Nombres complexes
Problème de recherche : dans ou hors de l'ensemble de Mandelbrot ?

Pour un nombre complexe c, on construit la suite de nombres complexes définie par

$$z_0 = 0 \qquad \text{et} \qquad z_{n+1} = z_n^2 + c.$$

— Si les modules $|z_n|$ restent bornés, on dit que c appartient à l'ensemble de Mandelbrot, et on colorie en noir le point d'affixe c.
— On admet que si un terme vérifie $|z_n| > 2$, alors les modules suivants grandissent indéfiniment : c n'appartient pas à l'ensemble.

La mission

1. Pour chacun des nombres $c = -1$, $c = \mathrm{i}$, $c = 1$, $c = 1 + \mathrm{i}$ et $c = -2$, déterminer si c appartient à l'ensemble de Mandelbrot.

2. Un logiciel de dessin teste chaque pixel en calculant au plus 100 termes. Écrire l'algorithme (ou le programme Python) qu'il utilise.

Pour aller plus loin

Tester $c = 0{,}25$ et $c = 0{,}3$. Que remarque-t-on ? Pourquoi limiter le nombre de termes calculés peut-il conduire à une erreur ?

Coups de pouce

Bloqué ? Des indices progressifs sont disponibles en ligne. N'en ouvrir qu'un à la fois, et seulement après avoir cherché !

Éléments de correction

1. Les cinq points

— $c = -1$: $z_1 = -1$, $z_2 = 1-1 = 0$, $z_3 = -1$, $z_4 = 0\ldots$ La suite alterne entre 0 et -1 : **appartient**.

— $c = \mathrm{i}$: $z_1 = \mathrm{i}$, $z_2 = \mathrm{i}^2 + \mathrm{i} = -1 + \mathrm{i}$, $z_3 = (-1+\mathrm{i})^2 + \mathrm{i} = 1 - 2\mathrm{i} - 1 + \mathrm{i} = -\mathrm{i}$, $z_4 = (-\mathrm{i})^2 + \mathrm{i} = -1 + \mathrm{i}$, puis $z_5 = -\mathrm{i}\ldots$ La suite alterne entre $-1+\mathrm{i}$ (module $\sqrt{2}$) et $-\mathrm{i}$: **appartient**.

— $c = 1$: $z_1 = 1$, $z_2 = 2$, $z_3 = 5$: $|z_3| > 2$, **n'appartient pas**.

— $c = 1+\mathrm{i}$: $z_1 = 1+\mathrm{i}$, $z_2 = (1+\mathrm{i})^2 + 1 + \mathrm{i} = 2\mathrm{i} + 1 + \mathrm{i} = 1 + 3\mathrm{i}$, $|z_2| = \sqrt{10} > 2$: **n'appartient pas**.

— $c = -2$: $z_1 = -2$, $z_2 = 4 - 2 = 2$, $z_3 = 2\ldots$ La suite est constante égale à 2 (module 2, pas strictement supérieur) : **appartient** ; c'est le point le plus à gauche de l'ensemble.

2. L'algorithme

```
def mandelbrot(c, nmax=100):
    z = 0
    for n in range(nmax):
        z = z**2 + c
        if abs(z) > 2:
            return False    # le point s'échappe : hors de l'ensemble
    return True             # supposé dans l'ensemble

print(mandelbrot(-1), mandelbrot(1j), mandelbrot(1))   # True True False
```

Pour aller plus loin

$c = 0{,}25$: les termes croissent lentement vers 0,5 sans jamais dépasser 2 : **appartient**. $c = 0{,}3$: les termes croissent lentement, puis $|z_{12}| > 2$: **n'appartient pas**. Près du bord, un point peut s'échapper après plus de 100 termes : le logiciel le colorierait en noir par erreur. C'est pourquoi les images détaillées demandent beaucoup de calculs.

Ce qu'on a utilisé du chapitre

— Calcul algébrique dans $\mathbb{C}$: carré, somme ; $\mathrm{i}^2 = -1$.

— Module d'un nombre complexe ; suite définie par récurrence.

Pour aller plus loin (enrichissement)

L'ensemble de Mandelbrot (1980) est une *fractale* : son bord présente des détails infinis à toutes les échelles. Python gère directement les complexes : `1j` désigne i.