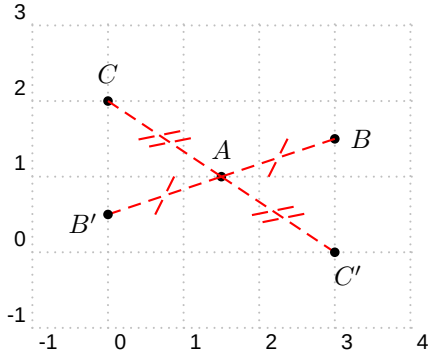
Chronique 16

Euclide – Transformations

Dernier volet de la présentation de l'excellent package `euclide`, consacré aux transformations.

16.1 Symétrie centrale

Un centre A, deux points B et C, et leurs images B' et C' par la symétrie de centre A, composent la figure ci-dessous. L'instruction `\pstSymO` a besoin de deux paramètres : le centre de la symétrie, la liste des points dont il faut chercher les symétriques. Il faut avoir défini ces points préalablement.




```
\psset{unit=1cm,CodeFig=true,CodeFigColor=red}
\def\xmin{-1}   \def\xmax{4}
\def\ymin{-1}   \def\ymax{3}
\begin{pspicture}[showgrid]
      (\xmin,\ymin)(\xmax,\ymax)
  \pstGeonode[PosAngle={90,0,90}]
           (1.5,1){A}(3,1.5){B}(0,2){C}
  \pstSymO[PosAngle={200,-60}]{A}{B,C}
\end{pspicture}
```

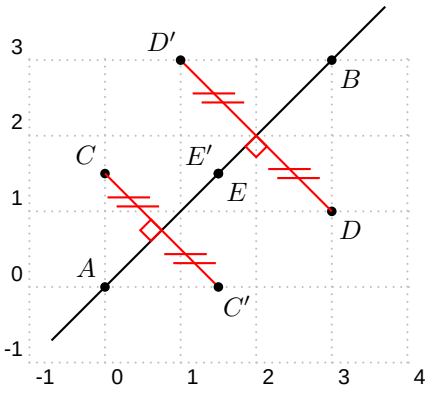
Tout comme pour la projection orthogonale (voir page 77), les images de B et de C sont automatiquement appelées B' et C' ; si on veut les appeler respectivement D et E on écrira :

```
\pstSymO[PosAngle={200,-60}]{A}{B,C}[D,E]
```

En activant le booléen `CodeFig`, le codage de la figure est rajouté ; la couleur de ce codage est gérée par `CodeFigColor`.

16.2 Symétrie orthogonale

On définit une symétrie orthogonale par rapport à une droite très simplement en utilisant l'instruction `\pstOrtSym`. Cette instruction a besoin de trois paramètres : les deux premiers désignent la droite qui sert d'axe de symétrie, et le troisième est la liste des points dont on veut les images.
Là aussi, on peut activer le codage de la figure par le booléen `CodeFig`. Comme dans les autres exemples, le codage est en mode `tirets` (`dashed`) ; si on veut changer ce style, on utilisera la variable `CodeFigStyle` comme dans l'exemple ci-dessous.
Si on cherche l'image d'un point situé sur l'axe (AB) (comme le point E de la figure), il vaut mieux désactiver le booléen `CodeFig` en le mettant à `false`.
Enfin comme les angles droits sont automatiquement codés quand `CodeFig` est à `true`, on réduira la taille du codage au moyen de `RightAngleSize`.



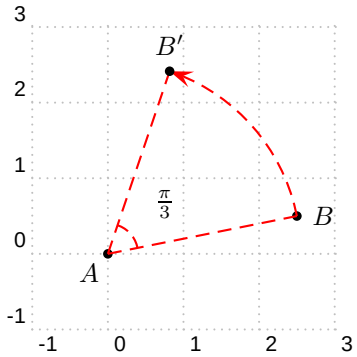

```
\psset{unit=1cm,CodeFig=true,CodeFigColor=red,
       CodeFigStyle=solid}
\def\xmin{-1}   \def\xmax{4}
\def\ymin{-1}   \def\ymax{3}
\begin{pspicture}[showgrid](\xmin,\ymin)(\xmax,\ymax)
  \pstGeonode[PosAngle={135,-45,135,-45}]
       (0,0){A}(3,3){B}(0,1.5){C}(3,1){D}(1.5,1.5){E}
  \pstLineAB[nodesep=-1]{A}{B}
  \pstOrtSym[RightAngleSize=0.2,PosAngle={-45,135}]
          {A}{B}{C,D}
  \pstOrtSym[CodeFig=false,PosAngle=135]{A}{B}{E}
\end{pspicture}
```

16.3 Rotation

Pas de difficulté non plus avec la rotation.
L'angle de la rotation est défini par `RotAngle`.

16.3.1 Angle défini par une valeur

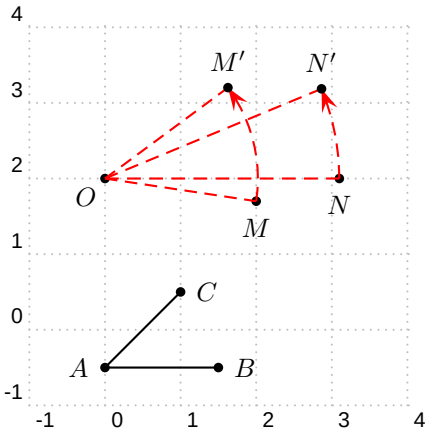
On donne une valeur (en degrés) à l'angle de la rotation : `RotAngle=60`. On passe cette affectation dans `\psset` ou en option dans l'instruction `\pstRotation`. Une fois l'angle défini, cette instruction a besoin de deux paramètres : le centre de la rotation, et la liste des points dont on veut les images.
Le booléen `CodeFig` permet le codage de la figure en traçant un arc de cercle fléché du point vers son image. On modifie la taille des flèches au moyen de la variable `arrowscale`.
On peut faire afficher la mesure de l'angle de la rotation grâce à la variable `TransformLabel`.




```
\psset{unit=1cm,CodeFig=true,CodeFigColor=red}
\def\xmin{-1}   \def\xmax{3}
\def\ymin{-1}   \def\ymax{3}
\begin{pspicture}[showgrid](\xmin,\ymin)(\xmax,\ymax)
  \pstGeonode[PosAngle={-135,0}](0,0){A}(2.5,0.5){B}
  \pstRotation[RotAngle=60,arrowscale=2,
      PosAngle=90,TransformLabel=\frac{\pi}{3}]{A}{B}
\end{pspicture}
```

16.3.2 Angle défini par trois points

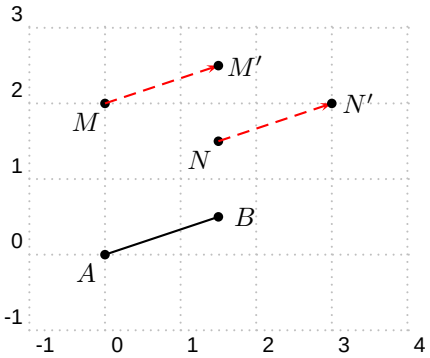
L'angle de la rotation peut être défini au moyen de trois points de la figure ; c'est alors l'instruction `\pstAngleAOB` qu'il faut utiliser.
Si on utilise plusieurs fois l'angle (comme dans l'exemple ci-dessous), on peut le stocker dans une variable par l'instruction `\def`.
On peut également multiplier l'angle précédemment défini par un coefficient `AngleCoef` ; seule contrainte : il faut définir le coefficient avant de définir l'angle par `RotAngle`.
Dans l'exemple qui suit, le point M a pour image M' dans la rotation de centre O et d'angle $\left(\overrightarrow{AB},\overrightarrow{AC}\right)$; le point N a pour image N' dans la rotation de centre O et d'angle $\frac{1}{2}\left(\overrightarrow{AB},\overrightarrow{AC}\right)$.




```
\psset{unit=1cm,CodeFig=true,CodeFigColor=red}
\def\xmin{-1}   \def\xmax{4}
\def\ymin{-1}   \def\ymax{4}
\begin{pspicture}[showgrid](\xmin,\ymin)(\xmax,\ymax)
  \pstGeonode[PosAngle={0,180,0},CurveType=polyline]
      (1.5,-0.5){B}(0,-0.5){A}(1,0.5){C}
  \def\angle{\pstAngleAOB{B}{A}{C}}
  \pstGeonode[PosAngle={-135,-90,-90}]
      (0,2){O}(2,2){M}(3.1,2.4){N}
  \psset{PosAngle=90,arrowscale=2}
  \pstRotation[RotAngle=\angle]{O}{M}
  \pstRotation[AngleCoef=0.5,RotAngle=\angle]{O}{N}
\end{pspicture}
```

16.4 Translation

Deux points permettent de définir une translation ; ce sont les deux premiers paramètres qu'il faudra donner à `\pstTranslation`, le troisième étant la liste des points dont on veut les images.

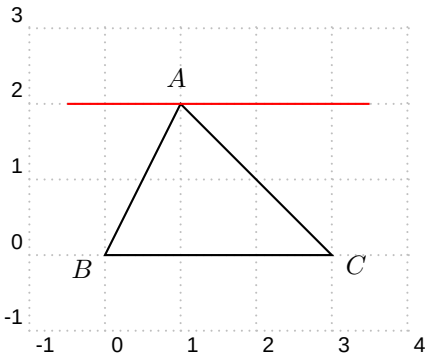



```
\psset{unit=1cm,CodeFig=true,CodeFigColor=red}
\def\xmin{-1}   \def\xmax{4}
\def\ymin{-1}   \def\ymax{3}
\begin{pspicture}[showgrid](\xmin,\ymin)(\xmax,\ymax)
  \pstGeonode[PosAngle={-135,0,-135,-135}]
          (0,0){A}(1.5,0.5){B}(0,2){M}(1.5,1.5){N}
  \pstLineAB{A}{B}
  \pstTranslation{A}{B}{M,N}
\end{pspicture}
```

L'option `CodeFig=true` trace les segments $[MM']$ et $[NN']$.

On peut utiliser une translation pour tracer une droite passant par un point donné et parallèle à une droite donnée, comme dans l'exemple ci-dessous.

Soit ABC un triangle. On veut tracer la droite Δ passant par A et parallèle à (BC).

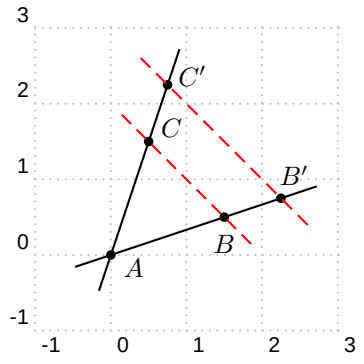



```
\psset{unit=1cm,PointSymbol=none}
\def\xmin{-1}   \def\xmax{4}
\def\ymin{-1}   \def\ymax{3}
\begin{pspicture}[showgrid](\xmin,\ymin)(\xmax,\ymax)
  \pstTriangle(0,0){B}(3,0){C}(1,2){A}
  \pstTranslation[PointName=none]{B}{C}{A}
  \pstLineAB[nodesepA=-1.5,nodesepB=0.5,
             linecolor=red]{A}{A'}
\end{pspicture}
```

Dans cet exemple, le point A' n'est pas dessiné dans la figure (`PointName=none`) mais on peut l'utiliser pour tracer la parallèle à (BC) passant par A.

16.5 Homothétie

Un centre et un rapport permettent de définir une homothétie. Le rapport sera défini par `HomCoef` et sera passé en option (obligatoire!) dans l'instruction `\pstHomO` ; cette instruction nécessite deux paramètres : le centre de l'homothétie, et la liste des points dont on veut les images. Par défaut, l'image du point B sera appelée B' sauf si on entre comme option en fin de traitement un autre nom, comme dans les autres transformations étudiées dans cette chronique.




```
\psset{unit=1cm}
\def\xmin{-1}   \def\xmax{3}
\def\ymin{-1}   \def\ymax{3}
\begin{pspicture}[showgrid](\xmin,\ymin)(\xmax,\ymax)
  \pstGeonode[PosAngle={-30,-90,30}]
             (0,0){A}(2,0.5){B}(0.5,1.5){C}
  \pstHomO[HomCoef=1.5,PosAngle={60,20}]{A}{B,C}
  \psset{linecolor=red,linestyle=dashed,nodesep=-0.5}
  \pstLineAB{B}{C} \pstLineAB{B'}{C'}
  \psset{linestyle=solid,linecolor=black}
  \pstLineAB{A}{B'}  \pstLineAB{A}{C'}
\end{pspicture}
```

L'option `CodeFig=true` est sans effet dans le cas d'une homothétie.

16.6 Références

Rappelons le nom de l'auteur de ce package fort efficace : Dominique Rodriguez.

Le premier mode d'emploi de ce package était en français et on peut le trouver en cliquant ici ; il date de 2005.
La dernière version, de 2015, est en anglais et on la trouve ici.

Vous retrouverez dans ces documents tout ce dont je vous ai parlé au cours de ces quatre chroniques consacrées au package `euclide`, et même un peu plus.