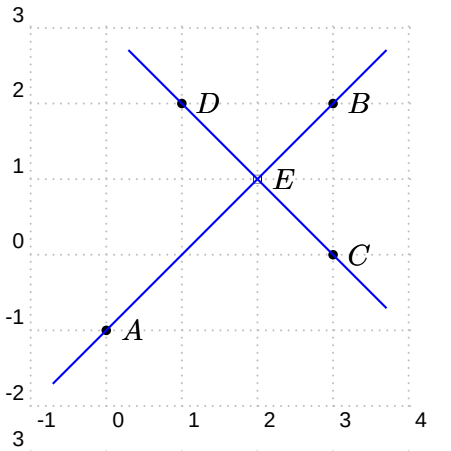
- line-line;
- line-circle;
- circle-circle;
- function-function;
- function-line;
- function-circle.

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.

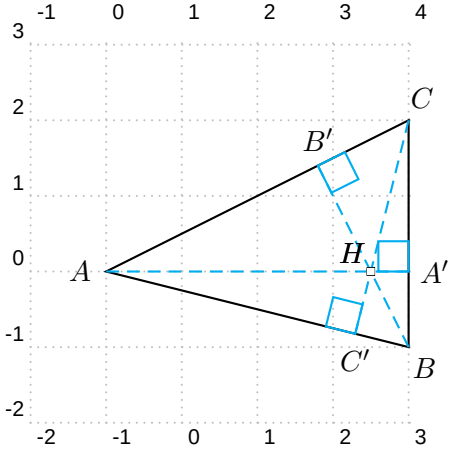
5.1. Line-Line

```
\pstInterLL [Options] {A}{B}{C}{D}{M}
```

Possible optional arguments are PointSymbol, PosAngle, PointName, PointNameSep, and PtNameMath. Draw the intersection point between lines (AB) and (CD).




```
\begin{pspicture}[showgrid](-1,-2)(4,3)
\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}
\end{pspicture}
```




```
\begin{pspicture}[showgrid](-2,-2)(3,3)
\psset{CodeFig,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}
\end{pspicture}
```

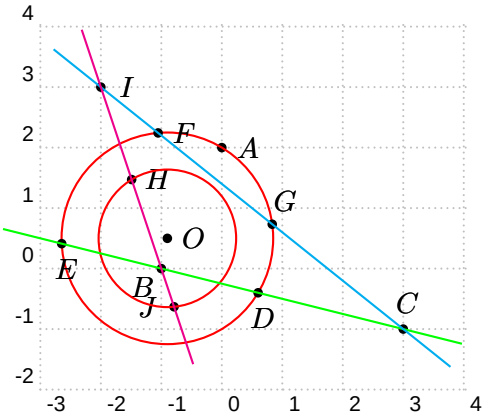
5.2. Circle–Line

\pstInterLC [Options] {A}{B}{O}{C}{M_1}{M_2}

Possible optional arguments are PointSymbol, PosAngle, PointName, PointNameSep, PtNameMath, PointSymbolA, PosAngleA, PointNameA, PointSymbolB, PosAngleB, PointNameB, Radius, and Diameter. 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 wo points is such that the vectors $\vec{AB}$ abd $\vec{M_1M_2}$ are in the same direction. Thus, if the points definig 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.

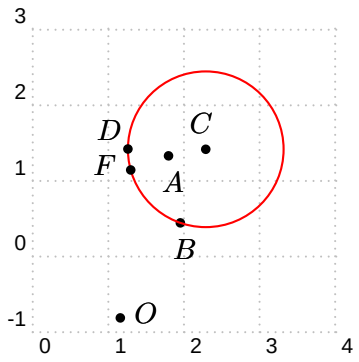



```
\psset{unit=0.8}
\begin{pspicture}[showgrid](-3,-2)(4,4)
\pstGeonode[PosAngle={-135,80,0}](-1,0){B}(3,-1){C
    }(-.9,.5){O}(0,2){A}
\pstGeonode(-2,3){I}
\pstCircleOA[linecolor=red]{O}{A}
\pstInterLC[PosAngle=-80]{C}{B}{O}{A}{D}{E}
\pstInterLC[PosAngleB=60, Radius=\pstDistAB{O}{D}]{I
    }{C}{O}{}{F}{G}
\pstInterLC[PosAngleB=180,DistCoef=1.3,Diameter=\
    pstDistAB{O}{D}]
  {I}{B}{O}{}{H}{J}
\pstCircleOA[linecolor=red,DistCoef=1.3,Diameter=\
    pstDistAB{O}{D}]{O}{}
\psset{nodesep=-1}
\pstLineAB[linecolor=green]{E}{C}
\pstLineAB[linecolor=cyan]{I}{C}
\pstLineAB[linecolor=magenta]{J}{I}
\end{pspicture}
```

5.3. Circle–Circle

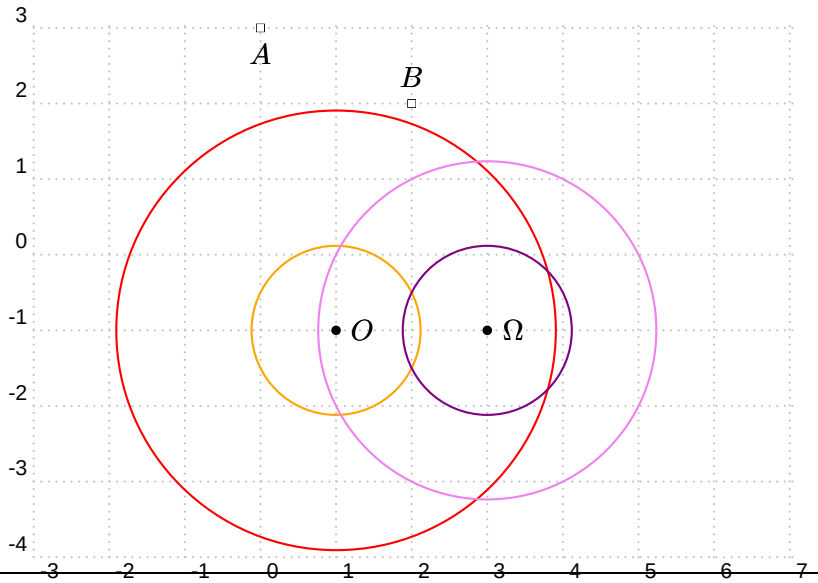
`\pstInterCC` `[Options]` {O_1}{B}{O_2}{C}{M_1}{M_2}

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 `CodeFigAarc` and `CodeFigBarc` specified the direction of these optional arcs: trigonometric (by default) or clockwise. Here is a first example.




```
\begin{pspicture}[showgrid](0,-1)(4,3)
\psset{dash=2mm 2mm}
\rput{10}{%
  \pstGeonode[PosAngle={0,-90,-90,90}]
    (1,-1){O}(2,1){A}(2,0.1){B}(2.5,1){C}}
\pstCircleOA[linecolor=red]{C}{B}
\pstInterCC[PosAngleA=135, CodeFigA=true, CodeFigAarc=
    false,
  CodeFigColor=magenta]{O}{A}{C}{B}{D}{E}
\pstInterCC[PosAngleA=170, CodeFigA=true,
   CodeFigAarc=false,
   CodeFigColor=green]{B}{E}{C}{B}{F}{G}
\end{pspicture}
```

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`.



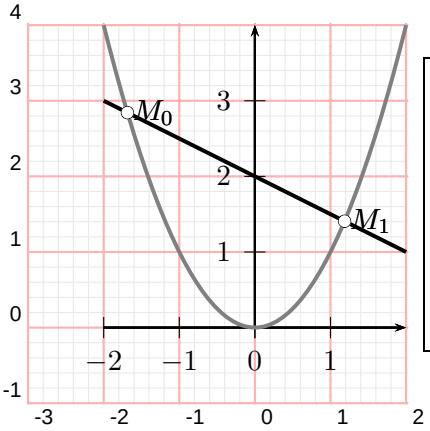

```
\begin{pspicture}[showgrid](-3,-4)(7,3)
\pstGeonode[PointName={\Omega,O}](3,-1){Omega}(1,-1){O}
\pstGeonode[PointSymbol=square, PosAngle={-90,90}](0,3){A}(2,2){B}
\psset{PointSymbol=o}
\pstCircleOA[linecolor=red, DistCoef=1 3 10 div add, Radius=\pstDistAB{A}{B}]{O}{}
\pstCircleOA[linecolor=Orange, Diameter=\pstDistAB{A}{B}]{O}{}
\pstCircleOA[linecolor=Violet, Radius=\pstDistAB{A}{B}]{Omega}{}
\pstCircleOA[linecolor=Purple, Diameter=\pstDistAB{A}{B}]{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}
\end{pspicture}
```

5.4. Function–function

\pstInterFF [Options] {f}{g}{x_0}{M}

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 ise this function several time if more than one intersection is present. Each function is describerd in PostScriptin the same way as the description used by the \psplot macro of PSTricks. A constant function can be specified, and then seaching function root is possible.

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

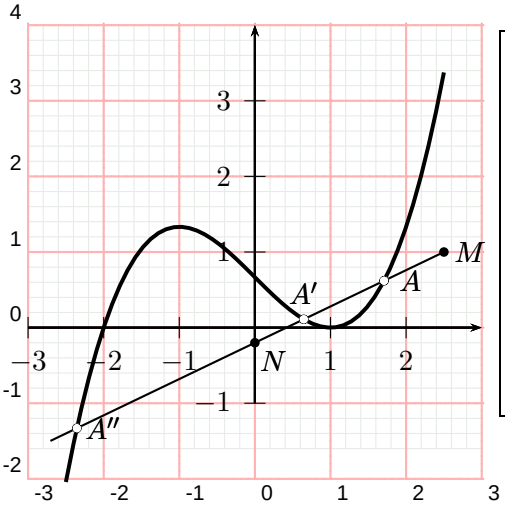



```
\begin{pspicture}[showgrid](-3,-1)(2,4)
\psaxes{->}(0,0)(-2,0)(2,4)
\psset{linewidth=1.5pt,algebraic}
\psplot[linecolor=gray]{-2}{2}{x^2}
\psplot{-2}{2}{2-x/2}
\psset{PointSymbol=o}
\pstInterFF{2-x/2}{x^2}{1}{M_1}
\pstInterFF{2-x/2}{x^2}{-2}{M_0}
\end{pspicture}
```

5.5. Function–line

\pstInterFL [Options] {f}{A}{B}{x_0}{M}

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

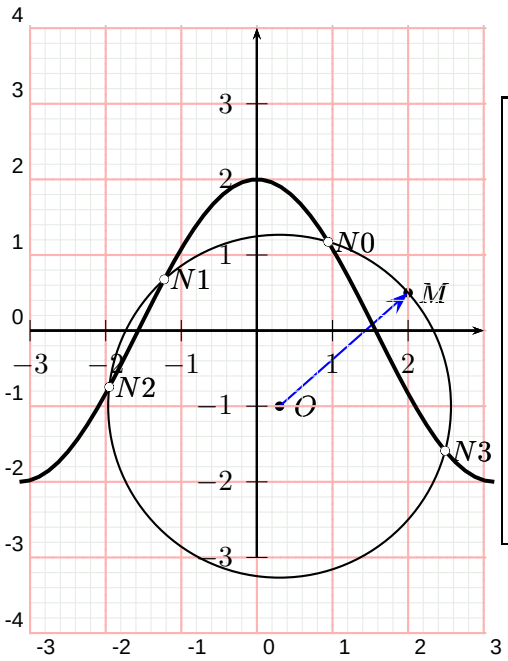



```
\begin{pspicture}[showgrid](-3,-1.5)(3,4)
\def\F{x^3/3 - x + 2/3 }
\psaxes{->}(0,0)(-3,-1)(3,4)
\psplot[linewidth=1.5pt,algebraic]{-2.5}{2.5}{\F}
\psset{PointSymbol=*}
\pstGeonode[PosAngle={-45,0}](0,-.2){N}(2.5,1){M}
\pstLineAB[nodesepA=-3cm]{N}{M}
\psset{PointSymbol=o,algebraic}
\pstInterFL{\F}{N}{M}{2}{A}
\pstInterFL[PosAngle=90]{\F}{N}{M}{0}{A'}
\pstInterFL{\F}{N}{M}{-2}{A''}
\end{pspicture}
```

5.6. Function–Circle

\pstInterFC [Options] {f}{O}{A}{x_0}{M}

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

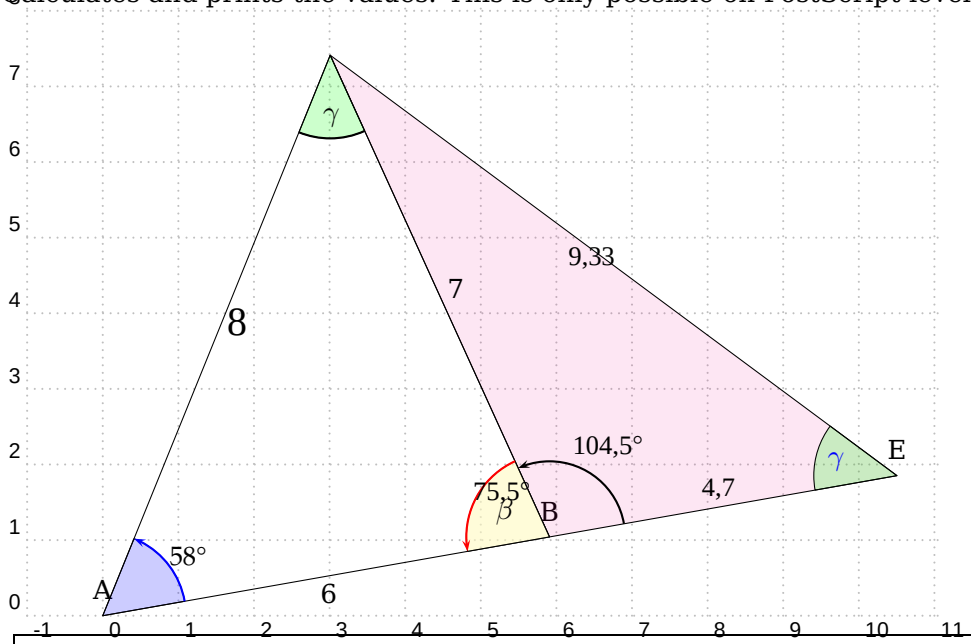



```
\begin{pspicture}[showgrid](-3,-4)(3,4)
\def\F{2*cos(x)}
\psset{algebraic}
\pstGeonode(0.3,-1){O}(2,.5){M}
\ncline[linecolor=blue, arrowscale=2]{->}{O}{M}
\psaxes{->}(0,0)(-3,-3)(3,4)
\psplot[linewidth=1.5pt]{-3.14}{3.14}{\F}
\pstCircleOA[PointSymbol=*]{O}{M}
\psset{PointSymbol=o}
\pstInterFC{\F}{O}{M}{1}{N0}
\pstInterFC{\F}{O}{M}{-1}{N1}
\pstInterFC{\F}{O}{M}{-2}{N2}
\pstInterFC{\F}{O}{M}{2}{N3}
\end{pspicture}
```

6. Helper Macros

```
\psGetDistanceAB [Options] (x1,y1)(x2,y2){<name>}
\psGetAngleABC [Options] (x1,y1)(x2,y2)(x3,y3){<symbol>}
```

Calculates and prints the values. This is only possible on PostScript level!



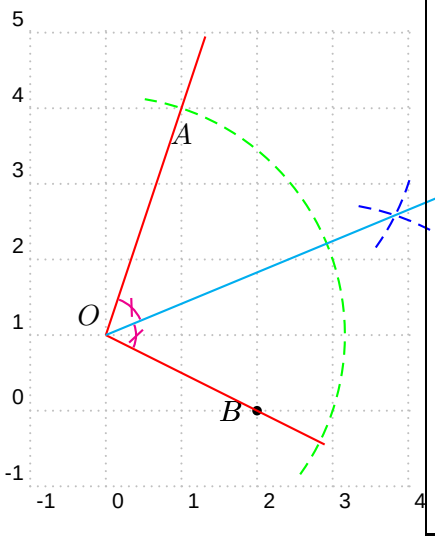

```
\begin{pspicture}(-1,0)(11,8)
\psgrid[gridlabels=0pt,subgriddiv=2,gridwidth=0.4pt,subgridwidth=0.2pt,gridcolor=black
    !60,subgridcolor=black!40]
\def\sideC{6} \def\sideA{7} \def\sideB{8}
\psset{PointSymbol=none,linejoin=1,linewidth=0.4pt,PtNameMath=false,labelsep=0.07,
    MarkAngleRadius=1.1,decimals=1,comma}
\pstGeonode[PosAngle={90,90}](0,0){A}(\sideC;10){B}
```

```
%   \pstGeonode[PosAngle={225,-75}](0,0){A}(\sideC;10){B}
\psset{PointName=}
\pstInterCC[RadiusA=\pstDistVal{\sideB},RadiusB=\pstDistVal{\sideA},PosAngle=-90,
    PointNameA=C]{A}{}{B}{}{C}{C-}
\pstInterCC[RadiusA=\pstDistAB{A}{B},RadiusB=\pstDistAB{B}{C}]{C}{}{A}{}{D-}{D}
\pstInterLC[Radius=\pstDistAB{A}{C}]{C}{D}{C}{}{A'-}{A'}
\pstInterCC[RadiusA=\pstDistAB{A}{B},RadiusB=\pstDistAB{B}{C}]{A'}{}{C}{}{B'}{B'-}
\pstInterLL[PosAngle=90,PointName=default]{B'}{C}{A}{B}{E}
\pspolygon(A)(B)(C)
\pspolygon[fillstyle=solid,fillcolor=magenta,opacity=0.1](C)(E)(B)
%
\psGetAngleABC[ArcColor=blue,AngleValue=true,LabelSep=0.7,arrows=->,decimals=0,PSfont=
    Palatino-Roman](B)(A)(C){}
\psGetAngleABC[AngleValue=true,ArcColor=red,arrows=->,WedgeOpacity=0.6,WedgeColor=yellow
    !30,LabelSep=0.5](C)(B)(A){$\beta$}
\psGetAngleABC[LabelSep=0.7,WedgeColor=green,xShift=-6,yShift=-10](A)(C)(B){$\gamma$}
\psGetAngleABC[LabelSep=0.7,AngleArc=false,WedgeColor=green,arrows=->,xShift=-15,yShift
    =0](C)(E)(B){\color{blue}$\gamma$}
\psGetAngleABC[AngleValue=true,MarkAngleRadius=1.0,LabelSep=0.5,ShowWedge=false,xShift
    =-5,yShift=7,arrows=->](E)(B)(C){}
%
\pcline[linestyle=none](A)(B)\nbput{\sideC}
\pcline[linestyle=none](C)(B)\naput{\sideA}
\psGetDistanceAB[xShift=-8,yShift=4](B)(E){MW}
\psGetDistanceAB[fontscale=15,xShift=4,decimals=0](A)(C){MAC}
\psGetDistanceAB[xShift=-17,decimals=2](E)(C){MEC}
\end{pspicture}
```

Part II.
Examples gallery

A. Basic geometry

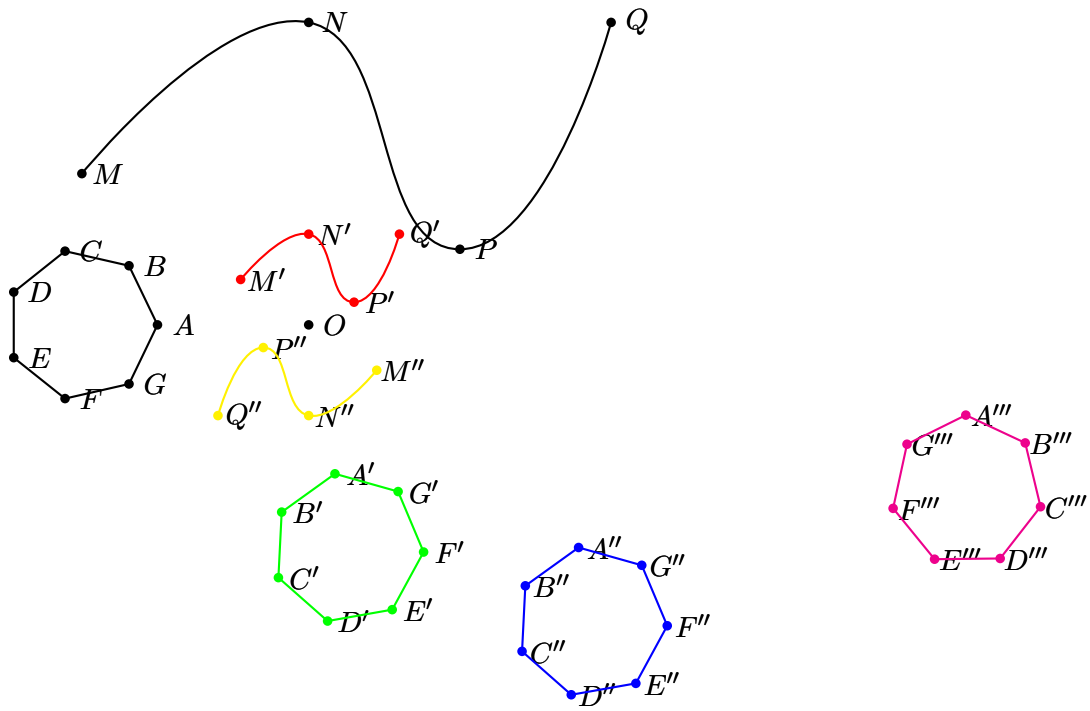
A.1. Drawing of the bissector




```
\begin{pspicture}[showgrid](-1,-1)(4.4,5)
\psset{PointSymbol=none,PointName=none}
\pstGeonode[PosAngle={180,130,-90},PointSymbol={default,
    none},
 PointName=default](2,0){B}(0,1){O}(1,4){A}
\pstLineAB[nodesepB=-1,linecolor=red]{O}{A}
\pstLineAB[nodesepB=-1,linecolor=red]{O}{B}
\pstInterLC[PosAngleB=-45]{O}{B}{O}{A}{G}{C}
\psset{arcsepA=-1, arcsepB=-1}
\pstArcOAB[linecolor=green,linestyle=dashed]{O}{C}{A}
\pstInterCC[PosAngleA=100]{A}{O}{C}{O}{O'}{OO}
\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=1pt,linecolor=magenta,Mark=MarkHash}
\pstMarkAngle{C}{O}{O'}{}
\pstMarkAngle[MarkAngleRadius=.5]{O'}{O}{A}{}
\end{pspicture}
```

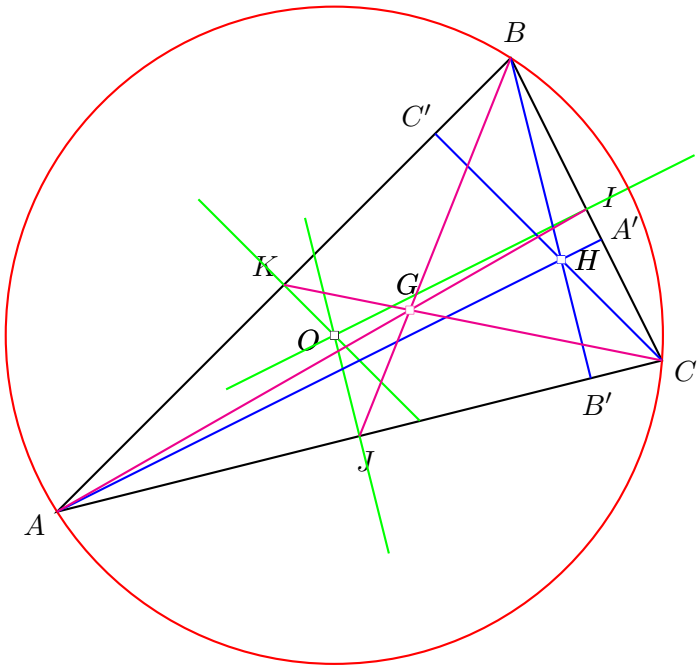
A.2. Transformation de polygones et courbes

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



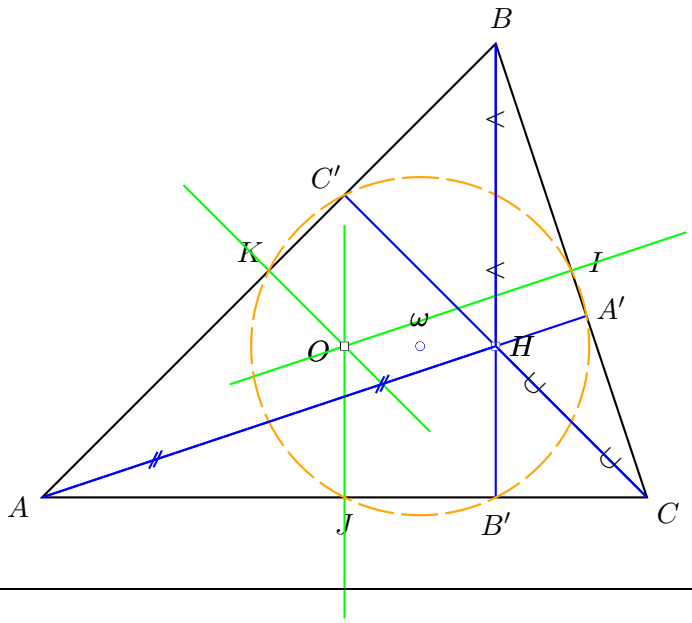

```
\begin{pspicture}(-5,-5)(10,5)
\pstGeonode{O}
\rput(-3,0){\pstGeonode[CurveType=polygon](1,0){A}(1;51.43){B}(1;102.86){C}
  (1;154.29){D}(1;205.71){E}(1;257.14){F}(1;308.57){G}}
\rput(-4,-1){\pstGeonode[CurveType=curve](1,3){M}(4,5){N}(6,2){P}(8,5){Q}}
\pstRotation[linecolor=green, RotAngle=100, 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''']
\end{pspicture}
```

A.3. Triangle lines




```
\psset{unit=2}
\begin{pspicture}(-3,-2)(3,3)
\psset{PointSymbol=none}
\pstTriangle[PointSymbol=none](-2,-1){A}(1,2){B}(2,0){C}
{ \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 parametres
  \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 medianes et le centre de gravite
\psset{linecolor=magenta}\ncline{A}{I}\ncline{C}{K}\ncline{B}{J}
\pstCGravABC[PointSymbol=square, PosAngle=95]{A}{B}{C}{G}
\end{pspicture}
```

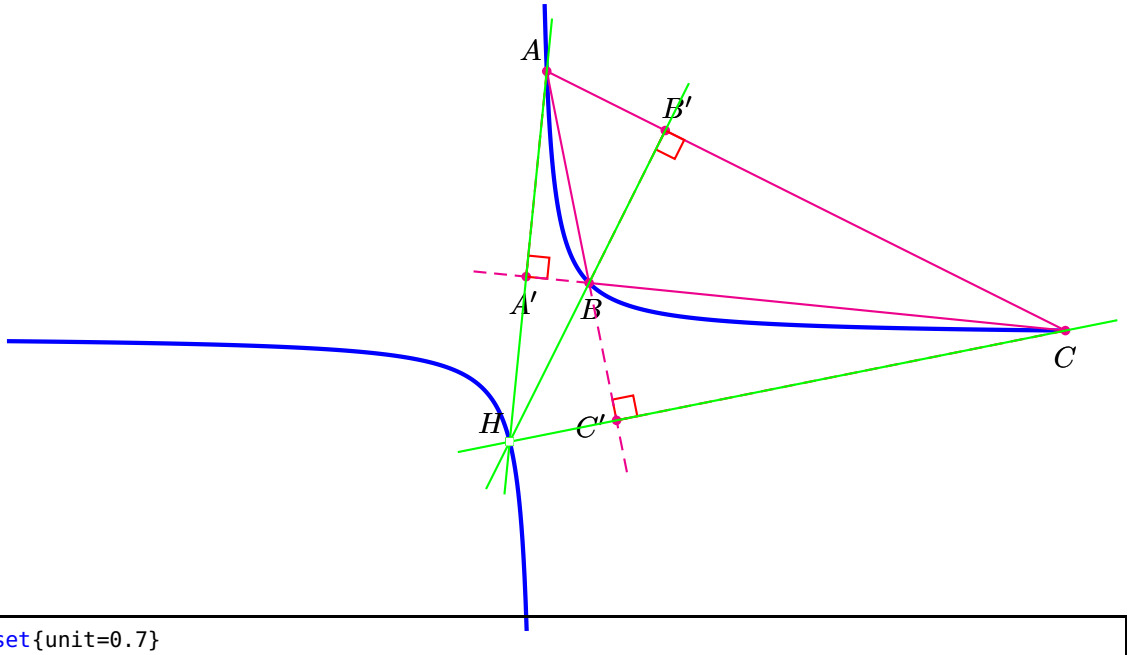
A.4. Euler circle




```
\psset{unit=2}
\begin{pspicture}(-3,-1.5)(3,2.5)
\psset{PointSymbol=none}
\pstTriangle(-2,-1){A}(1,2){B}(2,-1){C}
{% encapsulation de modif parametres
  \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 parametres
  \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}
\end{pspicture}
```

A.5. Orthocenter and hyperbola

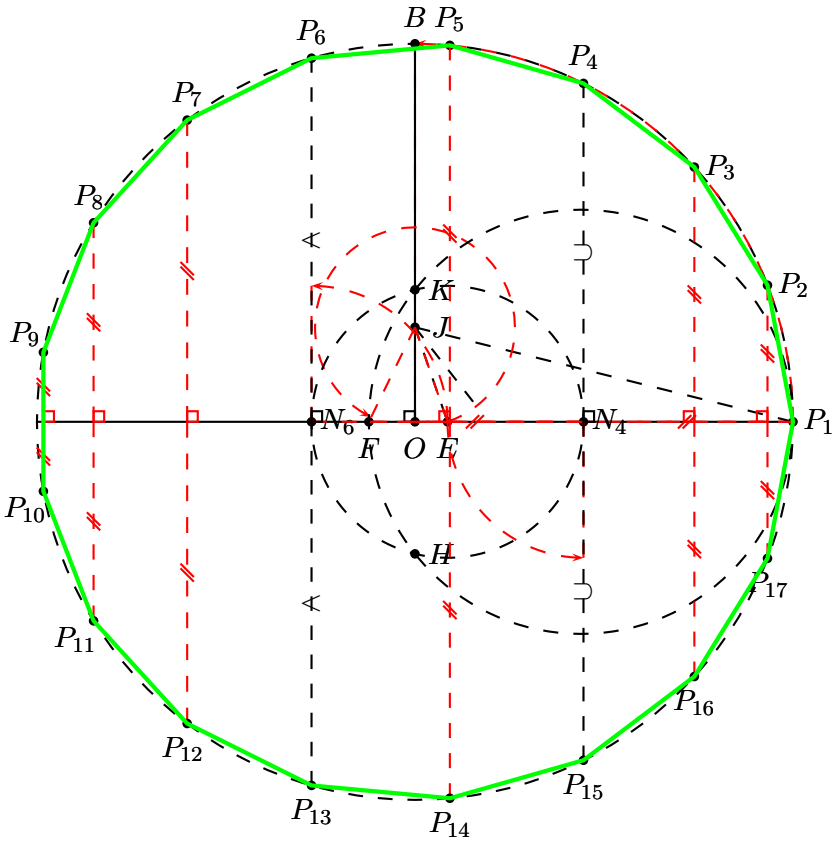
The orthocenter of a triangle whose points are on the branches of the hyperbola $\mathcal{H} : y = a/x$ belong to this hyperbola.




```
\psset{unit=0.7}
\begin{pspicture}(-11,-5)(11,7)
\psset{linecolor=blue, linewidth=2\pslinewidth}
\psplot[yMaxValue=6,plotpoints=500]{-10}{-.1}{1 x div}
\psplot[yMaxValue=6,plotpoints=500]{.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}
\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)
\end{pspicture}
```

A.6. 17 sides regular polygon

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

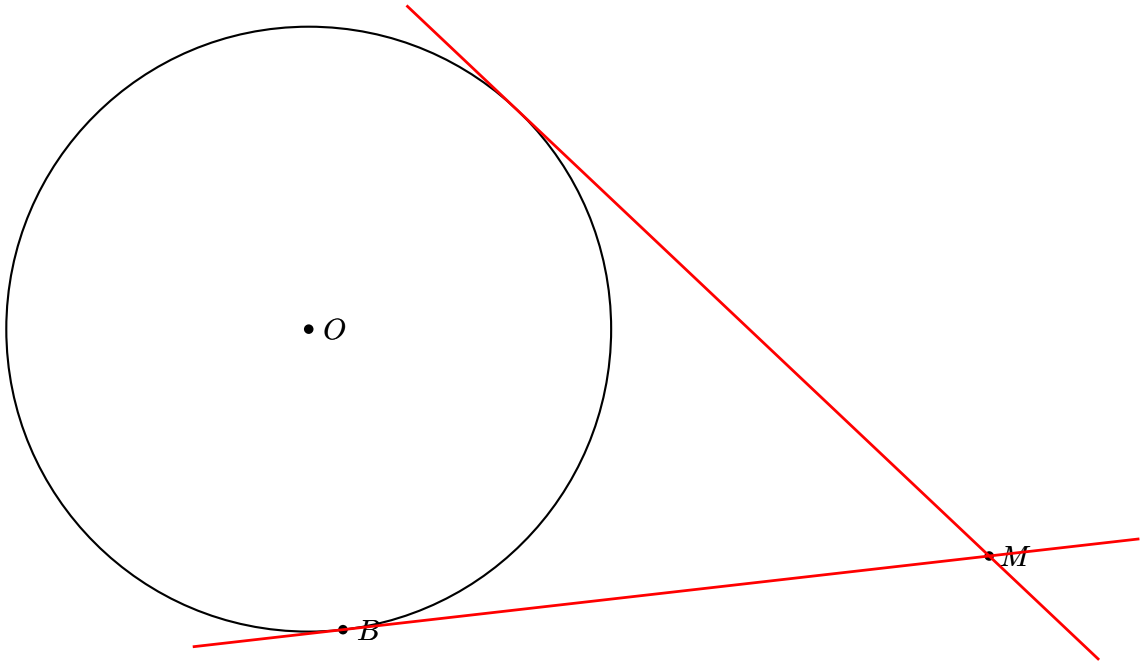



```
\begin{pspicture}(-5.5,-5.5)(5.5,6)
 \psset{CodeFig, RightAngleSize=.14, CodeFigColor=red,
   CodeFigB=true, linestyle=dashed, dash=2mm 2mm}
 \pstGeonode[PosAngle={-90,0}]{O}(5;0){P_1}
 \pstCircleOA{O}{P_1}
 \pstSymO[PointSymbol=none, PointName=none, CodeFig=false]{O}{P_1}[PP_1]
 \ncline[linestyle=solid]{PP_1}{P_1}
 \pstRotation[RotAngle=90, PosAngle=90]{O}{P_1}[B]
 \pstRightAngle[linestyle=solid]{B}{O}{PP_1}\ncline[linestyle=solid]{O}{B}
 \pstHomO[HomCoef=.25]{O}{B}[J] \ncline{J}{P_1}
 \pstBissectBAC[PointSymbol=none, PointName=none]{O}{J}{P_1}{PE1}
 \pstBissectBAC[PointSymbol=none, PointName=none]{O}{J}{PE1}{PE2}
 \pstInterLL[PosAngle=-90]{O}{P_1}{J}{PE2}{E}
 \pstRotation[PosAngle=-90, RotAngle=-45, PointSymbol=none, PointName=none]{J}{E}[PF1]
 \pstInterLL[PosAngle=-90]{O}{P_1}{J}{PF1}{F}
 \pstMiddleAB[PointSymbol=none, PointName=none]{F}{P_1}{MFP1} \pstCircleOA{MFP1}{P_1}
 \pstInterLC[PointSymbolA=none, PointNameA=none]{O}{B}{MFP1}{P_1}{H}{K}
 \pstCircleOA{E}{K} \pstInterLC{O}{P_1}{E}{K}{N_6}{N_4}
 \pstRotation[RotAngle=90,PointSymbol=none, PointName=none]{N_6}{E}[PP_6]
 \pstInterLC[PosAngleA=90,PosAngleB=-90, PointNameB=P_{13}]{PP_6}{N_6}{O}{P_1}{P_6}{P
     _13}
 \pstSegmentMark[SegmentSymbol=wedge]{N_6}{P_6}
 \pstSegmentMark[SegmentSymbol=wedge]{P_13}{N_6}
```

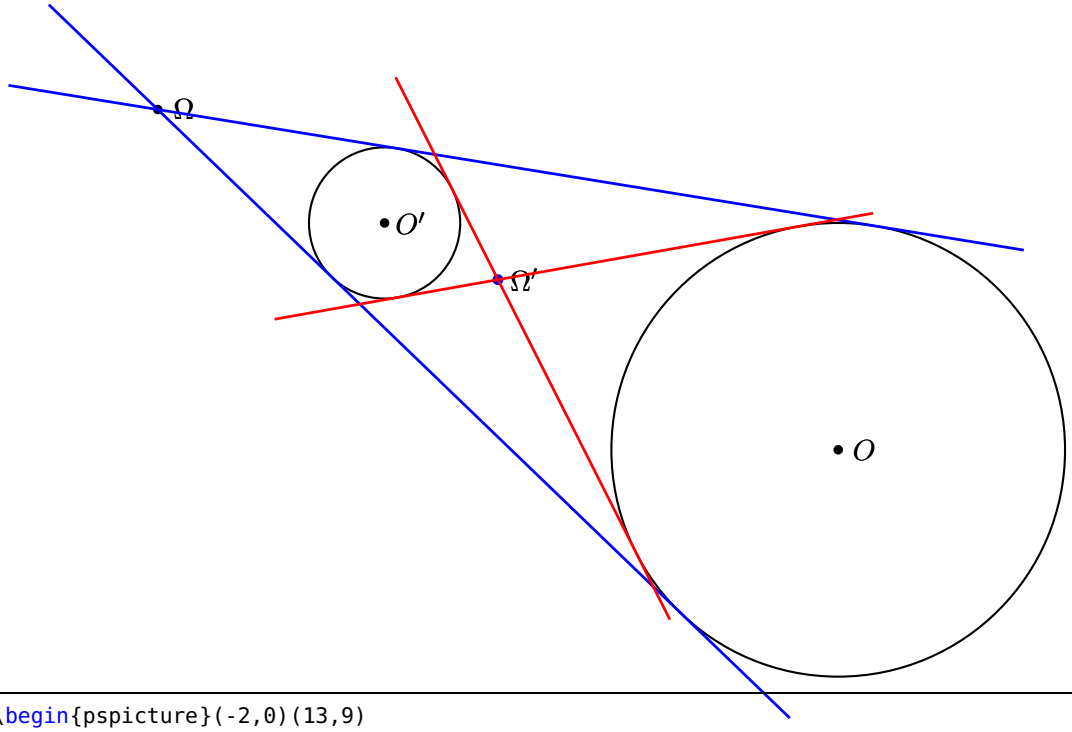
```
  \pstRotation[RotAngle=90,PointSymbol=none, PointName=none]{N_4}{E}[PP_4]
  \pstInterLC[PosAngleA=90,PosAngleB=-90,PointNameB=P_{15}]{N_4}{PP_4}{O}{P_1}{P_4}{P
      _15}
  \pstSegmentMark[SegmentSymbol=cup]{N_4}{P_4}
  \pstSegmentMark[SegmentSymbol=cup]{P_15}{N_4}
  \pstRightAngle[linestyle=solid]{P_1}{N_6}{P_6}
  \pstRightAngle[linestyle=solid]{P_1}{N_4}{P_4}
  \pstBissectBAC[PosAngle=90, linestyle=none]{P_4}{O}{P_6}{P_5}
  \pstInterCC[PosAngleB=90, PointSymbolA=none, PointNameA=none]{O}{P_1}{P_4}{P_5}{H}{P
      _3}
  \pstInterCC[PosAngleB=90, PointSymbolA=none, PointNameA=none]{O}{P_1}{P_3}{P_4}{H}{P
      _2}
  \pstInterCC[PosAngleA=90, PointSymbolB=none, PointNameB=none]{O}{P_1}{P_6}{P_5}{P_7}{H
      }
  \pstInterCC[PosAngleA=100, PointSymbolB=none, PointNameB=none]{O}{P_1}{P_7}{P_6}{P_8}{
      H}
  \pstInterCC[PosAngleA=135, PointSymbolB=none, PointNameB=none]{O}{P_1}{P_8}{P_7}{P_9}{
      H}
  \pstOrtSym[PosAngle={-90,-90,-90,-100,-135},PointName={P_{17},P_{16},P_{14},P_{12},P
      _{11},P_{10}}]
          {O}{P_1}{P_2,P_3,P_5,P_7,P_8,P_9}[P_17,P_16,P_14,P_12,P_11,P_10]
  \pspolygon[linecolor=green, linestyle=solid, linewidth=2\pslinewidth]
    (P_1)(P_2)(P_3)(P_4)(P_5)(P_6)(P_7)(P_8)(P_9)(P_10)(P_11)(P_12)(P_13)(P_14)(P_15)(P
        _16)(P_17)
\end{pspicture}
```

A.7. Circles & tangents

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



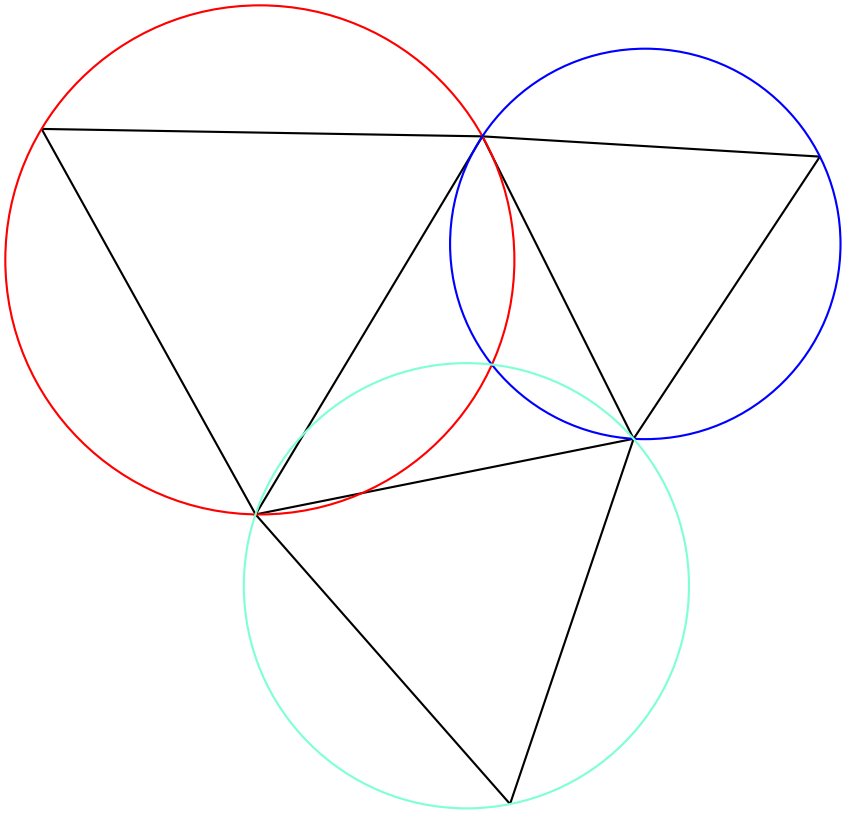

```
\begin{pspicture}(15,10)
\pstGeonode(5, 5){O}(14,2){M}
\pstCircleOA[Radius=\pstDistVal{4}]{O}{}
\pstMiddleAB[PointSymbol=none, PointName=none]{O}{M}{O'}
\pstInterCC[RadiusA=\pstDistVal{4}, DiameterB=\pstDistAB{O}{M},
         CodeFigB=true, CodeFigColor=magenta, PosAngleB=45]{O}{}{O'}{}{A}{B}
\psset{linecolor=red, linewidth=1.3\pslinewidth, nodesep=-2}
\pstLineAB{M}{A}\pstLineAB{M}{B}
\end{pspicture}
```




```
\begin{pspicture}(-2,0)(13,9)
\pstGeonode(9,3){O}(3,6){O'}\psset{PointSymbol=none, PointName=none}
\pstCircleOA[Radius=\pstDistVal{3}]{O}{}\pstCircleOA[Radius=\pstDistVal{1}]{O'}{}
\pstInterLC[Radius=\pstDistVal{3}]{O}{O'}{O}{}{M}{toto}
\pstInterLC[Radius=\pstDistVal{1}]{O}{O'}{O'}{}{M'}{toto}
\pstRotation[RotAngle=30]{O}{M}[N]
\pstRotation[RotAngle=30]{O'}{M'}[N']
\pstInterLL[PointSymbol=default, PointName=\Omega]{O}{O'}{N}{N'}{Omega}
\pstMiddleAB{O}{Omega}{I} \pstInterCC{I}{O}{O}{M}{A}{B}
\psset{nodesepA=-1, nodesepB=-3, linecolor=blue, linewidth=1.3\pslinewidth}
\pstLineAB[nodesep=-2]{A}{Omega}\pstLineAB[nodesep=-2]{B}{Omega}
\pstRotation[RotAngle=-150]{O'}{M'}[N'']
\pstInterLL[PointSymbol=default, PointName=\Omega']{O}{O'}{N}{N''}{Omega'}
\pstMiddleAB{O}{Omega'}{J}
\pstInterCC{J}{O}{O}{M}{A'}{B'}
\psset{nodesepA=-1, nodesepB=-3, linecolor=red}
\pstLineAB{A'}{Omega'}\pstLineAB{B'}{Omega'}
\end{pspicture}
```

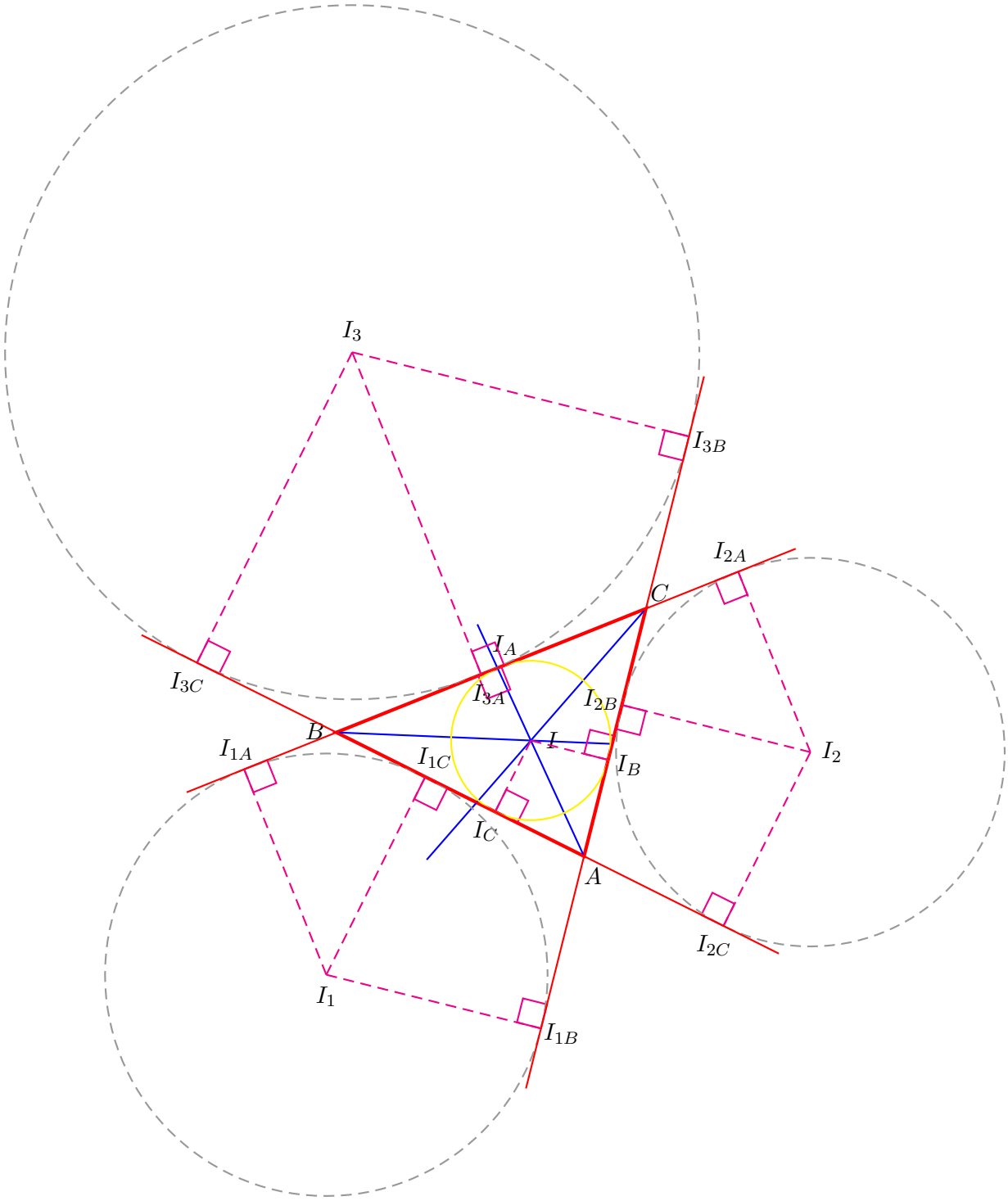
A.8. Fermat's point

Drawing of Manuel Luque.



```
\begin{pspicture}(-7,-6)(5,5)
\psset{PointSymbol=none, PointName=none}
\pstTriangle[PosAngleA=-160,PosAngleB=90,PosAngleC=-25](-3,-2){B}(0,3){A}(2,-1){C}%
\psset{RotAngle=-60}
\pstRotation[PosAngle=-90]{B}{C}[A']
\pstRotation{C}{A}[B']
\pstRotation[PosAngle=160]{A}{B}[C']
\pstLineAB{A}{B'}
\pstLineAB{C}{B'}
\pstLineAB{B}{A'}
\pstLineAB{C}{A'}
\pstLineAB{B}{C'}
\pstLineAB{A}{C'}
\pstCircleABC[linecolor=red]{A}{B}{C'}{O_1}
\pstCircleABC[linecolor=blue]{A}{C}{B'}{O_2}
\pstCircleABC[linecolor=Aquamarine]{A'}{C}{B}{O_3}
\pstInterCC[PointSymbolA=none]{O_1}{A}{O_2}{A}{E}{F}
\end{pspicture}
```

A.9. Escribed and inscribed circles of a triangle



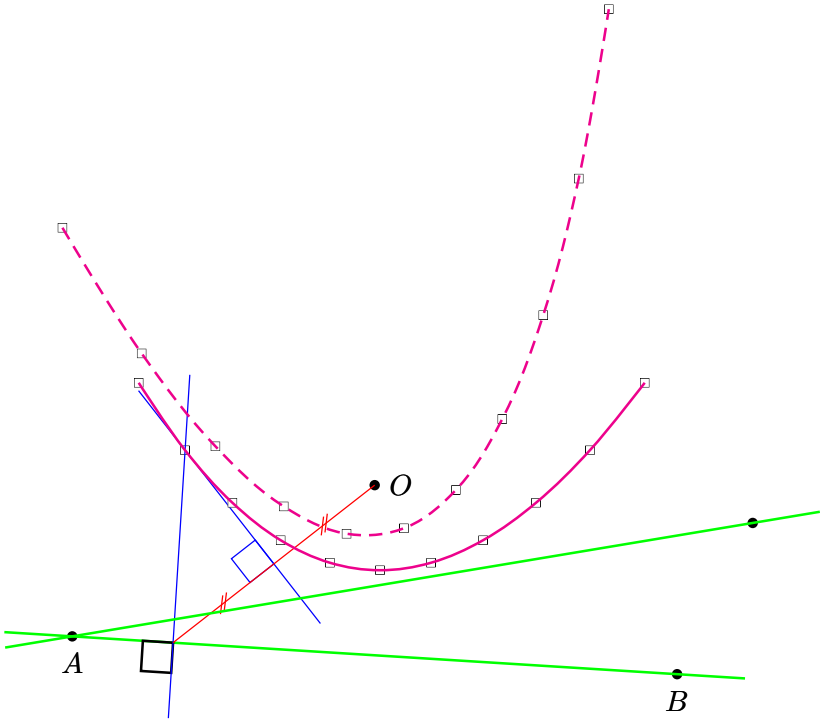

```
\begin{pspicture}(-6,-5)(11,15)
\psset{PointSymbol=none}
\pstTriangle[linewidth=2\pslinewidth,linecolor=red](4,1){A}(0,3){B}(5,5){C}
\psset{linecolor=blue}
```

```
\pstBissectBAC[PointSymbol=none,PointName=none]{C}{A}{B}{AB}
\pstBissectBAC[PointSymbol=none,PointName=none]{A}{B}{C}{BB}
\pstBissectBAC[PointSymbol=none,PointName=none]{B}{C}{A}{CB}
\pstInterLL{A}{AB}{B}{BB}{I}
\psset{linecolor=magenta, linestyle=dashed}
\pstProjection{A}{B}{I}[I_C]
\pstLineAB{I}{I_C}\pstRightAngle[linestyle=solid]{A}{I_C}{I}
\pstProjection{A}{C}{I}[I_B]
\pstLineAB{I}{I_B}\pstRightAngle[linestyle=solid]{C}{I_B}{I}
\pstProjection[PosAngle=80]{C}{B}{I}[I_A]
\pstLineAB{I}{IA}\pstRightAngle[linestyle=solid]{B}{I_A}{I}
\pstCircleOA[linecolor=yellow, linestyle=solid]{I}{I_A}
\psset{linecolor=magenta, linestyle=none}
\pstOutBissectBAC[PointSymbol=none,PointName=none]{C}{A}{B}{AOB}
\pstOutBissectBAC[PointSymbol=none,PointName=none]{A}{B}{C}{BOB}
\pstOutBissectBAC[PointSymbol=none,PointName=none]{B}{C}{A}{COB}
\pstInterLL[PosAngle=-90]{A}{AOB}{B}{BOB}{I_1}
\pstInterLL{A}{AOB}{C}{COB}{I_2}
\pstInterLL[PosAngle=90]{C}{COB}{B}{BOB}{I_3}
\psset{linecolor=magenta, linestyle=dashed}
\pstProjection[PointName=I_{1C}]{A}{B}{I_1}[I1C]
\pstLineAB{I_1}{I1C}\pstRightAngle[linestyle=solid]{I_1}{I1C}{A}
\pstProjection[PointName=I_{1B}]{A}{C}{I_1}[I1B]
\pstLineAB{I_1}{I1B}\pstRightAngle[linestyle=solid]{A}{I1B}{I_1}
\pstProjection[PointName=I_{1A}]{C}{B}{I_1}[I1A]
\pstLineAB{I_1}{I1A}\pstRightAngle[linestyle=solid]{I_1}{I1A}{C}
\pstProjection[PointName=I_{2B}]{A}{C}{I_2}[I2B]
\pstLineAB{I_2}{I2B}\pstRightAngle[linestyle=solid]{A}{I2B}{I_2}
\pstProjection[PointName=I_{2C}]{A}{B}{I_2}[I2C]
\pstLineAB{I_2}{I2C}\pstRightAngle[linestyle=solid]{I_2}{I2C}{A}
\pstProjection[PointName=I_{2A}]{B}{C}{I_2}[I2A]
\pstLineAB{I_2}{I2A}\pstRightAngle[linestyle=solid]{C}{I2A}{I_2}
\pstProjection[PointName=I_{3A}]{C}{B}{I_3}[I3A]
\pstLineAB{I_3}{I3A}\pstRightAngle[linestyle=solid]{C}{I3A}{I_3}
\pstProjection[PointName=I_{3C}]{A}{B}{I_3}[I3C]
\pstLineAB{I_3}{I3C}\pstRightAngle[linestyle=solid]{A}{I3C}{I_3}
\pstProjection[PointName=I_{3B}]{C}{A}{I_3}[I3B]
\pstLineAB{I_3}{I3B}\pstRightAngle[linestyle=solid]{I_3}{I3B}{A}
\psset{linecolor=yellow, linestyle=solid}
\pstCircleOA{I_1}{I1C} \pstCircleOA{I_2}{I2B} \pstCircleOA{I_3}{I3A}
\psset{linecolor=red, linestyle=solid, nodesepA=-1, nodesepB=-1}
\pstLineAB{I1B}{I3B}\pstLineAB{I1A}{I2A}\pstLineAB{I2C}{I3C}
\end{pspicture}
```

B. Some locus points

B.1. Parabola

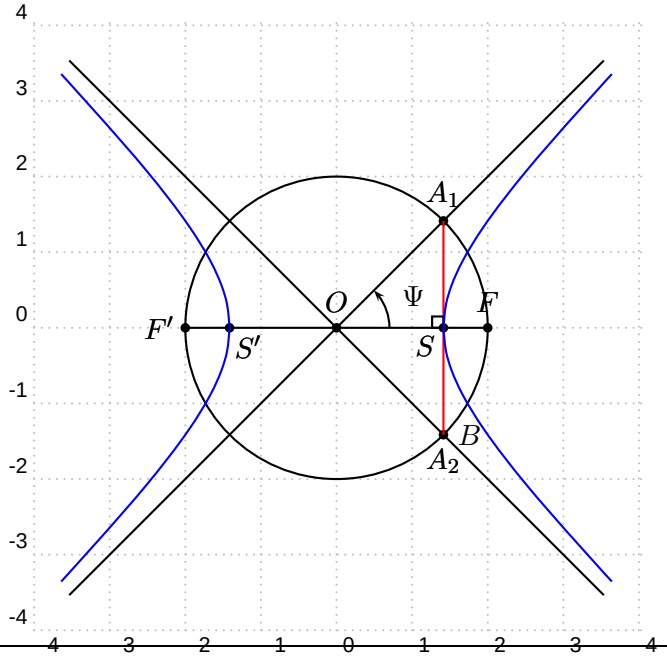
The parabola is the set of points which are at the same distance between a point and a line.




```
\def\NbPt{11}
\begin{pspicture}(-0.5,0)(11,10)
\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}}
\Parabole[2]\pstGenericCurve[linecolor=magenta]{P_}{1}{\NbPt}
\pstGeonode[PointSymbol=*, PosAngle=-90](10,3.5){B}
\Parabole\pstGenericCurve[linecolor=magenta,linestyle=dashed]{P_}{1}{\NbPt}
\end{pspicture}
```

B.2. Hyperbola

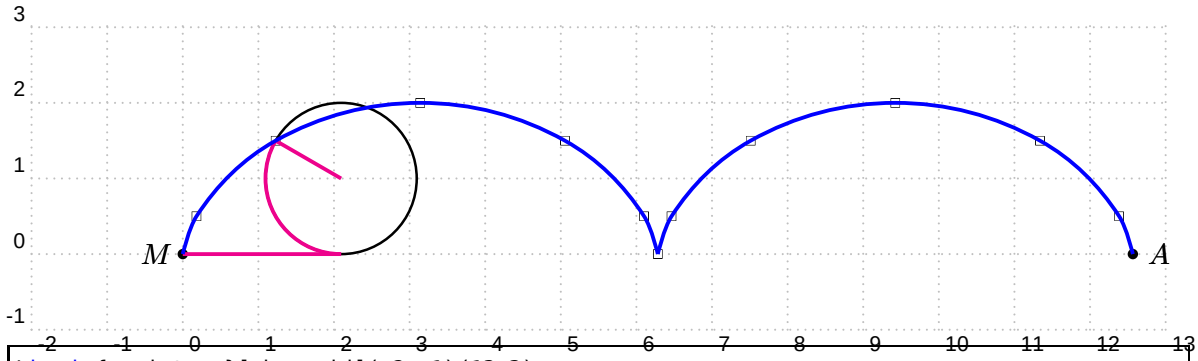
The hyperbola is the set of points whose difference between their distance of two points (the focus) is constant.




```
\begin{pspicture}[showgrid](-4,-4)(4,4)
\newcommand\Sommet{1.4142135623730951 } \newcounter{i} \setcounter{i}{1}
\newcommand\PosFoyer{2 } \newcommand\HypAngle{0}
\newcounter{CoefDiv}\setcounter{CoefDiv}{20}
\newcounter{Inc}\setcounter{Inc}{2} \newcounter{n}\setcounter{n}{2}
\newcommand\Ri{ \PosFoyer \Sommet sub \arabic{i}\space\arabic{CoefDiv}\space div add }
\newcommand\Rii{\Ri \Sommet 2 mul add }
\pstGeonode[PosAngle=90]{O}(\PosFoyer;\HypAngle){F}
\pstSymO[PosAngle=180]{O}{F}\pstLineAB{F}{F'} \pstCircleOA{O}{F}
\pstGeonode[PosAngle=-135](\Sommet;\HypAngle){S}
\pstGeonode[PosAngle=-45](-\Sommet;\HypAngle){S'}
\pstRotation[RotAngle=90, PointSymbol=none]{S}{O}[B]
\pstInterLC[PosAngleA=90, PosAngleB=-90]{S}{B}{O}{F}{A_1}{A_2}
\pstLineAB[nodesepA=-3,nodesepB=-5]{A_1}{O}\pstLineAB[nodesepA=-3,nodesepB=-5]{A_2}{O}
\pstMarkAngle[LabelSep=.8,MarkAngleRadius=.7,arrows=->,LabelSep=1.1]{F}{O}{A_1}{$\Psi$}
\ncline[linecolor=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}}
  \stepcounter{n}\addtocounter{i}{\value{Inc}}
  \addtocounter{Inc}{\value{Inc}}}%% fin de whiledo
\psset{linecolor=blue}
\pstGenericCurve[GenCurvFirst=S]{M}{2}{7}
\pstGenericCurve[GenCurvFirst=S]{P}{2}{7}
\pstGenericCurve[GenCurvFirst=S']{M'}{2}{7}
\pstGenericCurve[GenCurvFirst=S']{P'}{2}{7}
\end{pspicture}
```

B.3. Cycloid

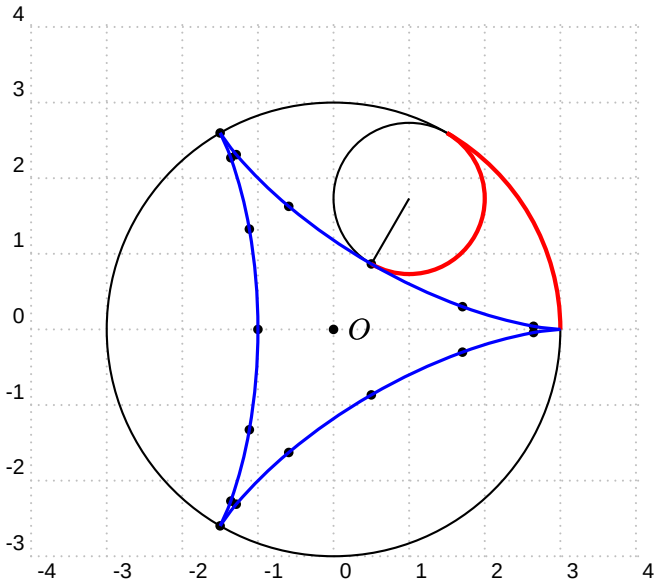
The wheel rolls from M to A. The circle points are on a cycloid.



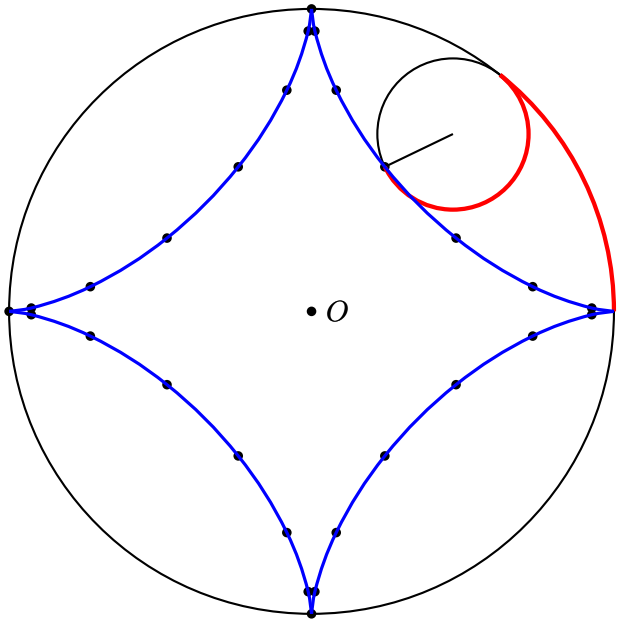

```
\begin{pspicture}[showgrid](-2,-1)(13,3)
\providecommand\NbPt{11}
\psset{linewidth=1.2\pslinewidth}
\pstGeonode[PointSymbol={*,none}, PointName={default,none}, PosAngle=180]{M}(0,1){O}
\pstGeonode(12.5663706144,0){A}
\pstTranslation[PointSymbol=none, PointName=none]{M}{A}{O}[B]
\multido{\nA=1+1}{\NbPt}{%
  \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]
  \pstCurvAbsNode[PointSymbol=square, PointName=none,CurvAbsNeg=true]%
    {O\nA}{P\nA}{M\nA}{\pstDistAB{O}{O\nA}}
  \ifnum\nA=2 \bgroup
    \pstCircleOA{O\nA}{M\nA}
    \psset{linecolor=magenta, linewidth=1.5\pslinewidth}
    \pstArcnOAB{O\nA}{P\nA}{M\nA}
    \ncline{O\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}
\end{pspicture}
```

B.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}
  \def\Anglen{\n\space 360 \NbPt\space 1 add div mul}
  \psset{PointSymbol=none,PointName=none}
  \pstGeonode[PointSymbol={*,none},PointName={default,none}, PosAngle=0]{O}(\R;0){P}
  \pstCircleOA{O}{P}
  \pstHomO[HomCoef=\petitR\space\R\space div]{P}{O}[M]
  \multido{\n=1+1}{\NbPt}{%
    \pstRotation[RotAngle=\Anglen]{O}{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}\ncline{M\n}{N\n}%
      {\psset{linecolor=red, linewidth=2\pslinewidth}
      \pstArcOAB{M\n}{N\n}{N}\pstArcOAB{O}{P}{N}}
    \fi}}%fin multido-newcommand
\begin{pspicture}[showgrid](-3.5,-3.4)(3.5,4)
\HypoCyclo[3]{3}{1}{17}
\psset{linecolor=blue,linewidth=1.5\pslinewidth}
\pstGenericCurve[GenCurvFirst=P]{N}{1}{6}
\pstGenericCurve{N}{6}{12}
\pstGenericCurve[GenCurvLast=P]{N}{12}{17}
\end{pspicture}
```




```
\newcommand\HypoCyclo[4][100]{%
  \def\R{#2}\def\petitR{#3}\def\NbPt{#4}
  \def\Anglen{\n\space 360 \NbPt\space 1 add div mul}
  \psset{PointSymbol=none,PointName=none}
  \pstGeonode[PointSymbol={*,none},PointName={default,none}, PosAngle=0]{O}(\R;0){P}
  \pstCircleOA{O}{P}
  \pstHomO[HomCoef=\petitR\space\R\space div]{P}{O}[M]
  \multido{\n=1+1}{\NbPt}{%
    \pstRotation[RotAngle=\Anglen]{O}{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}\ncline{M\n}{N\n}%
      {\psset{linecolor=red, linewidth=2\pslinewidth}
      \pstArcOAB{M\n}{N\n}{N}\pstArcOAB{O}{P}{N}}
    \fi}}%fin multido-newcommand
\begin{pspicture}(-4.5,-4)(4.5,4.5)
\HypoCyclo[4]{4}{1}{27}
\psset{linecolor=blue, linewidth=1.5\pslinewidth}
\pstGenericCurve[GenCurvFirst=P]{N}{1}{7}
\pstGenericCurve{N}{7}{14}\pstGenericCurve{N}{14}{21}
\pstGenericCurve[GenCurvLast=P]{N}{21}{27}
\end{pspicture}
```

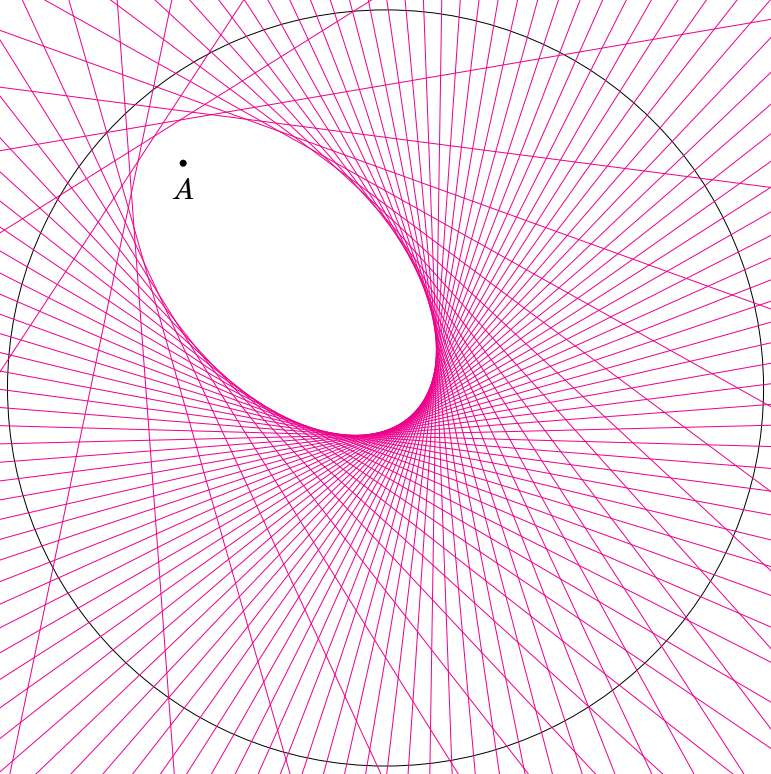
C. Lines and circles envelope

C.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.

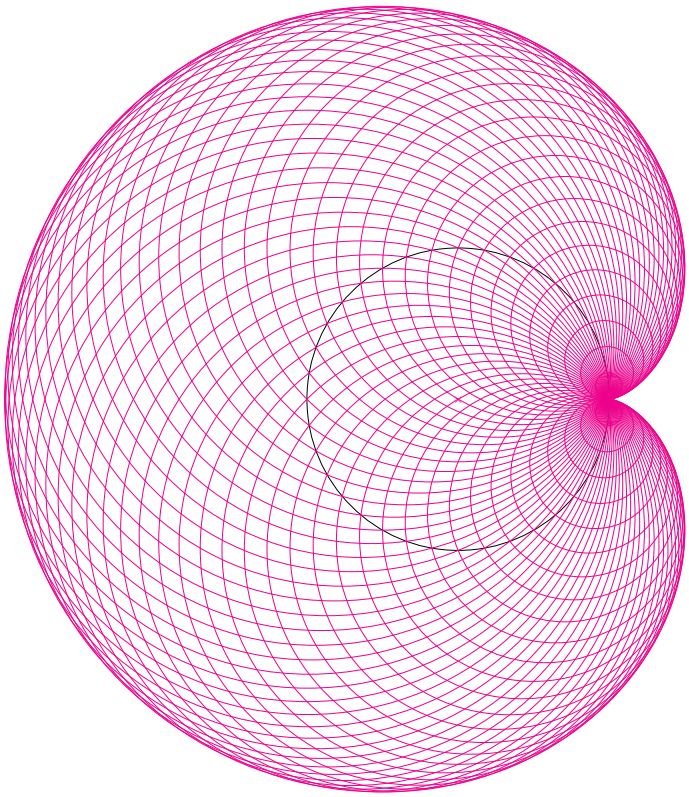
(figure of O. Reboux).




```
\begin{pspicture}(-6,-6)(6,6)
\psset{linewidth=0.4\pslinewidth,PointSymbol=none, PointName=none}
\pstGeonode[PosAngle=-90, PointSymbol={none,*,none}, PointName={none,default,none}]
  {O}(4;132){A}(5,0){O'}
\pstCircleOA{O}{O'}
\multido{\n=5+5}{72}{%
  \pstGeonode(5;\n){M_\n}
  \pstMediatorAB[nodesep=-15,linecolor=magenta]
    {A}{M_\n}{I}{J}}% fin multido
\end{pspicture}
```

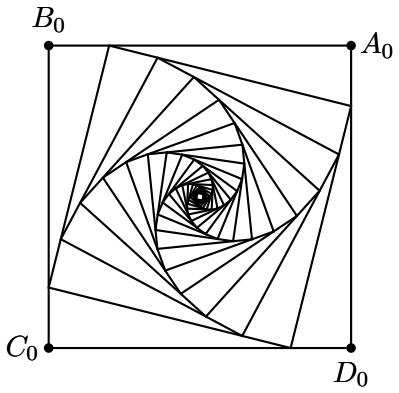
C.2. Cardioid

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



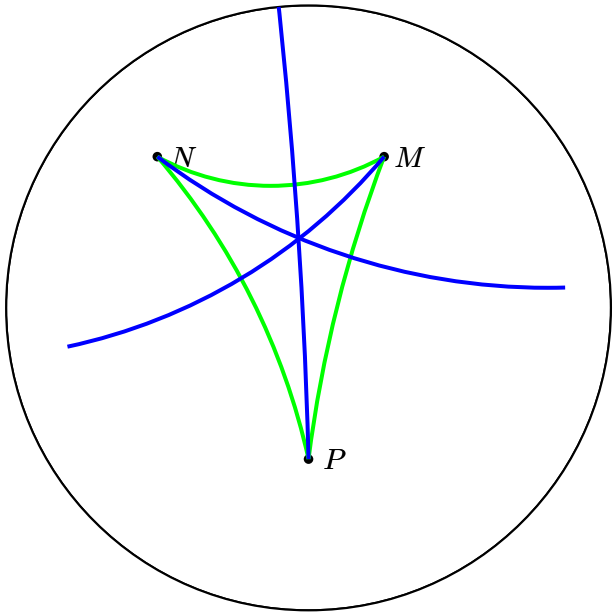
```
\begin{pspicture}(-6,-6)(3,5)
\psset{linewidth=0.4\pslinewidth,PointSymbol=x,nodesep=0,linecolor=magenta}
\pstGeonode[PointName=none]{O}(2,0){O'}
\pstCircleOA[linecolor=black]{O}{O'}
\multido{\n=5+5}{72}{%
  \pstGeonode[PointSymbol=none, PointName=none](2;\n){M_\n}
  \pstCircleOA{M_\n}{O'}}
  \end{pspicture}
```

D. Homotethy and fractals




```
\begin{pspicture}(-2.8,-3)(2.8,3)
\pstGeonode[PosAngle={0,90}](2,2){A_0}(-2,2){B_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}{%
  \pstHomO[PosAngle=0]{B_\i}{A_\i}[A_\n]
  \pstHomO[PosAngle=90]{C_\i}{B_\i}[B_\n]
  \pstHomO[PosAngle=180]{D_\i}{C_\i}[C_\n]
  \pstHomO[PosAngle=270]{A_\i}{D_\i}[D_\n]
  \pspolygon(A_\n)(B_\n)(C_\n)(D_\n)}% fin multido
\end{pspicture}
```

E. hyperbolic geometry: a triangle and its altitudes




```
\begin{pspicture}(-5,-5)(5,5)
\psclip{\pscircle(0,0){4}}
  \pstGeonode(1, 2){M}\pstGeonode(-2,2){N}\pstGeonode(0,-2){P}
  \psset{DrawCirABC=false, PointSymbol=none, PointName=none}%
  \pstGeonode(0,0){O}\pstGeonode(4,0){A}\pstCircleOA{O}{A}
  \pstHomO[HomCoef=\pstDistAB{O}{A} 2 mul \pstDistAB{O}{M} sub
    \pstDistAB{O}{M} div]{O}{M}[M']%
  \pstHomO[HomCoef=\pstDistAB{O}{A} 2 mul \pstDistAB{O}{P} sub
    \pstDistAB{O}{P} div]{O}{P}[P']%
  \pstHomO[HomCoef=\pstDistAB{O}{A} 2 mul \pstDistAB{O}{N} sub
    \pstDistAB{O}{N} div]{O}{N}[N']%
  \psset{linecolor=green, linewidth=1.5pt}%
  \pstCircleABC{M}{N}{M'}{OmegaMN}\pstArcOAB{OmegaMN}{N}{M}
  \pstCircleABC{M}{P}{M'}{OmegaMP}\pstArcOAB{OmegaMP}{M}{P}
  \pstCircleABC{N}{P}{P'}{OmegaNP}\pstArcOAB{OmegaNP}{P}{N}
  \psset{linecolor=blue}
  \pstHomO[HomCoef=\pstDistAB{OmegaNP}{N} 2 mul \pstDistAB{OmegaNP}{M} sub %% M
    \pstDistAB{OmegaNP}{M} div]{OmegaNP}{M}[MH']
  \pstCircleABC{M}{M'}{MH'}{OmegaMH}\pstArcOAB{OmegaMH}{MH'}{M} %% N
  \pstHomO[HomCoef=\pstDistAB{OmegaMP}{M} 2 mul \pstDistAB{OmegaMP}{N} sub
    \pstDistAB{OmegaMP}{N} div]{OmegaMP}{N}[NH']
  \pstCircleABC{N}{N'}{NH'}{OmegaNH}\pstArcOAB{OmegaNH}{N}{NH'} %% P
  \pstHomO[HomCoef=\pstDistAB{OmegaMN}{M} 2 mul \pstDistAB{OmegaMN}{P} sub
    \pstDistAB{OmegaMN}{P} div]{OmegaMN}{P}[PH']
  \pstCircleABC{P}{P'}{PH'}{OmegaPH}\pstArcOAB{OmegaPH}{P}{PH'}
\endpsclip
\end{pspicture}
```

F. List of all optional arguments for pst-eucl

Key	Type	Default
PointSymbol	ordinary	default
PointSymbolA	ordinary	undef
PointSymbolB	ordinary	undef
PointSymbolC	ordinary	undef
PointName	ordinary	default
PointNameA	ordinary	undef
PointNameB	ordinary	undef
PointNameC	ordinary	undef
PtNameMath	ordinary	false
SegmentSymbol	ordinary	MarkHashh
SegmentSymbolA	ordinary	MarkHashh
SegmentSymbolB	ordinary	MarkHashh
SegmentSymbolC	ordinary	MarkHashh
Mark	ordinary	undef
mark	ordinary	undef
MarkAngle	ordinary	undef
MarkHashLength	ordinary	1.25mm
MarkHashSep	ordinary	0.625mm
PointNameSep	ordinary	[none]
PosAngle	ordinary	[none]
PosAngleA	ordinary	undef
PosAngleB	ordinary	undef
PosAngleC	ordinary	undef
RightAngleSize	ordinary	4
RightAngleType	ordinary	default
MarkAngleRadius	ordinary	0.4
LabelAngleOffset	ordinary	0
LabelSep	ordinary	1
LabelRefPt	ordinary	c
CurveType	ordinary	none
HomCoef	ordinary	0.5
RotAngle	ordinary	60
TransformLabel	ordinary	none
Central@Sym	ordinary	false
DrawCirABC	ordinary	true
CodeFig	boolean	true
CodeFigA	ordinary	undef
CodeFigB	ordinary	undef
CodeFigColor	ordinary	cyan
CodeFigStyle	ordinary	dashed
CodeFigAarc	ordinary	true

Continued on next page

Continued from previous page

Key	Type	Default
CodeFigBarc	ordinary	true
Radius	ordinary	none
RadiusA	ordinary	undef
RadiusB	ordinary	undef
Diameter	ordinary	none
DiameterA	ordinary	undef
DiameterB	ordinary	undef
DistCoef	ordinary	none
AngleCoef	ordinary	none
CurvAbsNeg	ordinary	false
GenCurvFirst	ordinary	none
GenCurvLast	ordinary	none
GenCurvInc	ordinary	1
AngleValue	boolean	false
AngleArc	boolean	true
ShowWedge	boolean	true
ArcColor	ordinary	[none]
ArcLinestyle	ordinary	[none]
ArcLinewidth	ordinary	[none]
WedgeColor	ordinary	[none]
WedgeFillstyle	ordinary	[none]
WedgeOpacity	ordinary	[none]

References

[1] Victor Eijkhout. *TEX by Topic – A TEXnician Reference*. DANTE – lehmanns media, Heidelberg/Berlin, 1 edition, 2014.

[2] Denis Girou. Présentation de PSTricks. *Cahier GUTenberg*, 16:21–70, April 1994.

[3] Michel Goosens, Frank Mittelbach, Sebastian Rahtz, Dennis Roegel, and Herbert Voß. *The LATEX Graphics Companion*. Addison-Wesley Publishing Company, Boston, Mass., second edition, 2007.

[4] Nikolai G. Kollock. *PostScript richtig eingesetzt: vom Konzept zum praktischen Einsatz*. IWT, Vaterstetten, 1989.

[5] Herbert Voß. Die mathematischen Funktionen von Postscript. *Die TEXnische Komödie*, 1/02:40–47, March 2002.

[6] Herbert Voß. *PSTricks – Grafik für TEX und LATEX*. DANTE – Lehmanns, Heidelberg/Hamburg, 6. edition, 2010.

[7] Herbert Voß. *PSTricks – Graphics and PostScript for LATEX*. UIT, Cambridge – UK, 1. edition, 2011.

[8] Herbert Voß. *LaTeX quick reference*. UIT, Cambridge – UK, 1. edition, 2012.

[9] Herbert Voß. *Presentations with LaTeX*. DANTE – Lehmanns Media, Heidelberg/Berlin, 1. edition, 2012.

[10] Timothy Van Zandt. *multido.tex - a loop macro, that supports fixed-point addition*. CTAN:/macros/generic/multido.tex, 1997.

[11] Timothy Van Zandt and Denis Girou. Inside PSTricks. *TUGboat*, 15:239–246, September 1994.

Index

Symbols

*, 5, 6
+, 6
...A, 17
...B, 17

A

AngleCoef, 14
arrows, 9
asterisk, 6

C

CodeFig, 12–17, 20
CodeFigA, 20
CodeFigAarc, 20
CodeFigB, 20
CodeFigBarc, 20
CodeFigColor, 12, 13, 15–17
CodeFigStyle, 12, 13, 15–17
CurvAbsNeg, 11
CurveType, 13, 26
CurveType=none, 6

D

default, 5, 8
Diameter, 10, 19
DiameterA, 20
DiameterB, 20
diamond, 6
diamond*, 6
dimen, 5
DistCoef, 10, 14, 19
dotangle, 6
dotscale, 6
DrawCirABC, 16

F

false, 12

G

GenCurvFirst, 12
GenCurvInc, 12
GenCurvLast, 12
german, 8

H

HomCoef, 15

K

Keyvalue
– default, 8
– german, 8
– none, 5
– pstslashh, 7
– suisseromand, 8

Keyword
– *, 6
– +, 6
– ...A, 17
– ...B, 17
– AngleCoef, 14
– arrows, 9
– asterisk, 6
– CodeFig, 12–17, 20
– CodeFigA, 20
– CodeFigAarc, 20
– CodeFigB, 20
– CodeFigBarc, 20
– CodeFigColor, 12, 13, 15–17
– CodeFigStyle, 12, 13, 15–17
– CurvAbsNeg, 11
– CurveType, 13, 26
– CurveType=none, 6
– Diameter, 10, 19
– DiameterA, 20
– DiameterB, 20
– diamond, 6
– diamond*, 6
– dimen, 5
– DistCoef, 10, 14, 19
– dotangle, 6
– dotscale, 6
– DrawCirABC, 16
– false, 12
– GenCurvFirst, 12
– GenCurvInc, 12
– GenCurvLast, 12
– HomCoef, 15

– label, 8
– LabelAngleOffset, 8, 9
– LabelRefPt, 9
– LabelSep, 8
– Mark, 8, 9
– MarkAngle, 7
– MarkAngleRadius, 8, 9
– MarkHashLength, 7
– MarkHashSep, 7
– nodesep, 9
– nodesepA, 9
– nodesepB, 9
– o, 6
– oplus, 6
– otimes, 6
– pentagon, 6
– pentagon*, 6
– PointName, 5, 6, 8, 11, 13–19
– PointNameA, 8, 19
– PointNameB, 8, 19
– PointNameC, 8
– PointNameSep, 5, 6, 8, 11, 13–19
– PointSymbol, 5, 6, 8, 11, 13–19
– PointSymbolA, 8, 19
– PointSymbolB, 8, 19
– PointSymbolC, 8
– PosAngle, 5, 6, 8, 11, 13–19
– PosAngleA, 8, 19
– PosAngleB, 8, 19
– PosAngleC, 8
– PtNameMath, 5, 6, 11, 13–19
– Radius, 10, 19
– radius, 19
– RadiusA, 20
– RadiusB, 20
– RightAngleSize, 8
– RightAngleType, 8
– RotAngle, 14
– SegmentSymbol, 7, 15, 17
– SegmentSymbolA, 16
– SegmentSymbolB, 16
– SegmentSymbolC, 16
– square, 6
– square*, 6
– TransformLabel, 14
– triangle, 6
– triangle*, 6
– true, 12
– x, 6

L

label, 8
LabelAngleOffset, 8, 9
LabelRefPt, 9
LabelSep, 8

M

Macro
– \MarkCros, 7
– \MarkCross, 7
– \MarkHash, 7
– \MarkHashh, 7
– \MarkHashhh, 7
– \ncline, 7
– \psdot, 5
– \psGetAngleABC, 23
– \psGetDistanceAB, 23
– \psplot, 21
– \pstAngleABC, 14
– \pstAngleAOB, 14
– \pstArcnOAB, 11
– \pstArcOAB, 11
– \pstBissectBAC, 18
– \pstCGravABC, 16
– \pstCircleAB, 10
– \pstCircleABC, 16
– \pstCircleOA, 10
– \pstCurvAbsNode, 11
– \pstDistAB, 10
– \pstDistVal, 10
– \pstGenericCurve, 12
– \pstGeonode, 2, 5
– \pstHomO, 15
– \pstInterCC, 20
– \pstInterFC, 22
– \pstInterFF, 21
– \pstInterFL, 22
– \pstInterLC, 19
– \pstInterLL, 18
– \pstLineAB, 9
– \pstMarkAngle, 8

– \pstMarkSegment, 7
– \pstMediatorAB, 17
– \pstMiddleAB, 15
– \pstOIJGeonode, 6
– \pstOrtSym, 13
– \pstOutBissectBAC, 18
– \pstProjection, 15
– \pstRightAngle, 8
– \pstRotation, 14
– \pstslash, 7
– \pstslashh, 7
– \pstslashhh, 7
– \pstSymO, 13
– \pstTranslation, 14
– \pstTriangle, 8
– \SpecialCoor, 5
Mark, 8, 9
MarkAngle, 7
MarkAngleRadius, 8, 9
\MarkCros, 7
\MarkCross, 7
\MarkHash, 7
\MarkHashh, 7
\MarkHashhh, 7
MarkHashLength, 7
MarkHashSep, 7
middle, 5

N

\ncline, 7
nodesep, 9
nodesepA, 9
nodesepB, 9
none, 5

O

o, 6
oplus, 6
otimes, 6

P

Package
– pst-eucl, 2
pentagon, 6
pentagon*, 6
PointName, 5, 6, 8, 11, 13–19
PointNameA, 8, 19
PointNameB, 8, 19
PointNameC, 8
PointNameSep, 5, 6, 8, 11, 13–19
PointSymbol, 5, 6, 8, 11, 13–19
PointSymbolA, 8, 19
PointSymbolB, 8, 19
PointSymbolC, 8
PosAngle, 5, 6, 8, 11, 13–19
PosAngleA, 8, 19
PosAngleB, 8, 19
PosAngleC, 8
\psdot, 5
\psGetAngleABC, 23
\psGetDistanceAB, 23
\psplot, 21
pst-eucl, 2
\pstAngleABC, 14
\pstAngleAOB, 14
\pstArcnOAB, 11
\pstArcOAB, 11
\pstBissectBAC, 18
\pstCGravABC, 16
\pstCircleAB, 10
\pstCircleABC, 16
\pstCircleOA, 10
\pstCurvAbsNode, 11
\pstDistAB, 10
\pstDistVal, 10
\pstGenericCurve, 12
\pstGeonode, 2, 5
\pstHomO, 15
\pstInterCC, 20
\pstInterFC, 22
\pstInterFF, 21
\pstInterFL, 22
\pstInterLC, 19
\pstInterLL, 18
\pstLineAB, 9
\pstMarkAngle, 8
\pstMarkSegment, 7
\pstMediatorAB, 17
\pstMiddleAB, 15
\pstOIJGeonode, 6
\pstOrtSym, 13

\pstOutBissectBAC, 18
\pstProjection, 15
\pstRightAngle, 8
\pstRotation, 14
\pstslash, 7
\pstslashh, 7
pstslashh, 7
\pstslashhh, 7
\pstSymO, 13
\pstTranslation, 14
\pstTriangle, 8
PtNameMath, 5, 6, 11, 13–19

R

Radius, 10, 19
radius, 19
RadiusA, 20
RadiusB, 20
RightAngleSize, 8
RightAngleType, 8
RotAngle, 14

S

SegmentSymbol, 7, 15, 17
SegmentSymbolA, 16
SegmentSymbolB, 16
SegmentSymbolC, 16
\SpecialCoor, 5
square, 6
square*, 6
suisseromand, 8

T

TransformLabel, 14
triangle, 6
triangle*, 6
true, 12

V

Value
– *, 5
– default, 5
– middle, 5

X

x, 6