

1 Loi de Descartes sur la reflexion

2 Fig Edit Graphe Pointeur Moire Save

```

1 assume(a:=[10.0,0,10])
  parameter(a,0.0,10.0,10.0)
2 assume(x:=[2.1,0,a])
  parameter(x,0.0,10.0,2.1)
3 assume(Y=[10.0,0,10])
  parameter(Y,0.0,10.0,10.0)
4 O:=point(0,0)
  point(0,0)
5 A:=point(a,0)
  point(a)
6 R:=point(0,Y)
  point((i)*Y)
7 M:=point(x,0)
  point(x)
8 miroir:=droite(O,A)
  droite(y=0)
9 mur:=perpendiculaire(A,miroir)
  droite(x=a)
10 incident:=couleur(droite(R,M),vert)
  droite(y=(-Y)/x*x+Y)
11 axe:=couleur(perpendiculaire(M,miroir),ligne_point+bleu)
  droite(x=x)
12 r:=symetrie(axe,R)
  point((i)*Y-2*(i)*(x+(i)*a-x)*(-x)*(-a)/((i)*(x+(i)*a-x)*(-i)*(x+(-i)*a-x))
13 reflechi:=couleur(demi_droite(M,r),magenta)
  demi_droite(point(x),point((i)*Y-(2*i)*(x+(i)*a-x)*(-x)*(-a)*(-i)/(x+(-i)*a-x))
14 H:=inter_unique(mur,reflechi)
  point((-Y*a+Y*x-(-i)*a*x)/((i)*x)
15 h:=x->longueur(A,H)::
  // Warning: A H declared as global variable(s)
  // End defining h
  "Done"
16 factoriser(h(x))
  Y*(a-x)/x
17 graphe(h(x),x=0..a,couleur=jaune)
  plotparam(x+(i)*sqrt((a+(-(-a)*x)*(-x)/x^2)^2+(-(-Y*a+Y*x)*(-x)
18 Hp:=inter_droite(mur,incident)
  point((Y*a-Y*x+(i)*a*x)/((i)*x)
19

```

The graph displays the geometric construction of the reflection of a point R across a line (miroir) to a point H. The coordinate system has x and y axes. A horizontal line (miroir) is drawn at y=0. A vertical line (mur) is drawn at x=a. A point R is on the y-axis at (0, Y). A point M is on the x-axis at (x, 0). A yellow curve (incident ray) starts at R and passes through M. A magenta line (reflected ray) starts at M and passes through H. A green line (ax) is perpendicular to the mirror at M. A blue line (axe) is perpendicular to the mirror at M. A red line (Hp) is the perpendicular bisector of RM. The graph also shows a point O at (0,0) and a point A at (a,0).

3 Pour $Y=5\text{m}$, $a=18\text{m}$, on veut savoir où placer M pour atteindre sur le mur un objet à 6m de haut

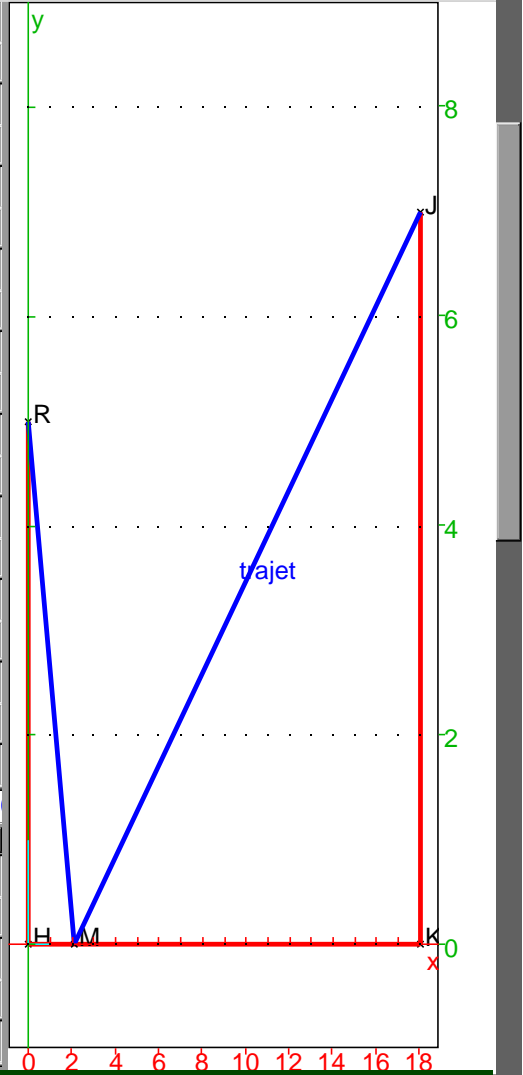
4 $Y:=5$; $a:=18$; résoudre($h(x)=6,x$)

(Done, Done, -90 $\frac{90}{11}$)

5 Exercice Roméo et Juliette :
Roméo veut rejoindre Juliette en allant lui cueillir une rose dans un jardin rectiligne. Sa trajectoire est une ligne brisée du style loi de Descartes sans bissectrice. Minimiser le trajet.
 $HR=5\text{m}$, $KJ=7\text{m}$, $HK=20\text{m}$

6 Fig Edit Graphe Pointeur Mode Save

```
1 H:=point(0,0)
  point(0,0)
2 R:=point(0,5)
  point(0,5)
3 K:=point(18,0)
  point(18,0)
4 J:=point(18,7)
  point(18,7)
5 couleur(polygone_ouvert(R,H,K,J),rouge+epaisseur_ligne_
  polygone(point(0,5),point(0,0),point(18,0),point(18,7))
6 assume(x:=14.04,18)
  parameter(x,0.0,18.0,14.04)
7 M:=point(x,0)
  point(x)
8 trajet:=couleur(polygone_ouvert(R,M,J),bleu+line_width_3
  polygone(point(0,5),point(x),point(18,7))
9 h:=unapply(longueur(R,M)+longueur(M,J),x)
  (x)->sqrt((-x)^2+25)+sqrt((x-18)^2+49)
10 graphe(h(x),x=0..18,couleur=vert)
  plotparam(x+(i)*(sqrt((-x)^2+25)+sqrt((x-18)^2+49)),x=
11 m:=point(x,h(x))
  point(x+(i)*(sqrt((-x)^2+25)+sqrt((x-18)^2+49)))
12 h(7.4);h(7.5);h(7.6)
  21.63360103,21.63330765,21.63359949
13
```



```
7 X:=7.4+10^(-6);
  Y:=h(7.4);
  tantque evalf(h(X))<Y faire
    Y:=evalf(h(X));
    X:=evalf(X+10^(-6));
  ftantque;
  X-10^(-6)
  Evaluation time: 5.54
```

(Done, Done, Done, 7.5)

8 Pour vérifier si on a trouvé la bonne solution, un petit bond dans le programme de 1ère

```

9 hp:=fonction_derivee(h)
// Success
' x' -> (-2*(-' x' ))*(1/2)*(1/((- ' x' )^2+25))*sqrt((- ' x' )^2+25)+2*(-' x' -18)*(1/2)*(1/((- ' x' -18)^2+49))*sqrt((- ' x' -18)^2+49)

```

```

10 resoudre(hp(t)=0,t)
-45 15
2

```

```

11 simplifier(hp(15/2))
0

```

12 EXERCICE DENTS DE LA MER

```

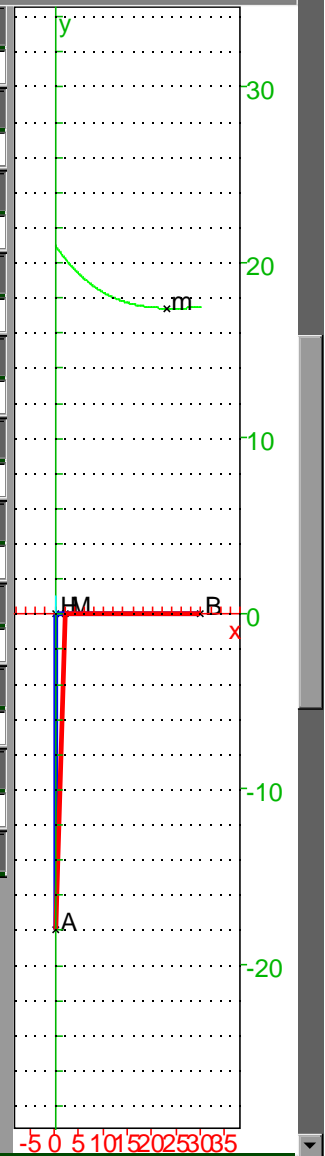
13 Fig Edit Graphe Pointeur Mode Save

```

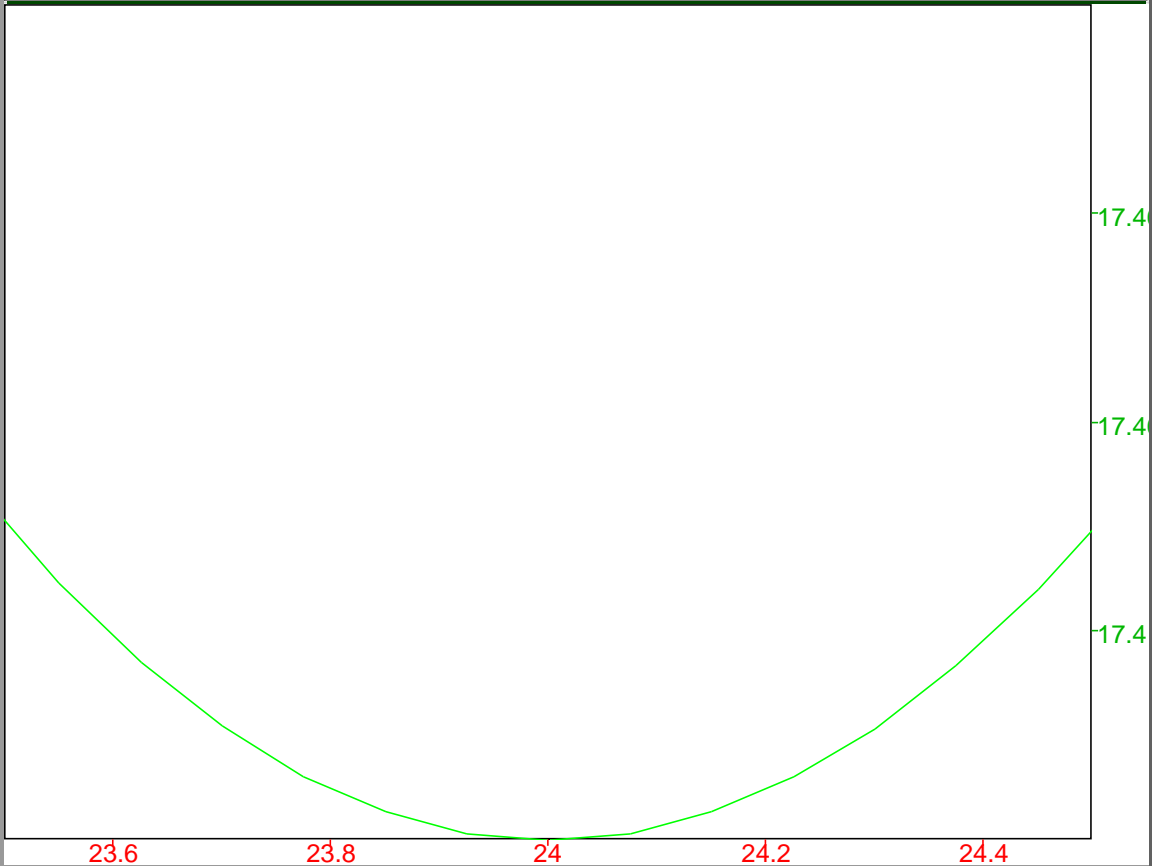
```

1 H:=point(0,0)
point(0,0)
2 B:=point(30,0)
point(30,0)
3 A:=point(0,-18)
point(0,-18)
4 assume(x:=[23.1,0,30])
parameter(x,0.0,30.0,23.1)
5 M:=point(x,0)
point(x)
6 couleur(polygone_ouvert(A,H,B),bleu+epaisseur_ligne_3)
polygone(point(0,-18),point(0,0),point(30,0))
7 couleur(polygone_ouvert(A,M,B),rouge+epaisseur_ligne_3)
polygone(point(0,-18),point(x),point(30,0))
8 t:=unapply(longueur(A,M)/(7.2/3.6)+longueur(M,B)/(9/3.6),x)
(x)->sqrt((-x)^2+324)/2.0+(-x+30)/2.5
9 graphe(t(x),x=0..30,couleur=vert)
plotparam(x+(i)*(sqrt((-x)^2+324)*0.5+(-x+30)*0.4),x=0.0..30.0375)
10 m:=point(x,t(x))
point(23.1,17.40248954)
11

```



14 graphe(t(x),x=0..30,couleur=vert)



```
15 mini(xo,p):={
X:=xo+10^(-p);Y:=t(xo);
tantque evalf(t(X))<Y faire
  Y:=evalf(t(X));
  X:=X+10^(-p);
ftantque;;
X-10^(-p)
}::
```

// Parsing mini
// Warning: X t Y declared as global variable(s) compiling mini

Done

16 mini(23.8,6)

Evaluation time: 8.86

24

17 En ttre

18 tp:=fonction_derivee(t)

// Success

$$'x' \rightarrow (-2 \cdot (-'x')) \cdot \left(\frac{1}{2}\right) \cdot \frac{1}{(-'x')^2 + 324} \cdot \sqrt{(-'x')^2 + 324} \cdot 0.5 + -0.4$$

19 tp(24)

0.0

20 resoudre(tp(t)=0,t)

[24]

