

DEVOIR MAISON D'INFORMATIQUE 1

Ensemble de Mandelbrot

Ce devoir maison est facultatif et non noté, et doit être pris comme une opportunité pour s'entraîner. Vous pouvez déposer votre travail (réalisé sur machine) sur cahier de prepa avant le 5 janvier. N'hésitez pas à me poser vos questions par mail (pompigne@crans.org) en cas de blocage.

1 Tracé de l'ensemble de Mandelbrot

Notre objectif est de tracer le célèbre ensemble fractal de Mandelbrot. Celui-ci est défini comme l'ensemble des points z du plan complexe tels que la suite (u_n) définie par $u_0 = z, u_{n+1} = u_n^2 + z$ a son module borné par 2.

Pour savoir si un point donné est dans l'ensemble, nous allons donc calculer les termes de cette suite et regarder si leur module dépasse 2. Bien entendu, on ne peut pas ainsi vérifier l'infinité des termes de la suite, nous allons donc nous limiter aux N premiers termes de la suite, avec N suffisamment grand pour avoir une bonne approximation de l'ensemble.

1. Importer les bibliothèques `numpy` et `matplotlib.pyplot`.
2. Définir une variable `N` de valeur 100.
3. Écrire une fonction `appartient` prenant en argument un nombre complexe z , renvoyant `True` si les N premiers termes de la suite (u_n) correspondante sont tous de module inférieur ou égal à 2, et `False` sinon.

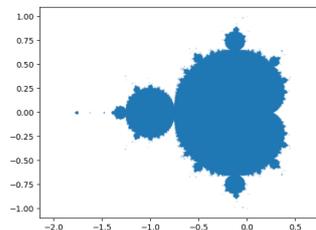
Un nombre complexe se manipule exactement comme un nombre flottant en Python. On peut obtenir son module en utilisant la fonction `abs`.

Les valeurs complexes se notent en Python sous la forme `a + bj`, où `a` et `b` sont des flottants. Vous pouvez tester votre fonction `appartient` sur les complexes suivants :

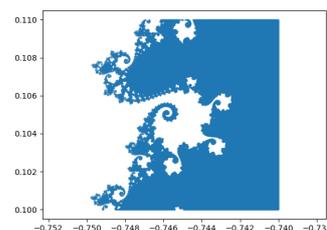
- `0` ; `1j` ; `-1+0.25j` doivent renvoyer `True`
 - `1` ; `-1 + 1j` ; `-0.75+0.2j` doivent renvoyer `False`
4. Nous allons à présent tracer l'ensemble de Mandelbrot entre les abscisses a et b , et les ordonnées c et d . Pour cela, on divise chacun de ces intervalles en 1000 avec la fonction `linspace` de `numpy`, puis pour chaque point (x, y) du rectangle ainsi formé, on considère le complexe $x + y * 1j$ et on l'ajoute aux points à tracer si `appartient` renvoie `True` dessus.

Écrire une fonction `afficher_mandelbrot` prenant en argument les quatre bornes `a`, `b`, `c`, `d` et réalisant ce tracé. On utilisera une instruction de la forme `plot(X, Y, ".", ms = 0.2)` pour tracer les points sous la forme de ronds fins, qu'on précédera par `axis('equal')` pour avoir la même échelle en abscisses et ordonnées.

`afficher_mandelbrot(-2,1,-1,1)` doit donner la figure suivante :



et `afficher_mandelbrot(-0.75,-0.74,0.1,0.11)` doit donner la figure suivante :



5. Diminuer la valeur de `N` et observer l'effet sur la figure affichée.

2 Tracé en couleur

L'affichage sans couleur de l'ensemble de Mandelbrot présente déjà une structure extrêmement riche, mais on peut souhaiter ajouter encore des détails en colorant les points en dehors de l'ensemble selon la vitesse à laquelle leur suite (u_n) associée dépasse 2 en module.

Pour cela, on définit le **rang** d'un complexe z comme le premier entier $k < N$ tel que $|u_k| > 2$, ou N si un tel k n'existe pas. Il ne reste plus qu'à afficher chaque point avec une couleur donnée par son rang.

1. Écrire une fonction **rang** prenant en argument un nombre complexe et renvoyant son rang.

Cette fonction doit en particulier passer les tests suivants :

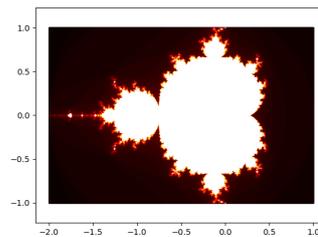
- 3 est de rang 0
- 2 est de rang 1
- 1 est de rang 2
- $-0.75 + 0.2j$ est de rang 15
- $-1 + 0.25j$ est de rang N

2. Ajouter la ligne `import matplotlib.cm as cm` au début du fichier pour importer des fonctions de gestion de couleur.

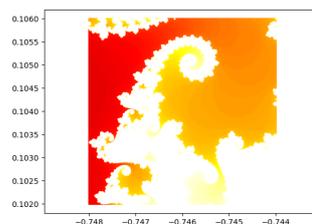
3. Écrire une fonction **afficher_mandelbrot_couleur** prenant les mêmes arguments que **afficher_mandelbrot**, et affichant chaque point du rectangle à afficher en fonction de son rang.

Pour cela, on créera pour chaque rang possible une liste $X[\text{rang}]$ des abscisses et une liste $Y[\text{rang}]$ des ordonnées des points de ce rang, puis on affichera ces points avec une instruction de la forme `plot(X[rang], Y[rang], ".", ms = 1, c = cm.hot(rang/N))`.

`afficher_mandelbrot_couleur(-2,1,-1,1)` doit afficher la figure suivante :



et `afficher_mandelbrot_couleur(-0.748,-0.744,0.102,0.106)` doit afficher :



4. Afficher votre partie préférée de l'ensemble de Mandelbrot.

