

Chronique 14

Euclide – Les intersections

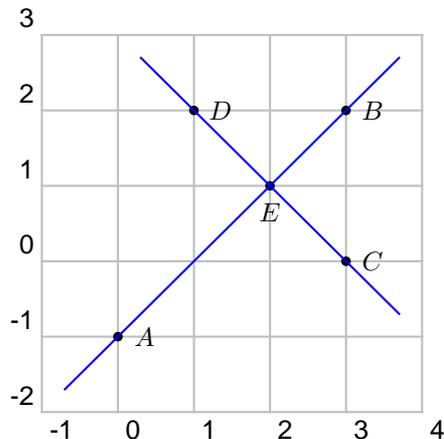
Le package `euclide` permet de déterminer des intersections de droites, cercles et courbes. Les quelques exemples de cette chronique devraient vous permettre de faire ce que vous voulez.

14.1 Intersection de droites

14.1.1 Figure simple

On commence par un exemple basique de détermination du point d'intersection de deux droites ; on définit quatre points A , B , C et D par `\pstGeonode`, puis on définit le point d'intersection des droites (AB) et (CD) par l'instruction `\pstInterLL`.

On se souviendra que LL veut dire `Line Line` pour désigner l'intersection de deux lignes.



```
\psset{unit=1cm}
\def\xmin {-1} \def\xmax {4}
\def\ymin {-2} \def\ymax {3}
\begin{pspicture}(\xmin,\ymin)(\xmax,\ymax)
  \psgrid[subgriddiv=1,gridcolor=lightgray]
  \pstGeonode(0,-1){A}(3,2){B}(3,0){C}(1,2){D}
  \pstInterLL[PosAngle=-90]{A}{B}{C}{D}{E}
  \psset{linecolor=blue,nodesep=-1}
  \pstLineAB{A}{B} \pstLineAB{C}{D}
\end{pspicture}
```

L'instruction `\pstInterLL` a besoin de cinq paramètres : les deux premiers correspondent à la première droite, les deux suivants à la seconde droite, et le cinquième est le point d'intersection de ces deux droites. Attention, ce point d'intersection peut ne pas exister !

Là aussi, on peut positionner le nom du point d'intersection au moyen de la variable `PosAngle`.

Une petite nouveauté dans ce dessin, la variable `nodesep` que l'on avait déjà vue lors de la création d'arbres pondérés :

- quand elle est à 0 (valeur par défaut), la ligne tracée démarre et arrive précisément sur les points ;
- quand elle est strictement positive, la ligne démarre après le point de départ et arrive avant le point d'arrivée ;

- quand elle est négative, la ligne démarre avant le point de départ et se poursuit après le point d'arrivée. Cela permet de tracer des demi-droites, ou des droites passant par deux points que l'on connaît sans avoir à utiliser l'équation de la droite et `\psplot`.

On peut différencier le point de départ et le point d'arrivée respectivement avec les variables `nodesepA` et `nodesepB`.

Voici ce que cela donne, après que l'on a défini les points A et B par :

```
\pstGeonode[PosAngle={90,90}](2,0){A}(6,0){B}
```

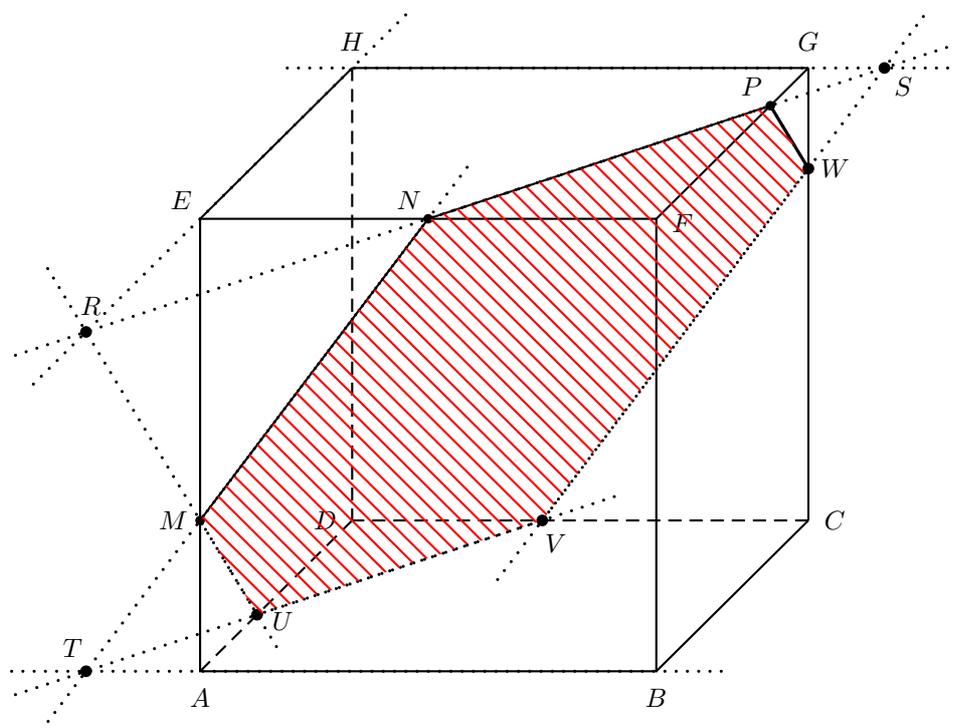
A •	_____	B •	<code>\pstLineAB{A}{B}</code>
A •	_____	B •	<code>\pstLineAB[nodesep=-1]{A}{B}</code>
A •	_____	B •	<code>\pstLineAB[nodesepB=-1]{A}{B}</code>
A •	_____	B •	<code>\pstLineAB[nodesep=1]{A}{B}</code>

14.1.2 Section de cube

Pour illustrer de façon plus convaincante (si besoin est) l'intersection de deux droites, j'ai ressorti de mes cartons un vieux exercice de section de cube que l'on faisait autrefois en 1^{re} S.

Dans le cube $ABCDEFGH$, on place les points M , N et P respectivement sur les segments $[AE]$, $[EF]$ et $[FG]$ comme sur la figure ci-dessous. On cherche la section du cube par le plan (MNP) .

La méthode consiste à tracer d'autres points extérieurs au cube qui vont permettre de déterminer les intersections des arêtes du cube avec le plan (MNP) . Avec `PstTricks`, il faut calculer les coordonnées des points d'intersection et tracer les bonnes droites ; avec `pst-eucl`, on peut directement définir les points comme intersections de droites puis les utiliser ensuite.



Voici le code de cette figure :

```

\psset{unit=1cm,PointSymbol=none}
\def\xmin{-2} \def\xmax{11} \def\ymin{0} \def\ymax{10}

\begin{pspicture}(\xmin,\ymin)(\xmax,\ymax)

%%% construction du cube

\pstGeonode[PosAngle={-90,-90,-10,135},CurveType=polygon]
(1,1){A}(7,1){B}(7,7){F}(1,7){E}%%% ABFE
\pstGeonode[PosAngle={0,180,90,90}](9,3){C}(3,3){D}(3,9){H}(9,9){G}

\pstLineAB{B}{C} \pstLineAB{C}{G}
\pstLineAB{F}{G} \pstLineAB{G}{H}
\pstLineAB{H}{E}

%%% arêtes cachées

\pstLineAB[linestyle=dashed]{A}{D}
\pstLineAB[linestyle=dashed]{C}{D}
\pstLineAB[linestyle=dashed]{H}{D}

%%% placement des points M, N et P et tracé des segments MN et NP

\psset{PointSymbol=*}%%% on marque les nouveaux points
\pstGeonode[PosAngle={180,135,135},CurveType=polyline]
(1,3){M}(4,7){N}(8.5,8.5){P}

%%% Construction de la section (tracés en pointillés)

\psset{linestyle=dotted,linewidth=1.2pt,nodesep=-1}
\pstInterLL[PosAngle=80]{H}{E}{N}{P}{R}% (HE) inter (NP) donne R
\pstLineAB{R}{H}
\pstInterLL[PosAngle=-45]{H}{G}{N}{P}{S}% (HG) inter (NP) donne S
\pstLineAB{S}{H}
\pstLineAB{R}{S}
\pstInterLL[PosAngle=120]{M}{N}{A}{B}{T}% (MN) inter (AB) donne T
\pstLineAB{T}{N}
\pstLineAB{T}{B}
\pstInterLL[PosAngle=-15]{R}{M}{A}{D}{U}% (RM) inter (AD) donne U
\pstLineAB[nodesepB=-0.5]{R}{U}
\pstInterLL[PosAngle=-60]{T}{U}{D}{C}{V}% (TU) inter (DC) donne V
\pstLineAB{T}{V}
\pstInterLL[PosAngle=0]{V}{S}{G}{C}{W}% (VS) inter GC) donne W
\pstLineAB{V}{S}
\pstLineAB[linestyle=solid,nodesep=0]{P}{W}

%%% remplissage de la section MNPWVU en hachuré et rouge

\pspolygon[fillstyle=vlines,hatchcolor=red](M)(N)(P)(W)(V)(U)

\end{pspicture}

```

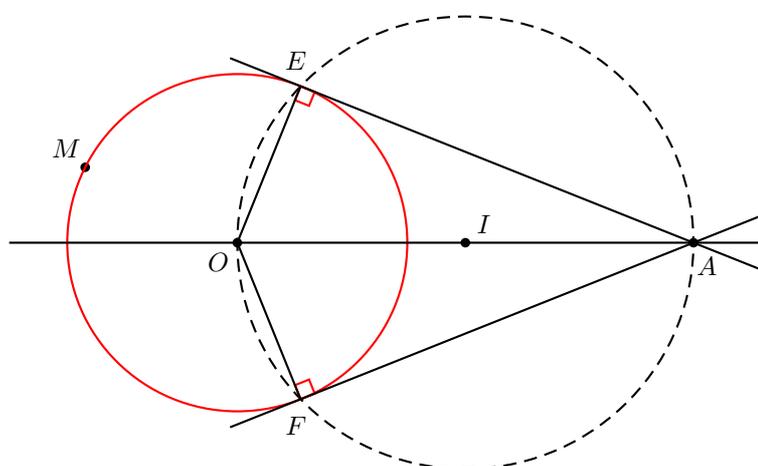
14.2 Intersection de cercles

C'est l'instruction `\pstInterCC` qui permet de déterminer l'intersection de deux cercles; on se souviendra que `CC` veut dire `Circle Circle`.

Chacun des cercles doit être défini par son centre et un point. L'instruction `\pstInterCC` aura besoin de six paramètres : le centre et un point du premier cercle, le centre et un point du second cercle, et les noms des deux points d'intersection des deux cercles (s'ils existent).

Pour illustrer les intersections de cercles, on va déterminer les deux tangentes à un cercle issues d'un point extérieur au cercle.

Soient O et M deux points. On appelle \mathcal{C} le cercle de centre O passant par M . Soit A un point extérieur au cercle \mathcal{C} . On veut tracer les deux tangentes au cercle \mathcal{C} passant par A ; elles coupent le cercle en E et F .



On place d'abord les trois points O , M et A au moyen de `\pstGeonode`, puis le cercle de centre O passant par M avec `\pstCircleOA`. Après avoir déterminé le milieu I de $[OA]$ avec `\pstMiddleAB`, on trace le cercle de centre I passant par A (en mode tirets).

On cherche ensuite les deux points d'intersection des deux cercles (une seule commande `\pstInterCC`) puis on complète la figure en traçant les tangentes.

Voici le code de la figure :

```
\psset{unit=1cm}
\def\xmin{-4} \def\xmax{7} \def\ymin{-4} \def\ymax{4}

\begin{pspicture}(\xmin,\ymin)(\xmax,\ymax)

%% placement des points O, M et A
\pstGeonode[PosAngle=-135,-60,135](0,0){O}(6,0){A}(-2,1){M}

%% tracés des cercles
\pstCircleOA[linecolor=red]{O}{M}
\pstMiddleAB[PosAngle=45]{O}{A}{I}
\pstCircleOA[linestyle=dashed]{I}{A}
```

```

%% intersection des deux cercles
\pstInterCC[PosAngleA=100,PosAngleB=-100,PointSymbol=none]{O}{M}{I}{A}{E}{F}

\pstLineAB[nodesepA=-3,nodesepB=-1]{O}{A}

%% codages angles droits
\pstRightAngle[linecolor=red,RightAngleSize=0.2]{O}{E}{A}
\pstRightAngle[linecolor=red,RightAngleSize=0.2]{O}{F}{A}

%% tangentes
\pstLineAB[nodesep=-1]{A}{E}
\pstLineAB[nodesep=-1]{A}{F}

%% rayons
\pstLineAB{O}{E}
\pstLineAB{O}{F}

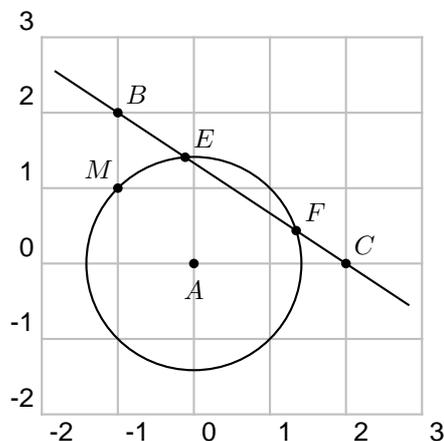
\end{pspicture}

```

14.3 Intersection droite – cercle

Cercle de centre A passant par M

Voici un premier exemple dans lequel on voit l'utilisation de `\pstInterLC` (L pour Line et C pour Circle) :



```

\psset{unit=1cm}
\def\xmin{-2} \def\xmax{3}
\def\ymin{-2} \def\ymax{3}
\begin{pspicture}(\xmin,\ymin)(\xmax,\ymax)
\psgrid[subgriddiv=1,gridcolor=lightgray]
\pstGeonode[PosAngle={-90,45,45,135}]
(0,0){A}(-1,2){B}(2,0){C}(-1,1){M}
\pstCircleOA{A}{M}
\pstLineAB[nodesep=-1]{B}{C}
\pstInterLC[PosAngle=45]{B}{C}{A}{M}{E}{F}
\end{pspicture}

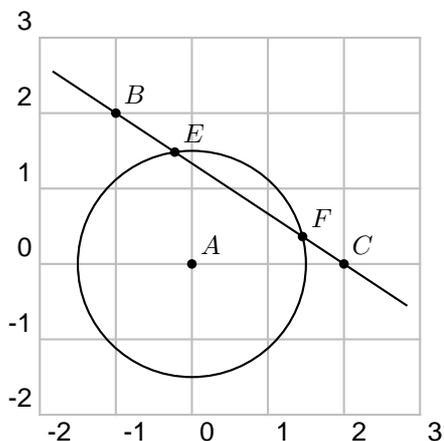
```

L'instruction `\pstInterLC` a besoin de six paramètres : deux points qui définissent la droite, le centre et le point du cercle, et les deux points d'intersection.

On retiendra que dans l'instruction `\pstInterLC`, le L est avant le C, donc on met d'abord les paramètres de la droite puis ceux du cercle.

Cercle de centre A et de rayon k

On peut tracer des cercles de différentes façons ;voici un autre exemple d'intersection . On remarque que le quatrième paramètre de `\pstInterLC` est vide puisque le cercle est défini par son centre et son rayon qui vaut ici 1,5.



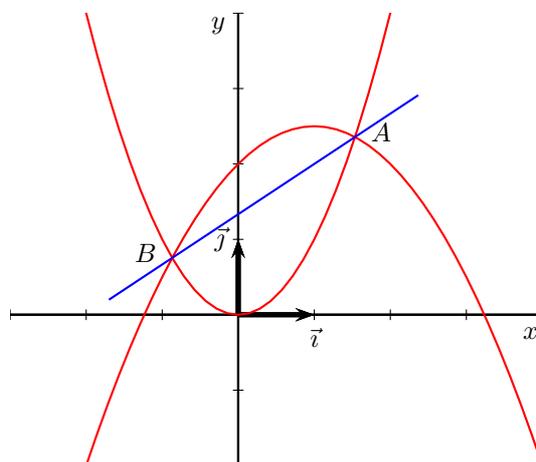
```
\psset{unit=1cm,PosAngle=45}
\def\xmin{-2} \def\xmax{3}
\def\ymin{-2} \def\ymax{3}
\begin{pspicture}(\xmin,\ymin)(\xmax,\ymax)
\psgrid[subgriddiv=1,gridcolor=lightgray]
\pstGeonode(0,0){A}(-1,2){B}(2,0){C}
\pstCircleOA[Radius=\pstDistVal{1.5}]{A}{}
\pstLineAB[nodesep=-1]{B}{C}
\pstInterLC[Radius=\pstDistVal{1.5}]
{B}{C}{A}{}{E}{F}
\end{pspicture}
```

14.4 Intersection de courbes

Avec `euclide` on peut déterminer les intersections de deux courbes représentant des fonctions ; c'est `\pstInterFF` qu'il faut utiliser (FF pour `Function Function`).

Le mode d'emploi de `pst-eucl` précise que l'intersection est déterminée par la méthode de NEWTON ; il faudra donc donner un point de départ à la recherche, tout en comprenant que cette recherche peut ne pas être fructueuse !

Il faut autant d'instructions `\pstInterFF` que de points d'intersection.



On va déterminer les points d'intersection A et B des courbes représentant les fonctions f et g définies par $f(x) = x^2$ et $g(x) = \frac{-x^2 + 2x + 4}{2}$. En traçant les courbes représentant ces fonctions,

on constate qu'elles ont deux points d'intersection dont les abscisses sont proches de -1 et de $1,5$; ce sont donc ces deux valeurs qu'il faudra donner à `\pstInterFF` pour obtenir les points d'intersection. L'instruction `\pstInterFF` a besoin de quatre paramètres : le code de la première fonction, le code de la seconde, une valeur qui correspond au point de départ de la recherche par la méthode de NEWTON, et le nom du point d'intersection (s'il est trouvé).

Voici le code de la figure :

```

\psset{unit=1cm}
\def\xmin{-3} \def\xmax{4} \def\ymin{-2} \def\ymax{4}

\begin{pspicture*}(\xmin,\ymin)(\xmax,\ymax)

%% axes et vecteurs
\psaxes[ticks=-2pt 2pt, labels=none]
(0,0)(\xmin,\ymin)(\xmax,\ymax)[$x$, -120][$y$, 210]
\psaxes[linewidth=1.8pt]{->}(0,0)(1,1)[$\vec{\imath}$, d][$\vec{\jmath}$, 180]

%% définitions des deux fonctions
\def\f{x x mul}
\def\g{x -1 x mul 2 add mul 4 add 2 div}

%% tracés des deux fonctions
\psplot[linecolor=red]{\xmin}{\xmax}{\f}
\psplot[linecolor=red]{\xmin}{\xmax}{\g}

\psset{PointSymbol=none}

%% premier point d'intersection A en partant de l'abscisse 1,5
\pstInterFF[PosAngle=10]{\f}{\g}{1.5}{A}

%% second point d'intersection B en partant de l'abscisse -1
\epstInterFF[PosAngle=170]{\f}{\g}{-1}{B}

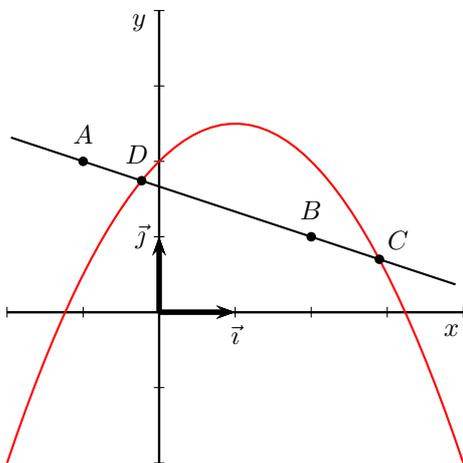
%% droite (AB)
\pstLineAB[nodesep=-1, linecolor=blue]{A}{B}

\end{pspicture*}

```

14.5 Intersection courbe – droite

Un exemple d'intersection courbe – droite créée par `\pstInterFL`.



```

\psset{unit=1cm}
\def\xmin{-2} \def\xmax{4}
\def\ymin{-2} \def\ymax{4}
\begin{pspicture*}(\xmin,\ymin)(\xmax,\ymax)
\psaxes[ticks=-2pt 2pt, labels=none](0,0)
(\xmin,\ymin)(\xmax,\ymax)[$x$, -120][$y$, 210]
\psaxes[linewidth=1.8pt]{->}(0,0)(1,1)
[$\vec{\imath}$, d][$\vec{\jmath}$, 180]
\def\f{x -1 x mul 2 add mul 4 add 2 div}
\psplot[linecolor=red]{\xmin}{\xmax}{\f}
\pstGeonode[PosAngle=90](-1,2){A}(2,1){B}
\pstLineAB[nodesepA=-1, nodesepB=-2]{A}{B}
\pstInterFL[PosAngle=45]{\f}{A}{B}{2}{C}
\pstInterFL[PosAngle=100]{\f}{A}{B}{0}{D}
\end{pspicture*}

```

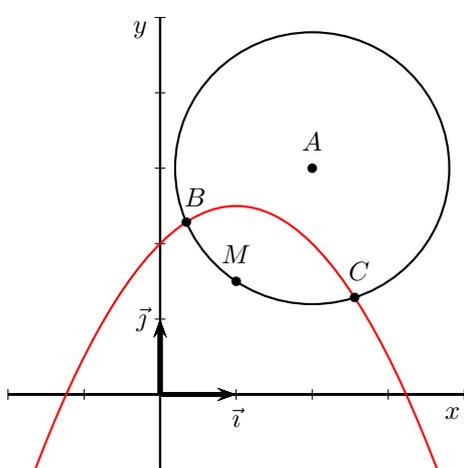
Il faut autant d'instructions `\pstInterFL` que de points d'intersection. Cette instruction a besoin de cinq paramètres : le code de la fonction, les deux points définissant la droite, la valeur de départ de la recherche par la méthode de NEWTON, et le nom du point d'intersection.

14.6 Intersection courbe – cercle

On termine cette série d'intersections par l'intersection d'une courbe et d'un cercle; vous avez sans doute deviné que l'instruction va s'appeler `\pstInterFC`! Et comme le F est avant le C, on mettra le code de la fonction avant les éléments de définition du cercle.

On définit et on trace la courbe représentant la fonction f qui à x associe $\frac{-x^2 + 2x + 4}{2}$. On place

les points $A(2; 3)$ et $M(1; 1,5)$. On trace le cercle de centre A passant par M , puis on cherche les points d'intersection de la courbe et du cercle.



```

\psset{unit=1cm}
\def\xmin{-2} \def\xmax{4}
\def\ymin{-1} \def\ymax{5}
\begin{pspicture}(\xmin,\ymin)(\xmax,\ymax)
\psaxes[ticksize=-2pt 2pt, labels=none](0,0)
(\xmin,\ymin)(\xmax,\ymax)[$x$, -120][$y$, 210]
\psaxes[linewidth=1.8pt]{->}(0,0)(1,1)
[$\vec{\imath}$, d][$\vec{\jmath}$, 180]
\def\f{x -1 x mul 2 add mul 4 add 2 div}
\psplot[linecolor=red]{\xmin}{\xmax}{\f}
\pstGeonode[PosAngle=90](2,3){A}(1,1.5){M}
\pstCircleOA{A}{M}
\pstInterFC[PosAngle=70]{\f}{A}{M}{0}{B}
\pstInterFC[PosAngle=80]{\f}{A}{M}{2}{C}
\end{pspicture}

```

La méthode de recherche est encore la méthode de NEWTON pour laquelle il faut entrer une valeur de départ. L'instruction `\pstInterFC` a besoin de cinq paramètres : le code de la fonction, le centre du cercle, un point du cercle, la valeur de départ de la recherche, et le nom du point d'intersection. Il faudra autant d'instructions `\pstInterFC` que de points d'intersection.

Attention! Pour déterminer l'intersection d'une courbe et d'un cercle, il est impératif que le cercle soit défini par son centre et un point; on ne pourra pas faire appel à `Radius` ni à `Diameter` dans ce cas.