

Pré-rapport de Projet

Stephanie NOEL et Sid BERRAF

10 octobre 2011

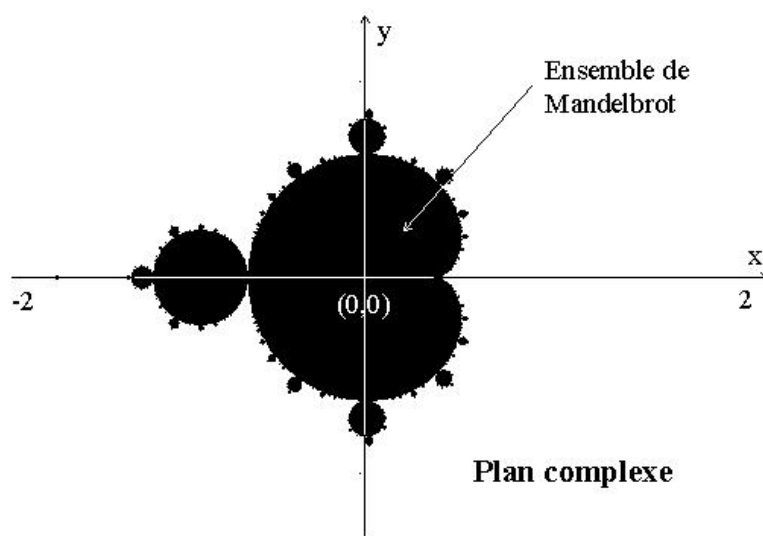
Table des matières

1 Introduction

2 Fractale de Mandelbrot

2.1 Définition mathématique

Pour définir mathématiquement la fractale de Mandelbrot on associe à chaque point M du plan complexe, la suite $z_{n+1} = z_n^2 + c$ avec $z_0 = 0$ et $c = x + iy$ l'affixe du point M . Tous les points pour lesquels la suite est bornée vont constituer l'ensemble de Mandelbrot. Or la suite diverge lorsque $\|z\| \leq 2$, donc l'ensemble est inclus dans un cercle de rayon 2 et centré à l'origine du plan.



2.2 Informatique

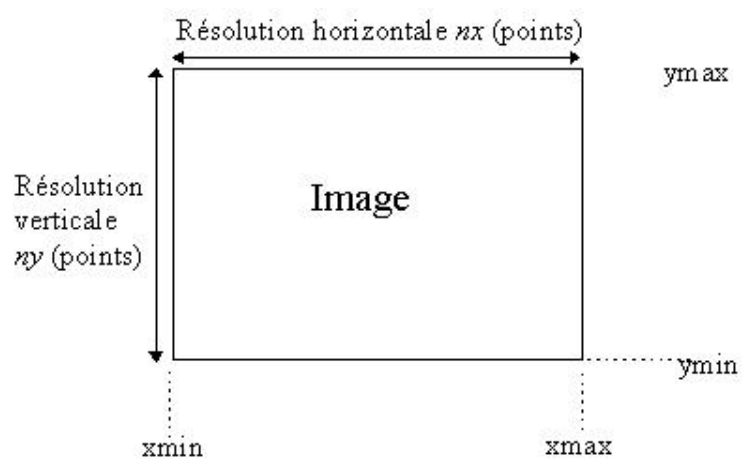
En pratique, on prends la grille de pixels de la fenetre. Pour chaque pixels on calcul la valeur de c , dont les parties imaginaire et réelle sont données avec les coordonnées du pixel. On considère la transformation du plan complexe définie par $z \mapsto z^2 + c$ avec le c calculé. Soit, partant de $z_0 = 0$ on construit la suite z_n avec $z_{n+1} = z_n^2 + c$.

Calculer z_{n+1} revient à doubler l'argument de z_n et à élever son module au carré puis de rajouter c . Ainsi, on obtient un nouveau point et on répète les même opérations jusqu'a que l'on trouve un point qui s'envole vers l'infini.

Si la suite des modules de ces nombres complexes ne tend pas vers l'infini, on noircit le point associé à c . Autrement, on colorie ce point d'une couleur correspondant au nombre de termes qu'on a dû calculer avant d'obtenir un module supérieur ou égal à 2. L'ensemble de Mandelbrot sera la figure formée par les points c tels que la suite z_n ne tend pas l'infini quand n tend vers l'infini.

2.3 Valeur de c

Il faut tout d'abord se donner les limites dans le plan complexe de l'image et choisir une résolution (nx ny).



Ensuite on définit c par $c = a + ib$ avec $\begin{cases} a = i \cdot pas + x_{min} \\ b = y_{max} - j \cdot pas \end{cases}$

2.4 Calcul de z_{n+1}

Sachant que z_n est un complexe, calculer z_n^2 n'est pas facile dans un programme d'où le cheminement suivant :

$$z_{n+1} = z_n^2 + c, \text{ on pose } \begin{cases} z_n = x_n + iy_n \\ c = a + ib \end{cases}$$

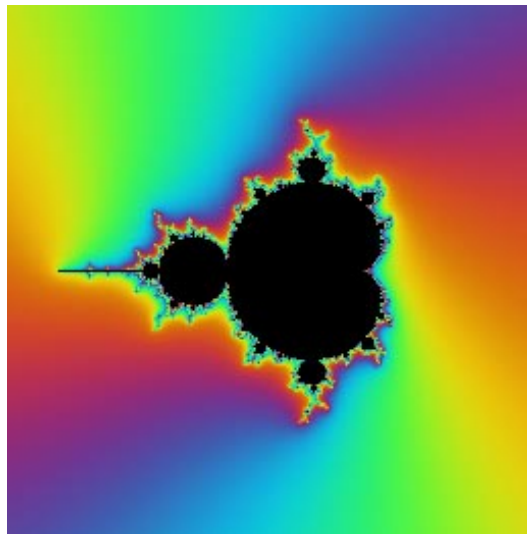
$$\begin{aligned} \Rightarrow z_{n+1} &= (x_n + iy_n)^2 + a + ib \\ \Rightarrow z_{n+1} &= x_n^2 + 2iy_nx_n - y_n^2 + a + ib \\ \Rightarrow z_{n+1} &= (x_n^2 - y_n^2 + a) + i(2y_nx_n + b) \end{aligned}$$

$$\text{On pose } \begin{cases} x_{n+1} = x_n^2 - y_n^2 + a \\ y_{n+1} = 2y_nx_n + b \end{cases} \text{ et on obtient } z_{n+1} = x_{n+1} + iy_{n+1}$$

2.5 Organigramme

2.6 Algorithme

2.7 Dessin



3 Fractale de Julia

3.1 Définition mathématique

La fractale de Julia est basée sur la même suite complexe que celle de Mandelbrot soit la suite $z_{n+1} = z_n^2 + c$ sauf que l'on ne fait plus varier c mais z_0 . Donc c devient une constante complexe arbitraire. La suite étant la même mathématiquement parlant, elle divergera toujours lorsque $\|z\| \leq 2$. Il y a une infinité d'ensembles de Julia puisque l'on peut donner n'importe quelle valeur à la constante c .

3.2 Informatique

En pratique, on prends la grille de pixels de la fenetre. On choisit un point sur cette grille. Ce point sera la valeur de c . Pour chaque pixels on calcul la valeur de z_0 , dont les parties imaginaire et réelle sont données avec les coordonnées du pixel. On considère la transformation du plan complexe définie par $z \mapsto z^2 + c$. Soit, partant de z_0 calculé, on construit la suite z_n avec $z_{n+1} = z_n^2 + c$. Calculer z_{n+1} revient à doubler l'argument de z_n et à élever son module au carré puis de rajouter c . Ainsi, on obtient un nouveau point et on répète les même opérations jusqu'a que l'on trouve un point qui s'envole vers l'infini.

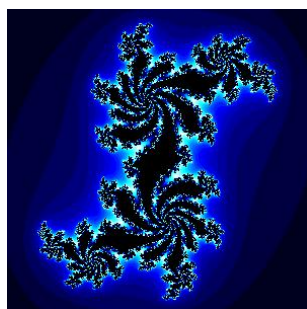
Si la suite des modules de ces nombres complexes ne tend pas vers l'infini, on noircit le point associé à z_0 . Autrement, on colorie ce point d'une couleur correspondant au nombre de termes qu'on a dû calculer avant d'obtenir un module supérieur ou égal à 2. L'ensemble de Mandelbrot sera la figure formée par les points z_0 tels que la suite z_n ne tend pas l'infini quand n tend vers l'infini.

Les calculs de z_{n+1} et z_0 sont similaire avec les calculs de z_{n+1} et c de l'ensemble de Mandelbrot.

3.3 Organigramme

3.4 Algorithme

3.5 Dessin



4 Fractale IFS

4.1 Définition mathématique

Un IFS est un système de fonctions itérées qui lorsqu'on l'applique à un objet géométrique permet d'obtenir une fractale. Ces fonctions sont affines et contractantes.

Le terme contractant signifie que lorsque l'on applique ces fonctions à une figure, les points de celle-ci ne s'éloignent pas mais se rapprochent ; il n'y a donc pas de divergence, on obtient une figure contenue dans un espace fini.

La figure fractale obtenue occupe donc une surface finie, qui n'est autre que la surface de l'objet initial, pour un contour de longueur infinie.

4.1.1 Transformation affine

Une transformation affine est définie par un vecteur de translation V et une matrice de transformation T . Pour obtenir les coordonnées du point M' image d'un point M par cette transformation, on applique la formule vectorielle :

$$OM' = T.OM + V$$

Si (x', y') et (x, y) représentent les coordonnées de M' et M , respectivement, on peut traduire cela par le système d'équations suivant :

$$\begin{cases} x' = ax + by + e \\ y' = cx + dy + f \end{cases}$$

où a, b, c et d sont les coefficients de la matrice T et e et f les composantes du vecteur V .

4.1.2 Opérateur d'Hutchinson

Généralement les IFS sont décrits à l'aide de plusieurs fonctions f_1, \dots, f_n .

Pour une image initiale A donnée, on lui applique chacune de ces fonctions donnant ainsi n images. La nouvelle image $F(A)$ est alors constituée de la réunion de ces n images :

$$F(A) = f_1(A) \cup \dots \cup f_n(A) \quad (1)$$

F est appelé l'opérateur d'Hutchinson ; c'est l'itération de cet opérateur qui détermine l'IFS. Pour une image initiale A_0 on obtient une suite d'images en répétant F .

$$A_{k+1} = F(A_k), k \in N \quad (2)$$

Un IFS génère une suite qui tend vers une image finale A_∞ que l'on appelle attracteur de l'IFS.

4.2 Loi de probabilité pour la génération d'IFS

Puisqu'à chaque étape il faut calculer la transformée d'une image par N applications et réaliser l'union des résultats, cela nécessite énormément de calcul. C'est pourquoi il existe une méthode plus rapide.

Associons à l'IFS défini par $\{w_i\}_{i=1,\dots,N}$ un jeu de probabilités $\{p_i\}_{i=1,\dots,N}$. p_i est la probabilité pour que la variable aléatoire discrète k prenne la valeur i . Considérons alors la suite de points de \mathbb{R}^2 définie par :

$$\begin{cases} r_0 \\ r_n = w_{k_n} r_{n-1} \end{cases}$$

où $\{k_n\}_{n=1,\dots,\infty}$ est une suite de variables aléatoires discrètes, indépendantes, identiquement distribuées selon les probabilités $\{p_i\}_{i=1,\dots,N}$. On peut alors montrer que l'ensemble des points $\{r_n\}$ converge vers l'attracteur de l'IFS.

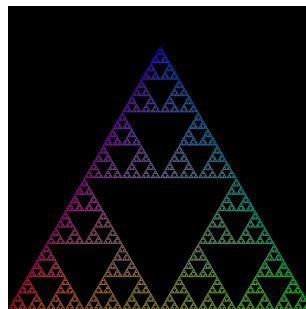
4.3 Informatique

En pratique, on prends la grille de pixels de la fenetre. Pour chaque pixels on associe ses coordonnées au point (x, y) . Pour un certain nombre d'itérations choisit par l'utilisateur, on lui associe une transformation décidée selon la loi de probabilité pour obtenir le point suivant (x_{n+1}, y_{n+1}) à traiter.

4.4 Organigramme

4.5 Algorithme

4.6 Dessin



5 Fractale L-system

5.1 Définition

Un L-système est une grammaire formelle, c'est-à-dire un ensemble de symboles et de règles, qui définissent la procédure pour transformer par itération une chaîne de caractères de départ en une autre chaîne.

Les transformations sont basées sur des règles qui spécifient comment des caractères ou parties de la chaîne de caractères sont remplacés par d'autres. Si on applique cet ensemble de règles de façon récursive à une chaîne de caractère, une nouvelle chaîne ayant une structure fractale peut être créée.

Un L-system est défini par :

- Un alphabet
- Un ensemble de symboles constants
- Un axiome
- Des règles de reproduction

5.2 Informatique

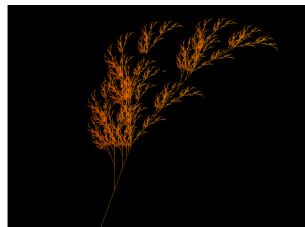
5.2.1 Notre L-system

Symbole	Action
a	Avancer en traçant
b	Avancer sans tracer
+	Rotation horaire de cycle
-	Rotation anti-horaire de cycle
	rotation de 180 degres
?	Mémorisation de la position actuelle
!	Retour à la dernière position mémorisée

5.3 Application

En pratique on écrira pour chaque symbole une fonction qui effectue l'action qui lui ai associé et qui renvoie le point d'arrivée. Chacune de ses fonctions prendra en paramètre la longueur du tracé, le dernier point tracé, l'angle par rapport au dernier tracé, Le programme principal prendra en paramètre l'axiome de départ sous forme d'une chaîne de caractère, parcourera chaque caractère en lui

5.4 Dessin



6 Fractale de type flamme

6.1 Dessin

7 Conclusion