1 Définition/Propriétés utiles

Définition

Soit $c \in \mathbb{C}$ et (z_n) la suite complexe définie par :

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

Cette suite est appelée suite de Mandelbrot.
On note $\mathcal{M} = \{c \in \mathbb{C} \,/\, (z_n) \text{ est bornée}\}$. $\mathcal{M}$ est appelé **l'ensemble de Mandelbrot**.

Exercice 1 (*Premières propriétés*)

1. Montrer que si $c \in \mathbb{R}$ et que $c > \frac{1}{4}$ alors $c \notin \mathcal{M}$
2. Montrer que -2, -1, 0 et i appartiennent à $\mathcal{M}$.

Propriété

— Si $|c| > 2$ alors $c \notin \mathcal{M}$
— S'il existe n_0 tel que $|z_{n_0}| > 2$, alors la suite (z_n) diverge.

Démonstration : En classe.

2 Exploration avec Python

Il vous faut importer le module `cmath` pour travailler en complexe. Les nombres complexes s'écrivent sous la forme `a+bj`. Quelques exemples :

```
>>> from cmath import *
>>> z=1+2j
>>> z**2
(-3+4j)
>>> z.real,z.imag
(1.0, 2.0)
>>> abs(z),phase(z)
(2.23606797749979, 1.1071487177940904)
>>> exp(pi*1j)
(-1+1.2246467991473532e-16j)
>>> exp(pi*1j/2)
(6.123233995736766e-17+1j)
```

Exercice 2

1. Programmer une fonction `suite(c,N)` qui affiche N termes de la suite de Mandelbrot pour $z_0 = c$. L'affichage devra se faire numériquement et aussi graphiquement. Vous pouvez utiliser `plot.scatter(X,Y)` qui place dans le plan les points dont la liste des abscisses est X, et la liste des ordonnées est Y.

2. A l'aide de cet outil, observer attentivement le comportement de la suite pour les valeurs suivantes de c : $c = -1.01$, $c = 0.0099$, $c = 0.1 + 0.25i$, $c = -0.1 + 0.75i$ et $c = -0.1 + 0.9i$
3. Essayer pour $c = -0.23 + 0.78i$ et pour $N = 200$. Conclure...
4. Toujours pour $N = 200$, essayer pour $c = -0.12 + 0.46i$ puis pour $c = -0.123575077549388 + 0.46635536741660255.i$. Qu'en déduire ?

Exercice 3 (*Création d'une image de $\mathcal{M}$ en noir et blanc*)

Il va nous falloir décider, pour un c donné, si la suite (z_n) est bornée. En vertu du théorème 1, on se fixe un nombre d'itérations maximal noté `maxiter`. Si $|z_{maxiter}| < 2$ on décide que c est dans $\mathcal{M}$, sinon on décide que $c \notin \mathcal{M}$.

Pour générer une image, on procède de la manière suivante :

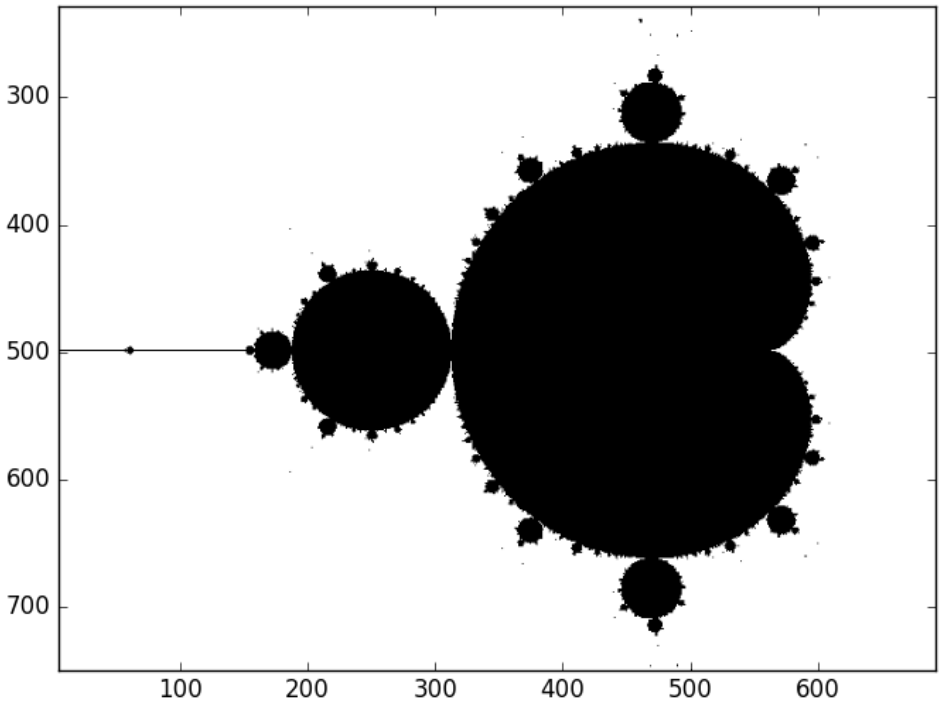
— On considère le cadran $[-2, 2] \times [-2, 2]$ sur lequel on "plaque" une matrice T de taille $N \times N$ qui sera notre matrice-image que l'on affichera avec `matplotlib`. Chaque pixel est un terme $T[k, l]$ de la matrice.

— A chaque pixel (k, l) on fait correspondre un point c du cadran (à vous de choisir lequel, le choix n'est pas unique).

— Au regard du critère précédent : si c est dans $\mathcal{M}$, on pose $T[k, l] = 0$ (noir), sinon on pose $T[k, l] = 1$ (blanc).

— On affiche la matrice T comme une image en noir et blanc avec `matplotlib`

Voici ce que vous devez obtenir pour $N = 1000$ et $maxiter = 300$:



Exercice 4 (*En couleur*)

On va affiner le programme précédent pour faire apparaître dans chaque pixel une couleur qui traduit la vitesse avec laquelle le module s'éloigne de 2.

Au lieu de renvoyer 0 ou 1 dans chaque pixel, vous renverrez $\frac{n_0}{maxiter}$ où $maxiter$ est un paramètre à fixer qui est en quelque sorte votre « niveau d'exploration », et n_0 est :

- Le premier rang entre 0 et $maxiter$ tel que $|z_{n_0}| > 2$ (s'il existe)
- $n_0 = maxiter$ si pour tout $n \leq maxiter$, $|z_n| \leq 2$.

Ainsi chaque pixel se verra attribuer un nombre entre 0 et 1 que `mathplotlib` convertira en une couleur.

On passera aussi les x_{min}, x_{max} , y_{min}, y_{max} en paramètre pour pouvoir zoomer.

Pour pouvoir lire les coordonnées sur les axes on rajoutera le code suivant :

```
ticks = np.linspace(0,M,11)
x_ticks = np.array([xmin + k*(xmax-xmin)/10 for k in range(11)])
plt.xticks(ticks,x_ticks)
y_ticks = np.array([ymax - k*(ymax-ymin)/10 for k in range(11)])
plt.yticks(ticks, y_ticks)
```

Voici quelques images que vous devez obtenir :

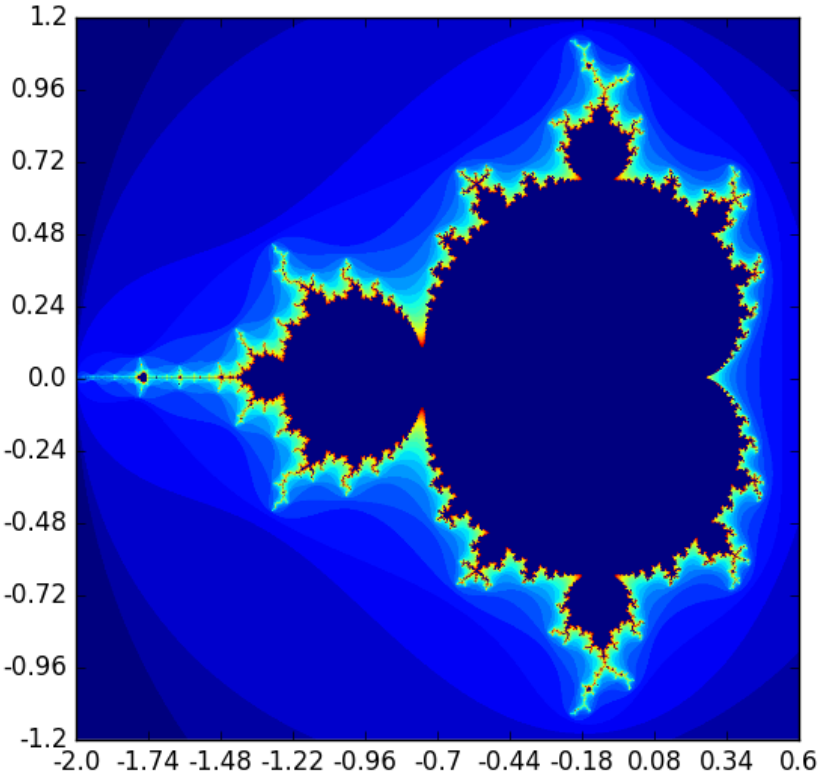


FIGURE 1 – $maxiter = 30$

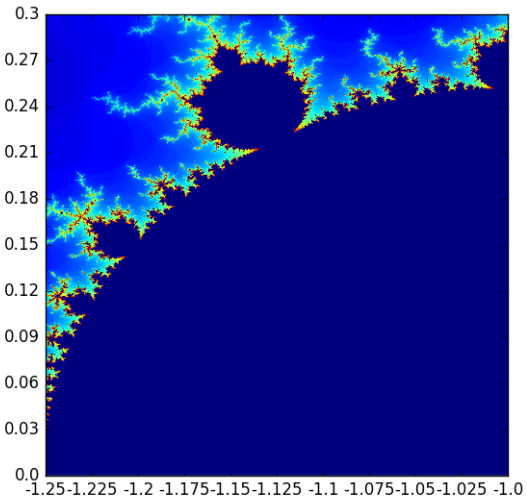


FIGURE 2 – $maxiter = 250$

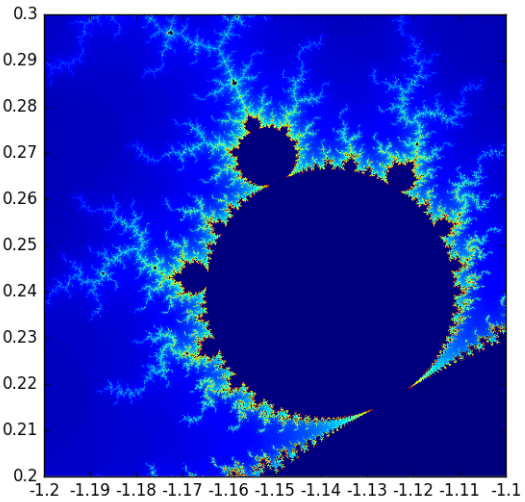


FIGURE 3 – Un bulbe

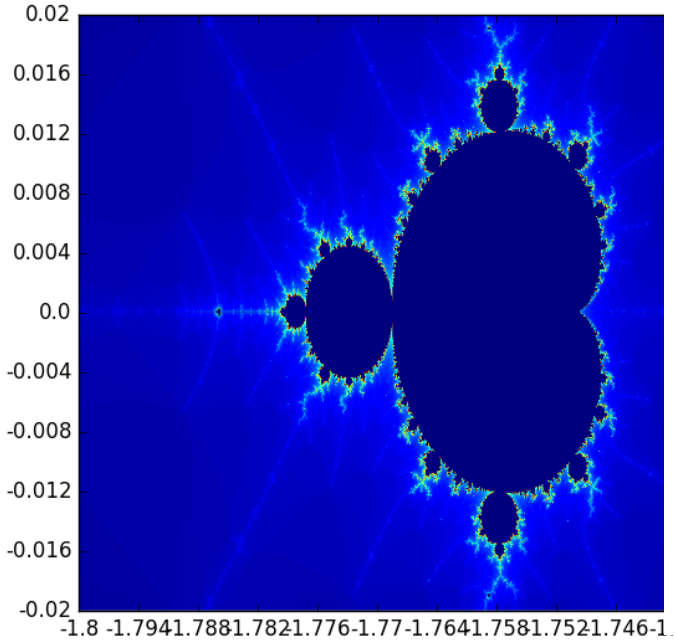


FIGURE 4 – Un ilôt - Non, ce n'est pas la même image que la première....

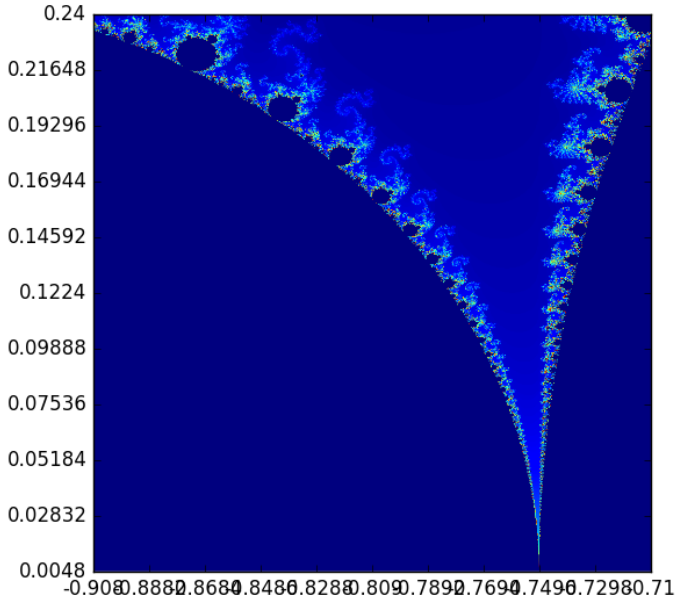


FIGURE 5 – La vallée des hyppocampes, $maxiter = 1000$