pst-pursuitcurve (version 2)

21 novembre 2020

1 La commande \psPursuitCurve[options](x,y)

C'est une commande destinée à illustrer le problème de la course poursuite, elle a été écrite en collaboration avec Jürgen Gilg. Le site suivant contient une superbe étude de ce problème :

https://mathcurve.com/courbes2d/poursuite/poursuite.shtml

qui a inspiré ce package ainsi que l'étude théorique de Jürgen Gilg.
Le couple choisi dans l'étude que propose Jürgen Gilg dans un autre document est (chien, maître), mais on peut citer d'autres couples de protagonistes comme (renard, lapin) et plus généralement (prédateur, cible).
Le chien court après son maître en le gardant en point de mire. Les deux ont des vitesses constantes, différentes ou égales et on s'intéresse à la trajectoire suivie par le chien selon le chemin suivi par le maître. Les deux partent de leurs positions respectives en même temps.
Cette version diffère peu de la version (1) où le cas $k > 1$ doit être précédé d'un calcul ou des essais pour déterminer au bout de combien de temps le chien rattrape le maître, car le package 'pst-ode' ne peut plus faire les calculs dès que la distance entre le chien et le maître s'annule et un message d'erreur s'affiche. Dans le cas du parcours rectiligne, Gilg Jürgen a établi la formule permettant de déterminer au bout de combien de temps le chien rattrape son maître, (a, b) sont les coordonnées du début de la course du chien :

$$t = b + \frac{a^2}{2\left(1-\frac{1}{k}\right)\left(\sqrt{a^2+b^2}+b\right)} - \frac{\sqrt{a^2+b^2}+b}{2\left(1+\frac{1}{k}\right)}$$

Par ailleurs, dans son document, il propose une macro avec xint pour le calculer.
Dans le cas du parcours circulaire, il faut, dans cette version, augmenter le temps maximal de calcul pour déterminer empiriquement au bout de combien de temps le chien rattrape le maître.
Dans cette version (2), on calcule la distance séparant le chien du maître à chaque date et lorsqu'elle est très petite $d < 10\text{—}3$, cet écart n'évolue plus, mais 'pst-ode' peut continuer à calculer sans se scratcher. Une fois repéré cet instant, on placer cette date comme valeur maximale de t pour 'pst-ode'. Si on laisse calculer 'pst-ode' au-delà de cette date chien et maître cheminent ensemble mais les vecteurs ne le vecteurs-vitesse du chien n'est plus cohérent avec son mouvement.

a) le maître suit un parcours rectiligne ;

b) le maître suit un parcours circulaire.

Cette commande comporte les options suivantes, dont les valeurs par défaut sont indiquées :

1) `[k=1]` : rapport entre la vitesse du chien et celle du maître :$k = \frac{v_c}{v_m}$:

2) `[t=10]` : durée en s ;

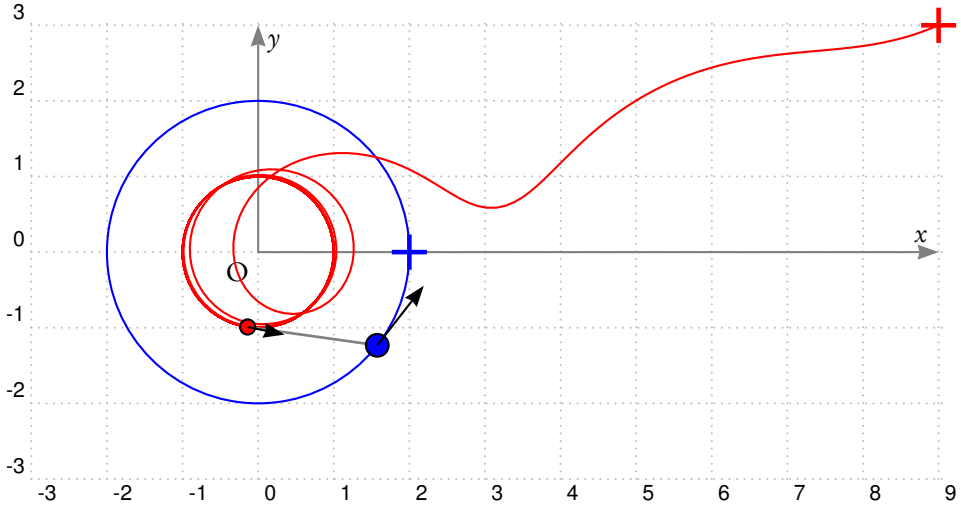
3) `[R=2]` : rayon du cercle décrit par le maître ;

4) `[target=0]` : indique le type de parcours suivi par la cible :

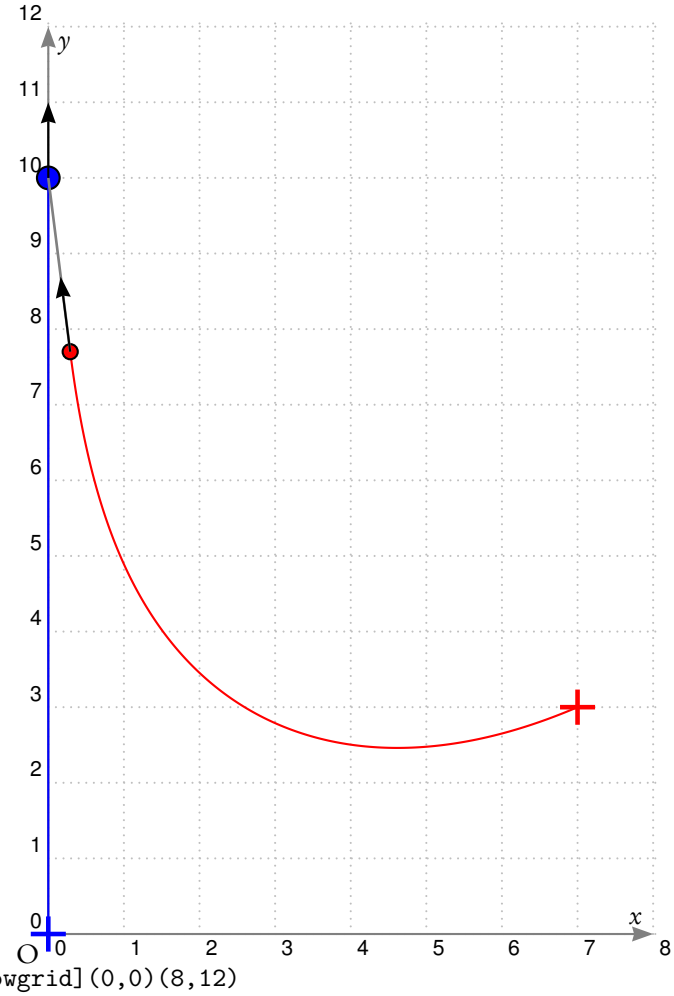
   - `[target=0]` = cercle
   - `[target=1]` = parcours suivant l'axe Oy

Le couple de coordonnées (x, y) qui doit suivre la commande donne la position initiale du chien. Le maître est alors au point $(0, 0)$ dans le cas du parcours rectiligne ou $(R, 0)$ dans le cas du parcours circulaire.

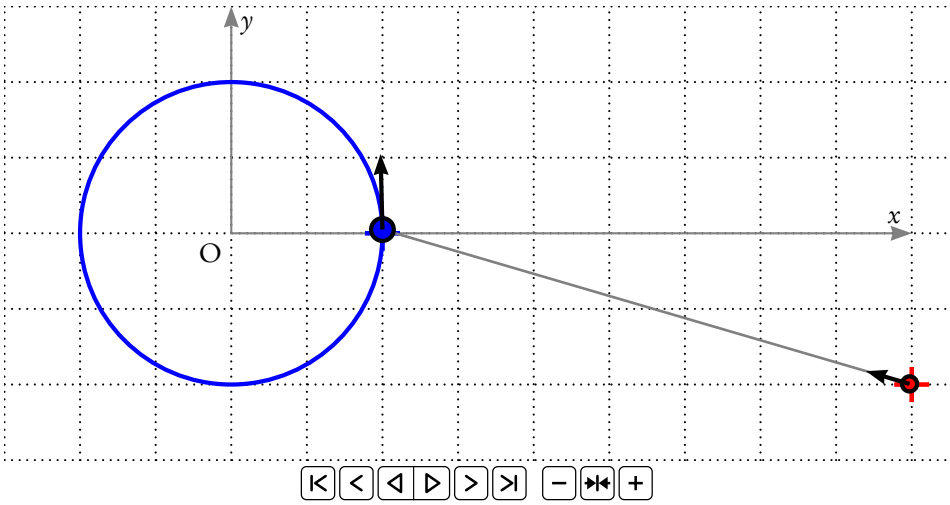
2 Exemples



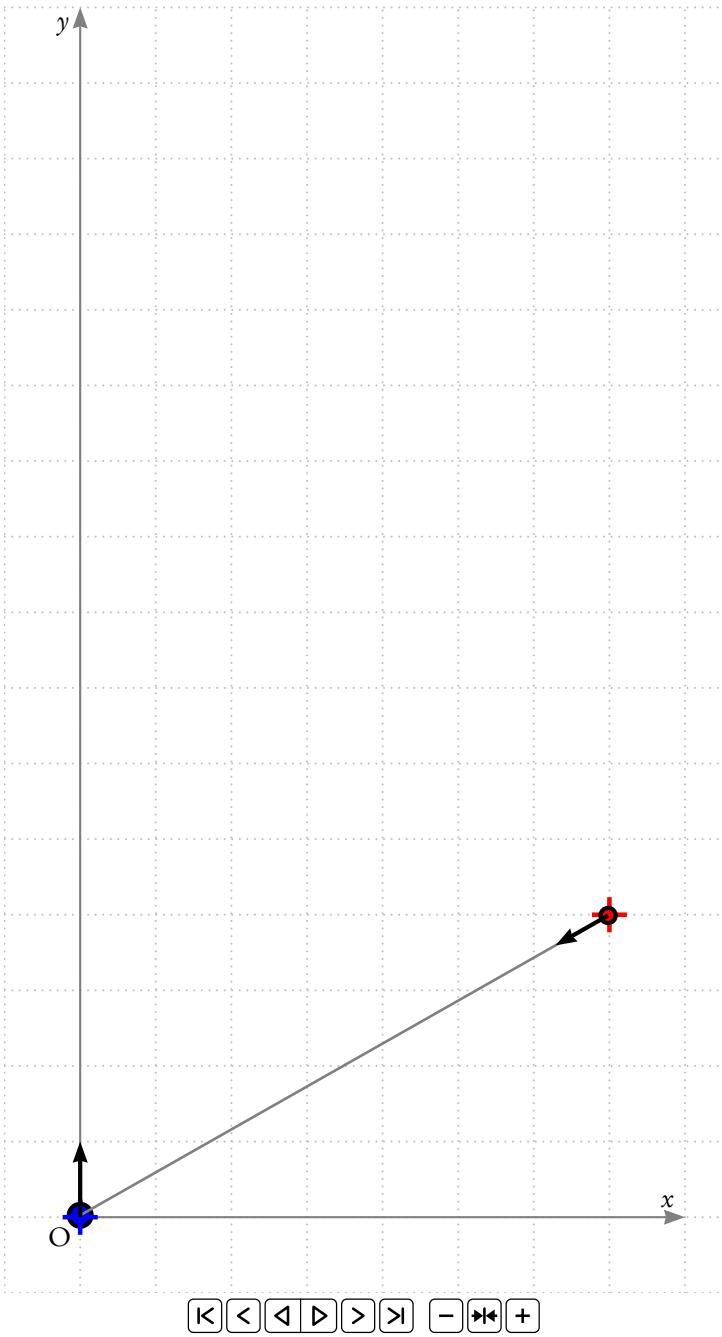

```
\begin{pspicture}[showgrid](-3,-3)(9,3)
\psline[arrowinset=0.1,arrowsize=0.2,linecolor=gray]{<->}(0,3)(0,0)(9,0)
\uput[dr](0,3){$y$}
\uput[ul](9,0){$x$}
\uput[dl](0,0){O}
\psPursuitCurve[t=125,plotpoints=1000,k=0.5,linecolor=red](9,3)
\end{pspicture}
```



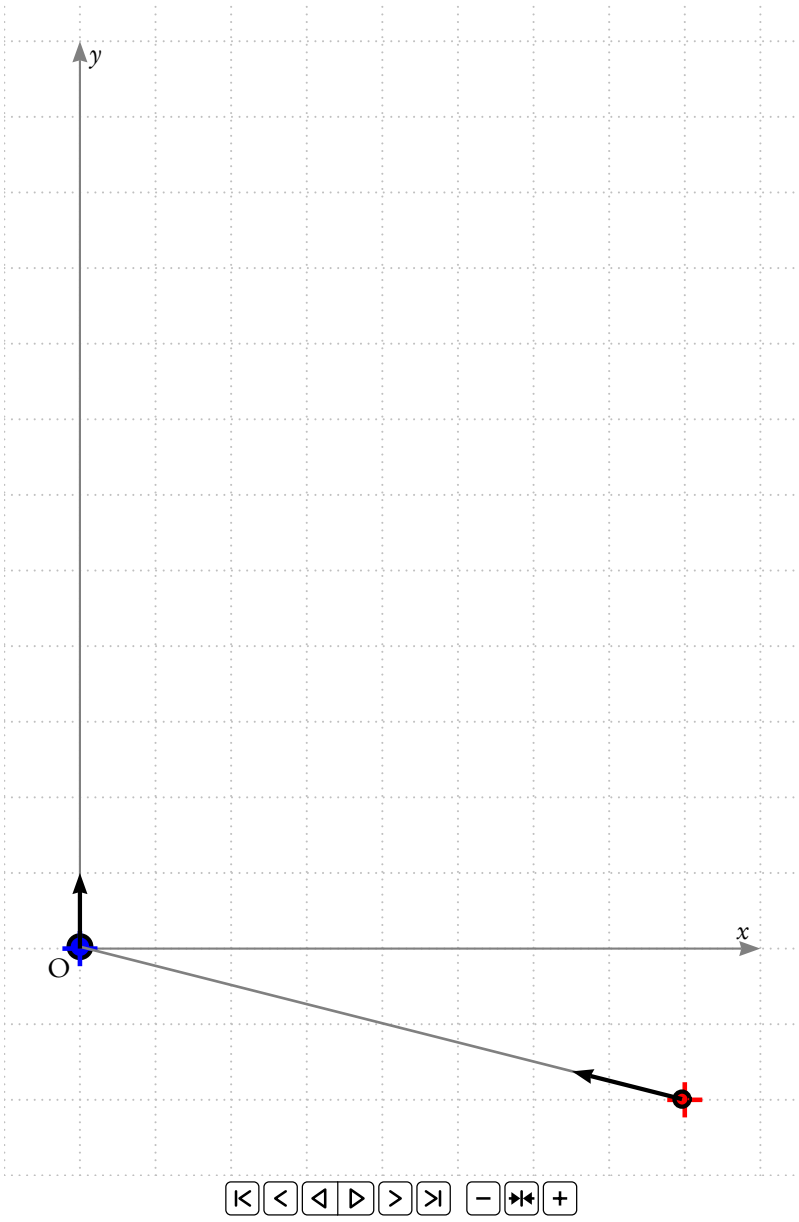

```
\begin{pspicture}[showgrid](0,0)(8,12)
\psline[arrowinset=0.1,arrowsize=0.2,linecolor=gray]{<->}(0,12)(0,0)(8,0)
\uput[dr](0,12){$y$}
\uput[ul](8,0){$x$}
\uput[dl](0,0){O}
\psPursuitCurve[t=10,plotpoints=200,k=1,linecolor=red,target=1](7,3)
\end{pspicture
```




```
\begin{animateinline}[controls,
 begin={\begin{pspicture}[showgrid](-3,-3)(9.5,3)
\psgrid[subgriddiv=0,griddots=10,gridlabels=0pt]
\psline[arrowinset=0.1,arrowsize=0.2,linecolor=gray]{<->}(0,3)(0,0)(9,0)
\uput[dr](0,3){$y$}
\uput[ul](9,0){$x$}
\uput[dl](0,0){O}},
end={\end{pspicture}}]{10}
\multiframe{100}{n=0.0+0.3,i=10+10}{
% on augmente le nombre de points de 10 a chaque intervalle de temps
\psPursuitCurve[t=\n,plotpoints=\i,k=0.6,linecolor=red,linewidth=2\pslinewidth](9,-2)
}
\end{animateinline}
```




```
\begin{animateinline}[controls,
 begin={\begin{pspicture}[showgrid](-1,-1)(8.5,15.5)
\psline[arrowinset=0.1,arrowsize=0.2,linecolor=gray]{<->}(0,15)(0,0)(8,0)
\uput[dr](0,12){$y$}
\uput[ul](8,0){$x$}
\uput[dl](0,0){O}},
end={\end{pspicture}}]{10}
\multiframe{100}{n=0.0+0.2,i=10+10}{
% on augmente le nombre de points de 10 a chaque intervalle de temps
\psPursuitCurve[t=\n,plotpoints=\i,k=0.8,linecolor=red,linewidth=2\pslinewidth,target=1](7,4)
}
\end{animateinline}
```




```
\begin{animateinline}[controls,
begin={\begin{pspicture}[showgrid](-1,-3)(9.5,12.5)
\psline[arrowinset=0.1,arrowsize=0.2,linecolor=gray]{<->}(0,12)(0,0)(9,0)
\uput[dr](0,12){$y$}
\uput[ul](9,0){$x$}
\uput[dl](0,0){O}},
end={\end{pspicture}}]{10}
\multiframe{24}{n=0.0+0.5,i=5+5}{
\psPursuitCurve[t=\n,plotpoints=\i,k=1.5,linecolor=red,linewidth=2\pslinewidth,target=1](8,-2)}
\end{animateinline}
```