

Chronique 15

Euclide – D’autres objets

On continue l’étude du package `euclide`.

15.1 Orthogonalité

15.1.1 Projection orthogonale

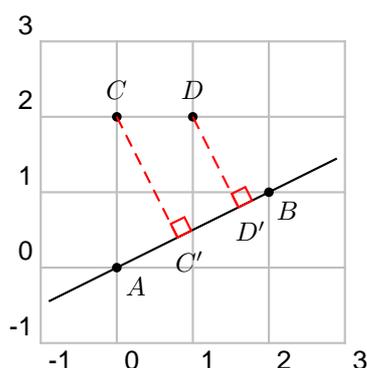
Avec le package `euclide`, on peut déterminer sans problème (et sans calcul!) le projeté orthogonal d’un point sur une droite, et donc tracer la droite perpendiculaire à une droite passant par un point extérieur à cette droite. C’est l’instruction `\pstProjection` qui fait ça parfaitement.

Cette commande nécessite trois paramètres, plus un optionnel que l’on met en fin de commande (ce qui est rare!). Les deux premiers paramètres sont deux points de la droite sur laquelle on va projeter. Le troisième paramètre est la liste des points dont il faut chercher les projetés.

Si on écrit `\pstProjection{A}{B}{C,D}`, on va chercher les projetés orthogonaux des points C et D sur la droite (AB) ; les projetés seront alors automatiquement nommés C' et D' .

Le paramètre optionnel de la commande sert à renommer les projetés si C' et D' ne nous conviennent pas : si on veut appeler E et F respectivement les projetés de C et D sur la droite (AB) , on écrira `\pstProjection{A}{B}{C,D}[E,F]`. Le paramètre optionnel s’écrit avec des crochets.

Voyons un petit exemple :



```
\psset{unit=1cm}
\def\xmin{-1} \def\xmax{3}
\def\ymin{-1} \def\ymax{3}
\begin{pspicture}(\xmin,\ymin)(\xmax,\ymax)
  \psgrid[subgriddiv=1,gridcolor=lightgray]
  \pstGeonode[PosAngle={-45,-45,90,90}]
    (0,0){A}(2,1){B}(0,2){C}(1,2){D}
  \pstLineAB[nodesep=-1]{A}{B}
  \pstProjection[CodeFig=true,CodeFigColor=red,
    PointSymbol=none,RightAngleSize=0.2]{A}{B}{C,D}
\end{pspicture}
```

Après avoir défini les points A , B , C et D , on trace la droite (AB) .

Regardons de plus près les options de l’instruction `\pstProjection`.

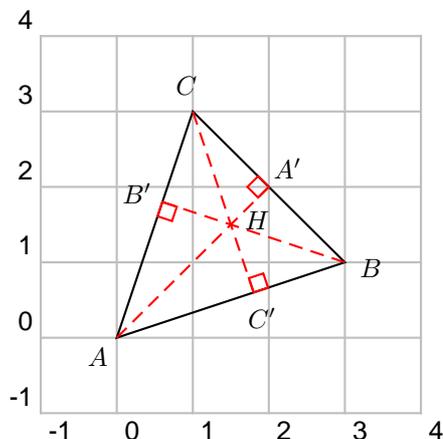
Le booléen `CodFig=true` trace automatiquement le segment en mode traits entre le point et son projeté, et marque l’angle droit. On comprend bien ainsi en quoi consiste le projeté orthogonal!

On comprend vite que `CodeFigColor` spécifie la couleur du codage de la figure.

Quant à `RightAngleSize` et `PointSymbol`, ils ont été vus dans une précédente chronique.

15.1.2 Hauteurs et orthocentre

On peut maintenant tracer les trois hauteurs d'un triangle ainsi que son orthocentre :



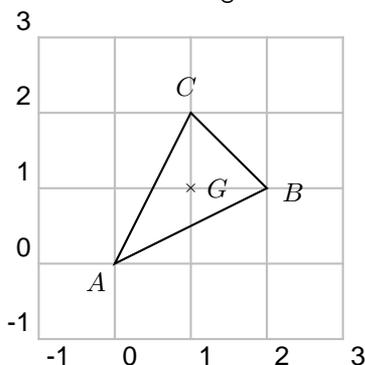
```
\psset{unit=1cm,PointSymbol=none}
\def\xmin{-1} \def\xmax{4}
\def\ymin{-1} \def\ymax{4}
\begin{pspicture}(\xmin,\ymin)(\xmax,\ymax)
\psgrid[subgriddiv=1,gridcolor=lightgray]
\pstTriangle(0,0){A}(3,1){B}(1,3){C}
\psset{CodeFig=true,CodeFigColor=red,
RightAngleSize=0.2}
\pstProjection{A}{B}{C}% définit C'
\pstProjection{B}{C}{A}% définit A'
\pstProjection{C}{A}{B}% définit B'
\pstInterLL[PosAngle=10]{A}{A'}{B}{B'}{H}
\end{pspicture}
```

15.2 Centre de gravité

On a déjà vu comment déterminer le milieu d'un segment avec `\pstMiddleAB` ; on peut ainsi tracer les médianes d'un triangle et déterminer son centre de gravité par intersection de deux médianes. Mais il y a une instruction qui fait ça directement : c'est `\pstCGravABC`.

Quatre paramètres pour cette instruction : les trois premiers sont les trois sommets du triangle, et le quatrième est le nom du centre de gravité.

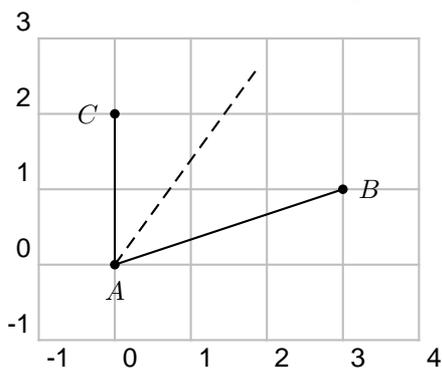
Le booléen `CodeFig` n'a aucun effet dans cette construction.



```
\psset{unit=1cm}
\def\xmin{-1} \def\xmax{3}
\def\ymin{-1} \def\ymax{3}
\begin{pspicture}(\xmin,\ymin)(\xmax,\ymax)
\psgrid[subgriddiv=1,gridcolor=lightgray]
\pstTriangle[PointSymbol=none]
(0,0){A}(2,1){B}(1,2){C}
\pstCGravABC[PointSymbol=x]{A}{B}{C}{G}
\end{pspicture}
```

15.3 Bissectrices

Tracer la bissectrice d'un angle est un jeu d'enfant avec l'instruction `\pstBissectBAC` :

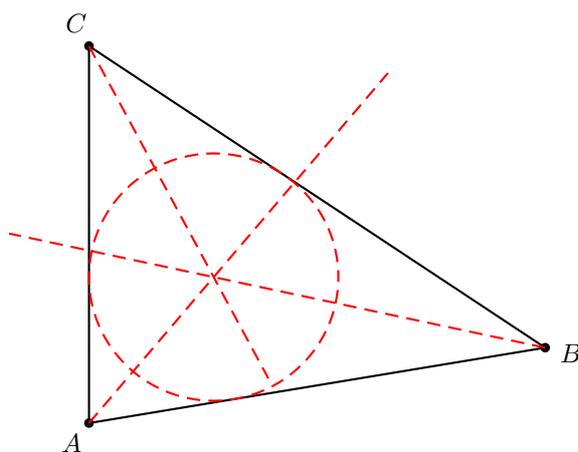


```
\psset{unit=1cm}
\def\xmin{-1} \def\xmax{4}
\def\ymin{-1} \def\ymax{4}
\begin{pspicture}(\xmin,\ymin)(\xmax,\ymax)
\psgrid[subgriddiv=1,gridcolor=lightgray]
\pstGeonode[CurveType=polyline,PosAngle={0,-90,180}]
(3,1){B}(0,0){A}(0,2){C}
\pstBissectBAC[PointSymbol=none,PointName=none,
linestyle=dashed]{B}{A}{C}{A'}
\end{pspicture}
```

L'instruction `\pstBissectBAC` a besoin de quatre paramètres : les trois sommets de l'angle dont on veut la bissectrice, un quatrième point qui est obtenu par rotation (et qui permet de tracer la bissectrice). On peut ne pas faire afficher ce quatrième point comme dans l'exemple.

Les angles sont pris dans le sens trigonométrique donc si l'on entre `\pstBissectBAC{B}{A}{C}{A'}`, on n'aura pas le même dessin qu'avec `\pstBissectBAC{C}{A}{B}{A'}`. À essayer !

Vous avez donc en main tous les outils pour tracer les trois bissectrices des angles d'un triangle, et le cercle inscrit dans ce triangle :



Enfin, pour tracer les cercles exinscrits au triangle, on utilisera l'instruction `\pstOutBissectBAC` qui trace une bissectrice extérieure au triangle.

15.4 Cercle circonscrit

15.4.1 Médiatrice d'un segment

On peut tracer la médiatrice d'un segment au moyen de l'instruction `\pstMediatorAB`.

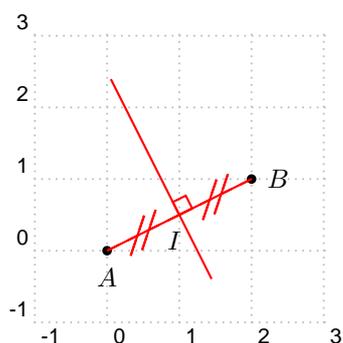
Voyons comment cette commande trace la médiatrice d'un segment $[AB]$: elle détermine le milieu du segment (dont il faut donner le nom, par exemple I) et un deuxième point (dont il faut également donner le nom, par exemple M_I) qui est l'image du deuxième point du segment (ici B) par la rotation de centre I et d'angle 90° . Enfin cette macro-commande trace le segment $[IM_I]$.

L'instruction `\pstMediatorAB` aura donc besoin de quatre paramètres : les extrémités du segment dont on veut la médiatrice, le nom du milieu du segment, et le nom du point de construction de la médiatrice. Cette instruction crée donc deux points I et M_I dont les attributs sont contrôlés respectivement par `PointNameA` et `PointSymbolA`, `PointNameB` et `PointSymbolB`. Comme le point M_I est un point de construction, il n'a pas vocation à apparaître dans la figure ; on écrira donc : `PointNameB=none` et `PointSymbolB=none` pour avoir une figure « propre ».

Le booléen `CodeFig` peut être activé dans cette construction de médiatrice.

Au passage, on remarquera que la grille est tracée au moyen du booléen `showgrid` qui est passé en option dans `\pspicture` (et qui veut dire `showgrid=true`).

Savez-vous comment on dit « médiatrice » en anglais ? C'est « perpendicular bisector ». Et quand on voit les similitudes entre les constructions (à la règle et au compas) d'une bissectrice d'un angle et d'une médiatrice d'un segment, on comprend pleinement le bien-fondé de cette appellation !

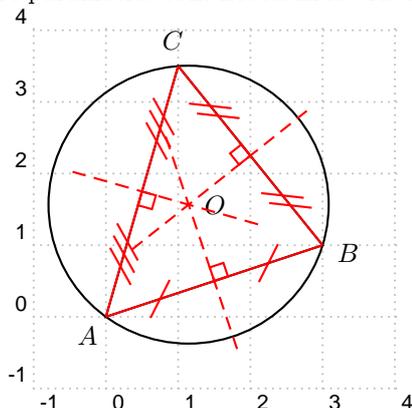


```
\psset{unit=1cm,CodeFig=true,CodeFigColor=red}
\def\xmin{-1} \def\xmax{3}
\def\ymin{-1} \def\ymax{3}
\begin{pspicture}(\xmin,\ymin)(\xmax,\ymax)
\psgrid[subgriddiv=1,gridcolor=lightgray]
\pstGeonode[PosAngle={-90,0}](0,0){A}(2,1){B}
\psset{PointNameB=none,PointSymbolB=none,
PointSymbolA=none,PosAngle=-100}
\pstMediatorAB[nodesep=-1,linecolor=red,
RightAngleSize=0.2]{A}{B}{I}{M_I}
\end{pspicture}
```

15.4.2 Cercle circonscrit

En utilisant deux des trois médiatrices des côtés d'un triangle, on peut maintenant déterminer le centre du cercle circonscrit au triangle, et tracer ce cercle. On va utiliser l'instruction `\pstCircleABC`.

Quatre paramètres sont nécessaires : les trois sommets du triangle et le centre du cercle circonscrit.



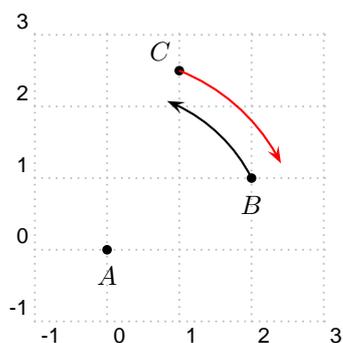
```
\psset{unit=1cm,PointSymbol=none}
\def\xmin{-1} \def\xmax{4}
\def\ymin{-1} \def\ymax{4}
\begin{pspicture}[showgrid]
(\xmin,\ymin)(\xmax,\ymax)
\pstTriangle(0,0){A}(3,1){B}(1,3.5){C}
\psset{CodeFig=true,CodeFigColor=red,
RightAngleSize=0.2}
\pstCircleABC{A}{B}{C}{O}
\end{pspicture}
\end{pspicture}
```

Pour tracer les trois médiatrices des côtés sans tracer le cercle, on utilisera le booléen `DrawCirABC` :

```
\pstCircleABC[DrawCirABC=false]{A}{B}{C}{O}
```

15.5 Arc de cercle

On peut, avec `PsTricks` tracer un arc, avec ou sans flèche, mais il faut donner la mesure de l'angle en degrés. Avec `euclide`, on peut également tracer un arc en utilisant trois points ; l'instruction `\pstArcOAB` nécessite trois points $\{A\}$, $\{B\}$, $\{C\}$, et trace l'arc de cercle correspondant à l'angle $(\overrightarrow{AB}, \overrightarrow{AC})$ sur le cercle de centre A passant par B :



```
\psset{unit=1cm,arrowsize=3pt 2}
\def\xmin{-1} \def\xmax{3}
\def\ymin{-1} \def\ymax{3}
\begin{pspicture}[showgrid](\xmin,\ymin)(\xmax,\ymax)
\pstGeonode[PosAngle={-90,-90,135}]
(0,0){A}(2,1){B}(1,2.5){C}
\pstArcOAB[arrows=->]{A}{B}{C}
\pstArcnOAB[arrows=->,linecolor=red]{A}{C}{B}
\end{pspicture}
```

Pour avoir un arc dans l'autre sens, on utilise `\pstArcnOAB`.