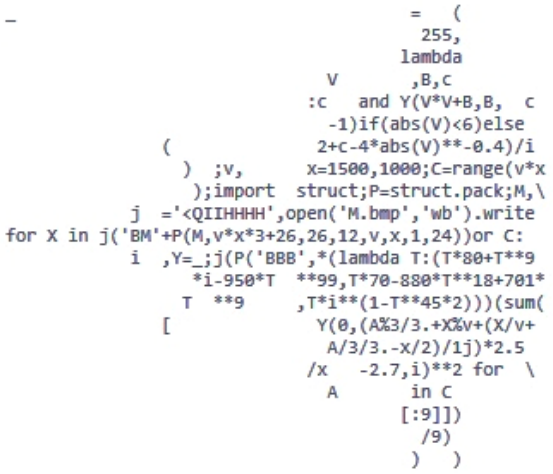
Chapter 1

Projet Fractales

L'ensemble de Mandelbrot et les ensembles de Julia sont des ensembles de nombres complexes. Leur étude s'effectue à l'aide des termes d'une suite numérique complexe.

1.1 Ensemble de Mandelbrot

Figure1: Illustration.



Benoit Mandelbrot était un mathématicien Franco-Polonais, au début des années 1980 il fut l'un des premiers à obtenir sur son antique IBM des représentations d'objets déjà étudiés par le passé: les **fractales**, figures géométriques auto-similaires découvertes par Gaston Julia et Pierre Fatou au début du XX^e siècle. On s'intéresse dans cette partie à l'ensemble qui porte son nom. Nous le noterons $\mathcal{M}$.

Les termes de cette suite sont définis par la relation de récurrence suivante:

$$\begin{cases} z_0 & = 0 \\ z_{n+1} & = z_n^2 + c \qquad c \in \mathbb{C} \end{cases}$$

Objectif: Vérifier si le nombre complexe c apparaissant dans cette relation appartient à $\mathcal{M}$.

Méthode (admise): Le nombre complexe c appartient à $\mathcal{M}$ si tous les termes de la suite (Z_n) qu'il génère ont un module $|z_n|$ inférieur ou égal à 2:

$$\forall n \in \mathbb{N}, |z_n| \leqslant 2$$

Autrement dit:

- Il s'agit de calculer **tous** les termes de la suite.
- Vérifier pour chacun si leur module est inférieur ou égal à 2.
- Si c'est le cas, on conclut: $c \in \mathcal{M}$.

Pour un aperçu (suffisant) sur les nombres complexes: :

- Paragraphe en annexe 1.3.1,
- https://www.youtube.com/watch?v=2GwSUDm_Rg8 .

1.1.1 PARTIE A : Etude d'exemple

Dans cette partie où vous justifierez tous les calculs à la main, on vérifie le module des cinq premiers termes de la suite en suivant la méthode de la page 1.

I-Le nombre i On teste le nombre i, la suite à étudier est donc:

$$\begin{cases} z_0 & = 0 \\ z_{n+1} & = z_n^2 + i \end{cases}$$

1. Remplir le premier tableau A.1 ci-dessous.
2. Le nombre i appartient-il à $\mathcal{M}$?

II-Le nombre 1+i On teste le nombre 1+i, la suite à étudier est donc:

$$\begin{cases} z_0 & = 0 \\ z_{n+1} & = z_n^2 + 1 + i \end{cases}$$

1. Remplir le second tableau A.II ci-dessous.
2. Le nombre complexe 1+i appartient-il à $\mathcal{M}$?

Tableau A.I

n	z_n	$\lvert z_n \rvert$	condition $\lvert z_n \rvert < 2$
0	0	0	VRAIE
1			
2			
3			
4			
5			

Tableau A.II

n	z_n	$\lvert z_n \rvert$	condition $\lvert z_n \rvert < 2$
0	0	0	VRAIE
1			
2			
3			
4			
5			

1.1.2 PARTIE B : premier algorithme

Dans le langage **Python**, un nombre complexe z=a+ib se déclare par z=complex(a,b). Par exemple, le nombre z=1+2i se déclare en tapant z=complex(1,2).

1. Ecrire un algorithme à la main prenant en paramètre un nombre complexe c et retournant un booléen indiquant si ce nombre appartien à $\mathcal{M}$.
2. Ecrivez ce prédicat est_dans_M(c) en *Python* et le tester avec i et 1+i.
3. Compléter ce jeu de test avec 0,5+0,5i et 0,2i+0,1 et 0.

1.1.3 PARTIE C : affichage

L'objectif est d'obtenir une représentation de l'ensemble de Mandelbrot en noir et blanc.

> **Méthode**:
> Chaque pixel de l'écran sera associé à un nombre complexe et colorié en noir s'il est à l'intérieur de $\mathcal{M}$, en blanc sinon.

Il faudra donc tester chaque pixel du canvas.

1. On remplace le nombre c saisi au début de l'algorithme par une boucle parcourant des nombres complexes d'un domaine. On choisit pour travailler comme valeurs minimales:

   - $x_{min} = -2$ et $y_{min} = -1,1$.
   - x_{max}=0,6 et y_{max}=1,5.

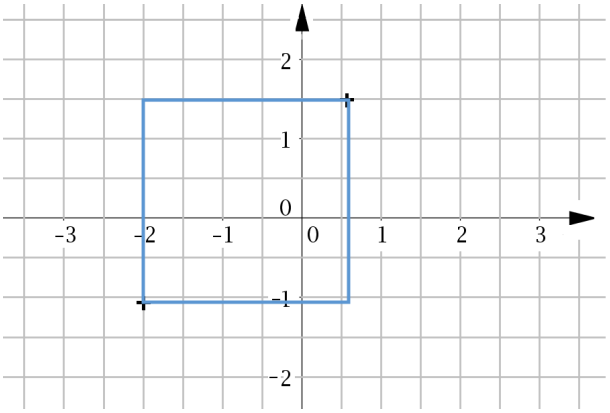
   et un canvas carré de 500 pixels de coté.

   (a) Compléter l'algorithme suivant pour obtenir un parcours des pixels en abscisse.

   ```
   pour i allant de 1 à 500
       abscisse⟵-2+i*.....
   fin pour
   ```

   (b) Expliquer alors l'algorithme utilisé ci-dessous, on obtient le domaine parcouru sur la figure suivante.

   ```
   pour i allant de 1 à 500
     pour j allant de 1 à 500
       abscisse⟵ −2 + i×.....
       ordonnée⟵ .....+j×.....
     fin pour
   fin pour
   ```

2. Ecrire une procédure parcours(largeur, hauteur) coloriant en noir un pixel si ses coordonnées sont dans l'ensemble $\mathcal{M}$.

   On pourra dans le programme principal créer un canvas d'une fenêtre *tkinter* avec un canvas de taille 500×500 et un bouton quitter.

1.1.4 PARTIE D:En couleur

Dans cette partie on passe à la couleur. Vous aurez besoin pour cela des deux fonctions dec2hex() et rvb2html() qui convertissent les couleurs RVB en HTML. Voir TP tkinter.

1. Ecrire une fonction convergence(c) prenant en paramètre un nombre c, elle renvoie l'indice du premier terme de la suite pour lequel le module est supérieur à 2.

2. Ecrire une procédure couleur(a, b, n) coloriant en couleur le pixel de coordonnée (a; b) en une couleur dépendant de n.

1.2 Ensemble de Julia

Le principe reste le même mais c est fixé et on fait varier z_0.Il existe donc une infinité d'ensemble de *Julia*, chacun correspondant à une valeur de c. Modifiez l'algorithme précédent pour la valeur de c=-0,123+0,745i. Cet ensemble de Julia est appelé *lapin de Douady*.

1.3 ANNEXE

1.3.1 Nombres complexes

On travaille avec la notion de nombre complexe. Ces nombres ont été imaginés pour résoudre certains problèmes algébriques où écrire un carré négatif était nécessaire. Il a donc fallu inventer un nouvel ensemble de nombre: $\mathbb{C}$.

Propriétés:

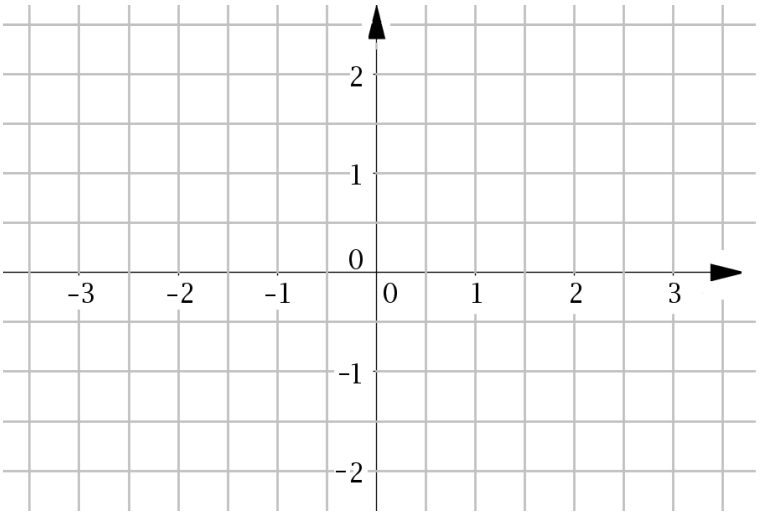
- Le nombre noté i est tel que $i^2 = -1$.
- Tout nombre complexe peut s'écri re sous la forme $a + ib$ où a et b sont des réels respectivement appelés partie réelle et partie imaginaire.
- Les calculs s'effectuent comme avec les réels, il suffit de veiller à transformer i^2 en -1.
- On peut placer dans un plan les point dont les coordonnées sont a et b. a+ib est alors appelée *affixe*, elle correspond aux coordonnées.
- Le module de a+ib noté $|a + ib|$ est la distance entre le point d'affixe $a + ib$ et le centre du repère. **Donner la formule permettant de calculer $|a + ib|$.**

Exemple On admet que les régles de calcul avec ces nombres sont les mêmes qu'avec les nombres réels, en prenant en considération l'égalité $i^2 = -1$. Pour chaque résultat, donnez la partie réelle et la partie imaginaire.

1. $3i^2 = -3$.
2. $1 - 3i + 5i - 2 = -1 + 2i$ ou $2i - 1$.
3. Calcul de $i \times (1 + i)$:

$$i(1+i) = i \times 1 + i \times i = i + i^2 = i - 1$$

4. Placer les points d'affixes respectives i; $1+i$; -3; $2i-1$; $i-1$ et $-0,5+2i$ dans ce repère.



1.3.2 Couleurs et tkinter

Le système de couleur *tkinter* est détaillé ici:

https://html-color-codes.info/Codes-couleur-HTML/

Il s'agit ici de convertir un code couleur RVB du type (x,y,z) avec $0 \leqslant x \leqslant 255$, $0 \leqslant y \leqslant 255$, $0 \leqslant z \leqslant 255$ en code couleur HTML du type que comprend tkinter. Chaque code HTML contient le symbole "#" et 6 lettres ou chiffres. Ces nombres sont sous forme hexadécimale. Par exemple "$\#FF$" en hexadécimal représente le nombre 255 en décimal.