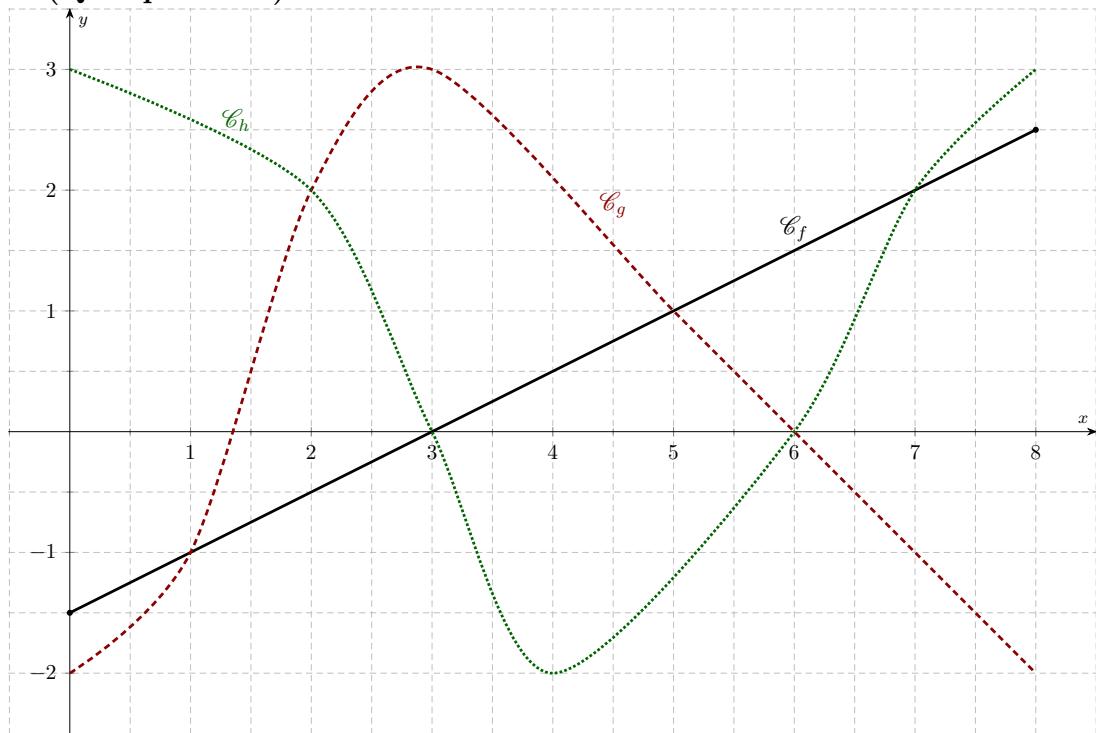
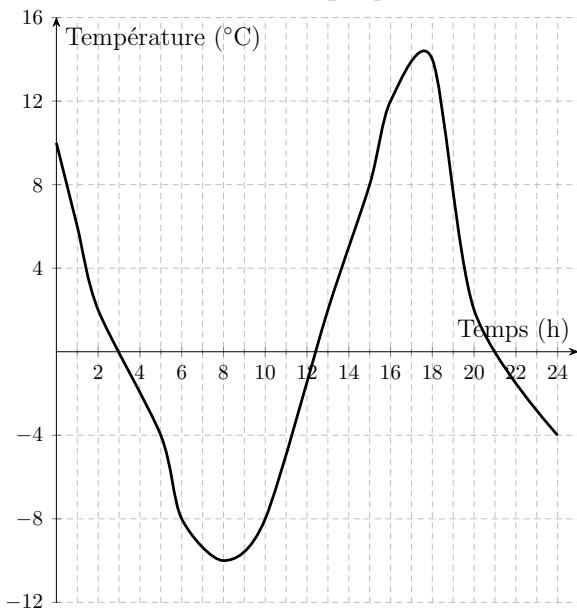


Fonctions - Exercices

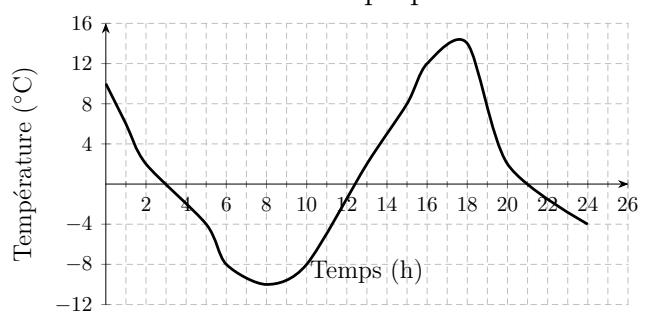
Théorème (Quelques test) :



Sans hauteurproptick

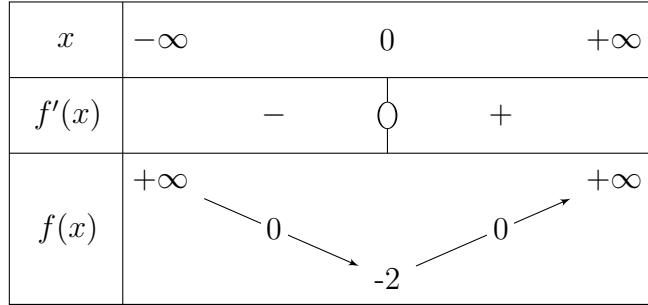


Avec hauteurproptick



x	$-\infty$	0	$+\infty$
$f'(x)$	-	0	+
$f(x)$	$+\infty$	0	$+\infty$

BLABLABLA taaaest



$\vec{u} \vec{AB} \vec{u} \vec{AB}$

$\vec{u} \vec{AB}$

NZDQRCUKEP

NZDQRCUKEP

Problème : minted et listing utilisent un package fragile qui ne peut pas s'encapsuler : compro ne fonctionne pas mais on peut le reproduire à la main :

```

1 def zero(f,a,b,seuil):
2     while (b-a)/2>seuil:
3         c=(a+b)/2
4         if f(c)==0:
5             return(c)
6         if f(a)*f(c)>0:
7             a=c
8         else:
9             b=c
10    return((a+b)/2)
11
12 def f(x):
13     return(x**2-2)
1
1 def zero(f,a,b,seuil):
2     while (b-a)/2>seuil:
3         c=(a+b)/2
4         if f(c)==0:
5             return(c)
6         if f(a)*f(c)>0:
7             a=c
8         else:
9             b=c
10    return((a+b)/2)
11
12 def f(x):
13     return(x**2-2)

```

Minted

Listing avec style minted

Ceci est un test : `def zero(f,a,b,seuil):` permet de Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

```

1 def zero(f,a,b,seuil):
2     while (b-a)/2>seuil:
3         c=(a+b)/2
4         if f(c)==0:
5             return(c)
6         if f(a)*f(c)>0:
7             a=c
8         else:
9             b=c

```

```
10     return((a+b)/2)
11
12 def f(x):
13     return(x**2-2)
```

Ceci est un test