

```

pour place :x :y //place le point de vraies coord x,y//
fpos entier((:L *(x -:A) / (:B -:A)) entier ((:H *(y -:D)) / (:C -:D))
fin

pour tracer :x :y //trace le segment vers le point de vraies coord x,y//
bc
va ph entier(:L *(x -:A) / (:B -:A)) entier ((:H *(y -:D)) / (:C -:D))
lc
fin

pour segm :x1 :y1 :x2 :y2
//trace le segment vers le point de vraies coord x1; y1 et x2 , y2//
donne "a entier((:L *(x1 -:A) / (:B -:A)) donne "b entier ((:H *(y1 -:D)) / (:C -:D))
donne "c entier(:L *(x2 -:A) / (:B -:A)) donne "d entier ((:H *(y2 -:D)) / (:C -:D))
segment [ :a :b ] [:c :d ]
fin

pour f :t
rends : t //ou autre définition//
fin

pour g :t :m
//ici m est un paramètre servant au problème suivant //
donne "v 0 donne "s 1
répète :m [ donne "v :v + ((:s * sin (boucle *:t)) / boucle) donne "s - :s ]
rends :v
fin

pour ro :t //cas de la spirale d'Archimède //
rends :t
fin

pour axes
si (:A *:B) <0 [ segm 0 :D 0 :C ]
si (:C *:D) <0 [ segm :A 0 :B 0 ]
fin

pour courbe :x1 :x2 :h :m
axes donne "x :x1
place :x g :x :m
tantque :x <= :x2 [tracer :x g :x :m donne "x :x +:h ]
fin

pour polaire :t1 :t2 :h :rang
donne "t :t1
si :rang = 1 place (ro :t) * (cos :t) (ro :t) *(sin :t)
si :rang = 2 place (cos (3* :t)) - (2*(sin (3 * :t))) Ln abs((sin(2* :t)) - (cos (5* :t)))
tantque :t <= :t2 [
    si :rang = 1 tracer (ro :t) *(cos :t) (ro :t) *(sin :t)
    si :rang = 2 tracer (cos (3* :t)) - (2*(sin (3 * :t))) Ln abs((sin(2* :t)) - (cos (5* :t)))
    donne "t :t + :h
]
fin

donne "A 0 donne "B 20 donne "C -2 donne "D 2
donne "L 480 donne "H 100
moderadian
déroule efftxt accélère
eff polaire -10,695 10,695 0,05 1
attends 2
eff polaire -60,695 60,695 0,01 2
modedegré

```