

SPIROGRAPHE

Quentin Huan

I Présentation du sujet

Dans ce sujet, nous allons créer un petit module permettant de dessiner des rosaces en Python avec Numpy et Matplotlib.

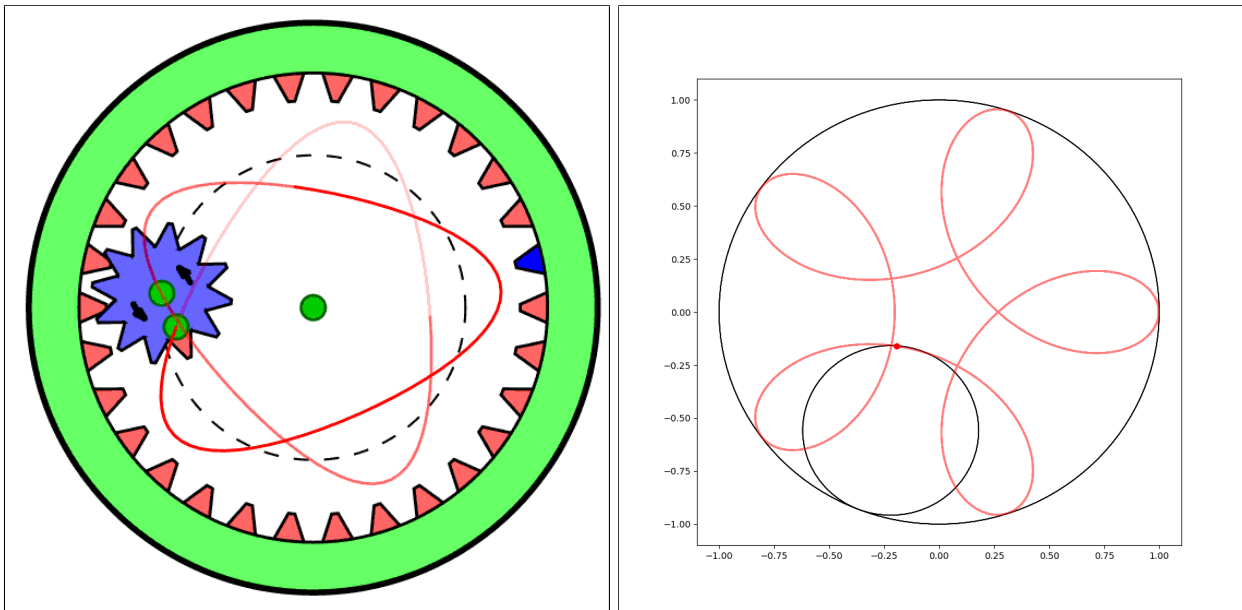


FIGURE 1 – principe de fonctionnement et exemple de rosace ($R_0 = 1, R_1 = 0.4, \omega_0 = 1, \omega_1 = 4$)

II Étapes :

A Cercle extérieur

i Équation paramétrique

On considère un point I placé sur le cercle extérieur $C_0(O_0, R_0)$: ce point se déplace sur le cercle à la vitesse angulaire ω_0 . Utilisez l'équation paramétrique d'un cercle $C(O_0, R_0)$ (1) pour tracer la trajectoire du point I au cours du temps.

$$\begin{cases} I_x = R_0 \cos(2\pi\omega_p \cdot t) \\ I_y = R_0 \sin(2\pi\omega_p \cdot t) \end{cases} \quad (1)$$

(on peut aussi utiliser les nombres complexes : $I = R_0 \exp(2\pi\omega_p \cdot j \cdot t)$)

ii Animation

Utilisez le module d'animation de Matplotlib pour animer I (voir `ex_animation.py` sur Sakai)

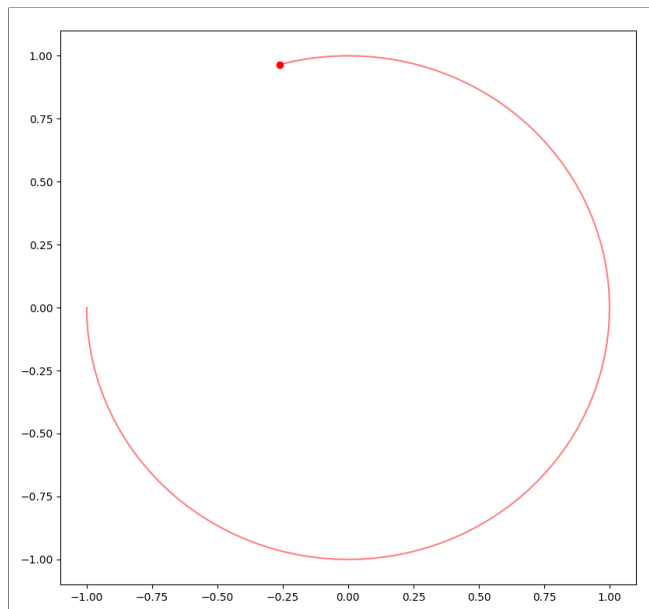


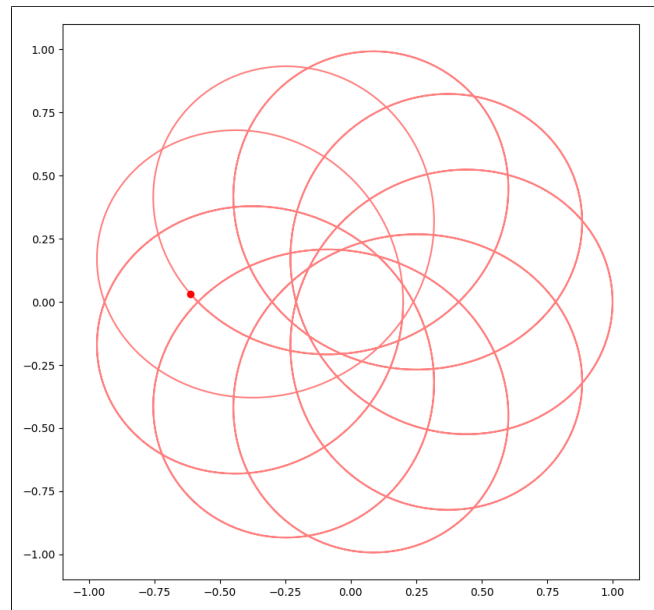
FIGURE 2 – trajectoire du point I

B Cercle intérieur

i Tracer la rosace

On considère un point P placé sur le cercle intérieur $C_1(O_1, R_1)$. Le cercle C_1 est en contact avec C_0 au point I et tourne à la vitesse ω_1 .

Tracez la trajectoire du point P au cours du temps en fonction de (O_1, R_1, ω_1)

FIGURE 3 – trajectoire du point P

ii Animer les cercles

Utilisez le module d'animation de Matplotlib pour animer toute la situation. On veut voir apparaître :

1. les deux cercles C_0 et C_1
2. le point P
3. la trajectoire du point P

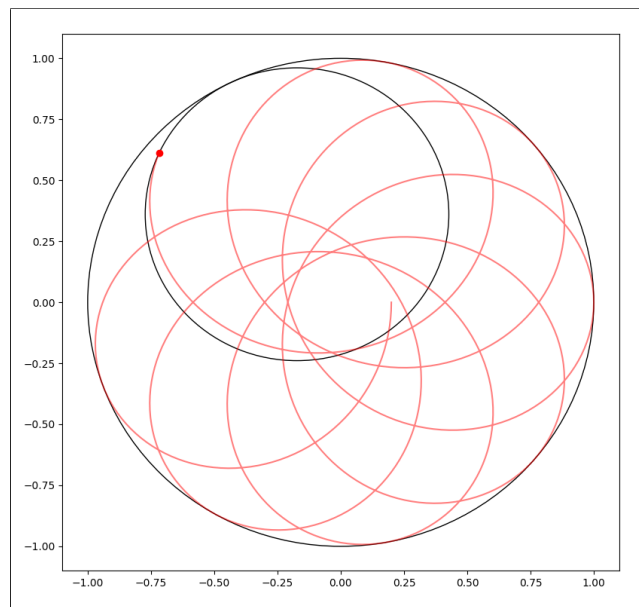


FIGURE 4 – Animation complète

C Idées supplémentaires

1. exporter en png
2. ajouter un paramètre de décallage
3. utiliser d'autres formes pour le contour externe
4. gérer un cercle qui roule sur l'extérieur du contour
5. ajouter un troisième cercle qui roule dans le deuxième
6. ajouter N cercles