

Tracer une ellipse point par point

Le problème*

L'ellipse est définie par la donnée de son centre I , de la longueur a de son grand axe et de celle, b , de son petit axe. Quand ses axes sont parallèles à ceux du repère, c'est l'ensemble des points M de coordonnées $(x_M; y_M)$ tels que :

$$x_M = x_I + a \cos t$$

$$y_M = y_I + b \sin t$$

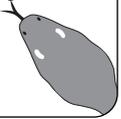
où l'angle t décrit l'intervalle $[0 ; 360[$ s'il est exprimé en degrés, $[0 ; 2\pi[$ s'il est exprimé en radians. On demande de programmer le tracé d'une telle ellipse.

Complément culturel* (1)

La définition de l'ellipse donnée sous la forme $x_M = f(t)$ et $y_M = g(t)$, où le réel t décrit un intervalle, est appelée « représentation paramétrique » de l'ellipse. On peut en donner d'autres définitions, plus géométriques. Ainsi, l'ellipse est – pour faire simple – la trajectoire décrite par la Terre lorsqu'elle tourne autour du Soleil. Elle fait partie de la famille des coniques, c'est-à-dire qu'on l'obtient en faisant la section d'un cône par un plan.

Les programmes

Python



Le programme « ellipse » définit la fonction ellipse dont les arguments sont les coordonnées de son centre x_I, y_I et la longueur de ses axes a et b . Python travaille en radians. Afin que l'ellipse soit décrite entièrement, il suffit de prendre un intervalle de longueur 2π .

Il est nécessaire d'importer les fonctions trigonométriques de la bibliothèque « turtle » en début de programme.

le programme

```
>>> from turtle import *
>>> from math import cos
>>> from math import sin
>>> def ellipse(xI,yI,a,b):
    reset()
    t=0
    up()
    goto(xI+a,yI)
    down()
    while t<6.29:
        xM=xI+a*cos(t)
        yM=yI+b*sin(t)
        goto(xM,yM)
        t=t+0.1
```

Scratch



```
quand [ ] pressé
effacer tout
à t attribuer 0
demander xI? et attendre
à xI attribuer réponse
demander yI? et attendre
à yI attribuer réponse
demander a? et attendre
à a attribuer réponse
demander b? et attendre
à b attribuer réponse
relever le stylo
aller à x: xI + a y: yI
abaissér le stylo
répéter indéfiniment si t < 360
  à xM attribuer xI + a * cos de t
  à yM attribuer yI + b * sin de t
  aller à x: xM y: yM
  changer t par 1
```



AlgoBox

Les fonctions circulaires (sin et cos) du logiciel AlgoBox fonctionnent avec un argument en radians (c'est-à-dire qu'il faut prendre la variable t en radians et non en degrés). Le tracé des points de l'ellipse se fait lors de l'exécution dans une fenêtre graphique dont les dimensions sont à régler avant le lancement du programme. Attention, la virgule des nombres décimaux doit être remplacée par un point.

```
VARIABLES
DEBUT_ALGORITHME
  t PREND_LA_VALEUR 0
  LIRE xI
  LIRE yI
  LIRE a
  LIRE b
  TRACER_POINT (xI+a,yI)
  TANT_QUE (t<6.28) FAIRE
    DEBUT_TANT_QUE
      xM PREND_LA_VALEUR xI+a*cos(t)
      yM PREND_LA_VALEUR yI+b*sin(t)
      TRACER_POINT (xM,yM)
      t PREND_LA_VALEUR t+0.1
    FIN_TANT_QUE
FIN_ALGORITHME
```

Complément culturel* (2)

L'ellipse peut être également définie par l'ensemble des points M dont la somme des distances $MF + MF'$ à deux points F et F' appelés « foyers » est une constante $2a$. Le centre I de l'ellipse est le milieu de $[FF']$ et la constante a est la longueur du grand axe $]IA[$ définie plus haut.

La distance focale $IF = c$ vérifie la relation : $b^2 + c^2 = a^2$, où b est la longueur du petit axe. L'angle t de la représentation paramétrique est également indiqué sur la figure ci-contre.

