Lycée Loges

Chapitre 5

Dessins, Graphes de fonctions Exercices

Simon Dauguet
simon.dauguet@gmail.com

11 décembre 2025

Exercice 1 :
On considère la fonction suivante :

```python
def sommef(n:int, f:'function' = (lambda x: x)) -> int:
    som = 0
    for i in range( n ):
        som += f( i )
    return som
```

Proposer un appel pour obtenir la valeur de $\sum_{k=0}^{10} k^2$. Puis pour obtenir la valeur de $\sum_{k=0}^{8} k^3$.

Exercice 2 (Courbe paramétrée) :
Une courbe paramétrée est une courbe constituée de points de la forme $(x(t), y(t))$ où x et y sont des fonctions du paramètres t.

Faire une fonction `cercle(ctr:tuple, r:float) -> None` qui trace le cercle de rayon `r` et de centre dont les coordonnées sont données par le tuple `ctr`.

Commenter le graphe que vous obtenez en appelant `cercle((0,0),1)`. Si un problème apparaît, modifier votre fonction pour y remédier.

Exercice 3 (Courbe du Dragon) :
On va construire la courbe du dragon dont le dessin est dessous.

1. Dans un repère orthonormé du plan, on considère une courbe polygonale $(M_0, M_1, \dots, M_n)$ définie par $M_0(0,0)$, $M_1(1,0)$ et telle que
   (i) $\forall k \in \{1, 2, \dots, n-1\}, \ \|\overrightarrow{M_k M_{k+1}}\| = 1$
   (ii) $\forall k \in \{2, 3, \dots, n-1\}$, l'angle formé par les vecteurs $\overrightarrow{M_{k-1}M_k}$ et $\overrightarrow{M_k M_{k+1}}$ est noté z_k. On peut montrer que
   $$\forall k \in \{1, \dots, n\}, \ \overrightarrow{M_{k-1}M_k} = (\cos(\theta_k), \sin(\theta_k)).$$
   où θ_k est l'angle formé par le vecteur $\overrightarrow{M_{k-1}M_k}$ et l'axe des abscisses. On a alors $z_k = \theta_{k+1} - \theta_k$.

   (a) Exprimer θ_k en fonction des z_p pour $p < k$.

(b) Écrire une fonction `tracer(z:list) -> None` qui prend en entrée une **z** et trace la courbe polygonale correspondante comme au dessus (donc le tracé $M_0, \ldots, M_n$)). On effacera les axes du graphes (et on n'oubliera pas que les axes sont orthonormés).

2. Tester la fonction `tracer` afin de dessiner un carré, un décagone (10 côtés), puis un cercle (polygone avec un grand nombre de côtés). Commenter la figure obtenue. Commenter ensuite le résultat obtenue par la commande `tracer([2*mt.pi/100 for k in range(100)])`.

3. La courbe que nous allons construire, appelée courbe du dragon, peut être obtenue à partir de pliage d'une feuille de papier. Penons une fine bande de papier et plions là plusieurs fois de suite dans le sens de la longueur. En la dépliant ensuite, on peut suivre le sens des plis et noter chaque pli par $+$ ou $-$ selon le sens observé. On obtient donc la séquence $\{1, 1, -1\}$ pour deux pliages et la séquence $\{1, 1, -1, 1, 1, -1, -1\}$ pour 3 pliages.

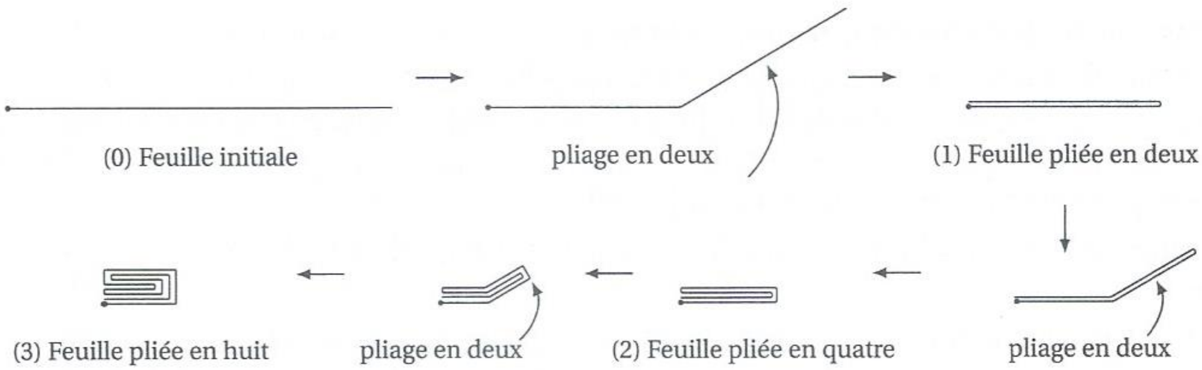


Figure 2.12. Étapes successives du pliage d'une bande de papier (vue de profil) permettant de déterminer la courbe du dragon.

Figure 2.13. Sens des plis formés après deux pliages (à gauche) et trois pliages (à droite).

Soit d_k la séquence obtenue après k pliages. On peut remarquer que la séquence d_{k+1} est de la forme

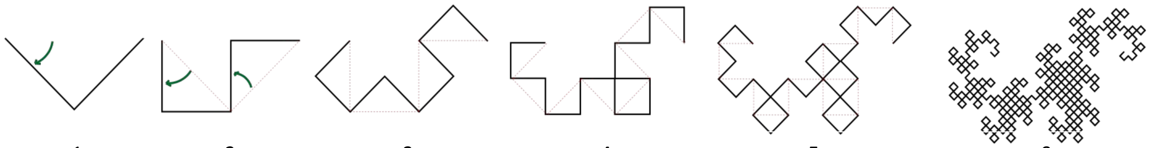
$$\{d_k, 1, \overline{d_k}\}$$

où $\overline{d_k}$ est la séquence d_k parcourue à l'envers et en inversant tous les signes. On pose $d_0 = \{\}$. On a donc $d_1 = \{1\}$, $d_2 = \{1, 1, -1\}$, $d_3 = \{1, 1, -1, 1, 1, -1, -1\}$, et
$d_4 = \{1, 1, -1, 1, 1, -1, -1, 1, 1, 1, -1, -1, 1, -1, -1\}$ etc.

Écrire une fonction `iter_dragon(d:list) -> list` qui effectue cette transformations pour une liste **d** de signes. Si **d** est de taille n, la fonction renverra donc une liste de signes de taille $2n + 1$.

4. Écrire une fonction `dragon(n:int) -> list` qui renvoie la liste correspondant à la séquence d_n.

5. La courbe du dragon de rang n est une courbe polygonale définie à la question 1 avec $z_k = \frac{\pi}{2} \times d_n[k]$. Écrire une procédure `Dragon(n:int) -> None` qui trace la courbe du dragon de rang n. Attention en testant cette fonction, au delà de $n = 15$, le tracé prendra du temps ! (10 min environ pour $n = 15$). Pour faire des test avec $n > 15$, il est recommandé de le faire en local depuis votre machine personnelle en ayant préalablement télécharger le fichier et python pour le lancer hors-ligne.



n=1 n=2 n=3 n=4 n=5 n=9