

# Des fractales en *Crocodile BASIC* (IV)

## 1. Introduction

Dans cet article, nous détaillons les algorithmes utilisés dans *Crocodile BASIC*, pour colorier les figures fractales. Les notions exposées dans le premier article de cette série seront supposées connues.

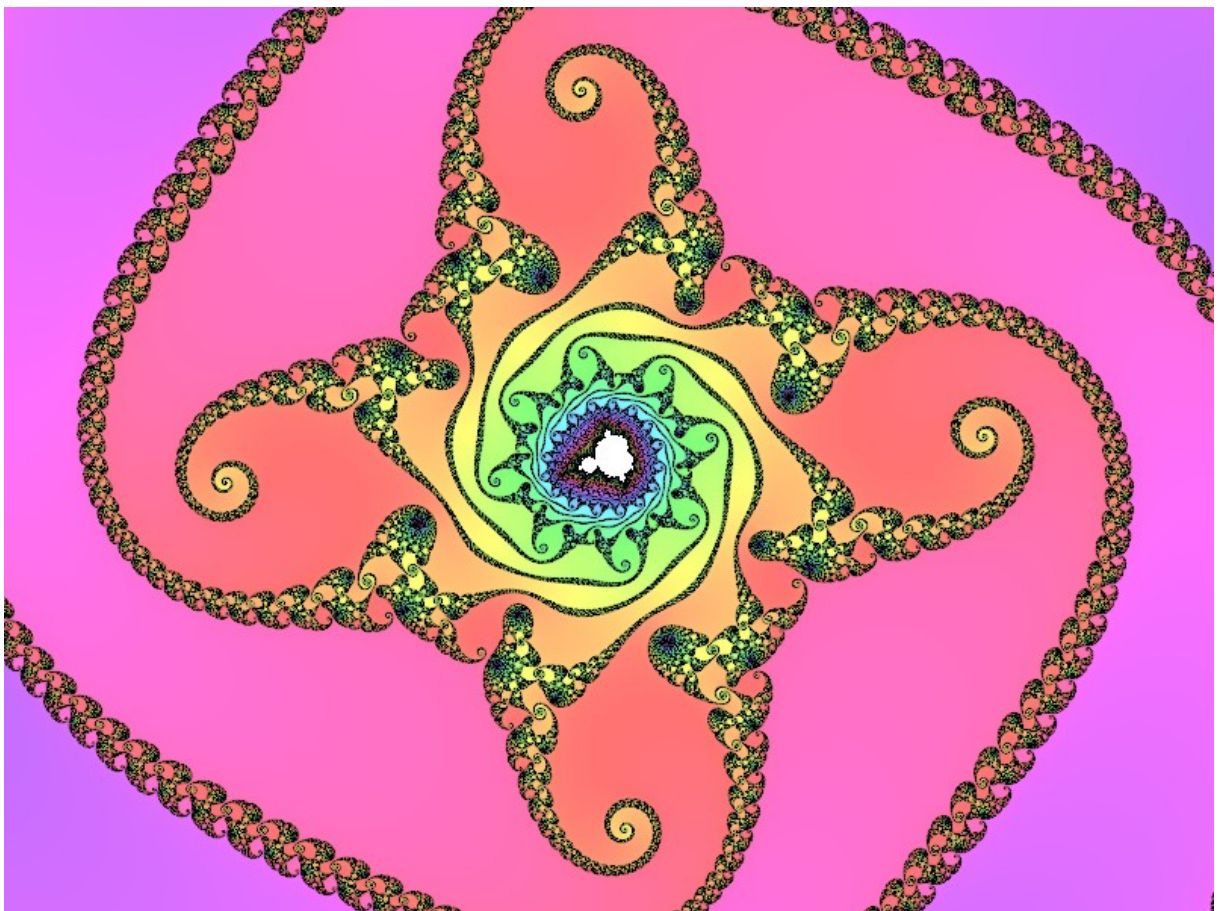
## 2. Le programme

Nous prendrons comme exemple le programme `mandel.bas` décrit dans l'article n° 1, et plus particulièrement la figure `elephant_05` définie par la procédure suivante :

```
init_params "elephant_05",0.26502185290119,0.00302941971904756,10000,1000000000,-1,-8
```

Les deux derniers paramètres, `DistFact` (= -1) et `ColorFact` (= -8), contrôlent la couleur de chaque pixel, selon le système HSV : `DistFact` contrôle la luminosité ( $V = Value$ ) ; `ColorFact` contrôle la teinte ( $H = Hue$ ) et la saturation ( $S$ ).

Avec les réglages précédents, l'image obtenue est la suivante :



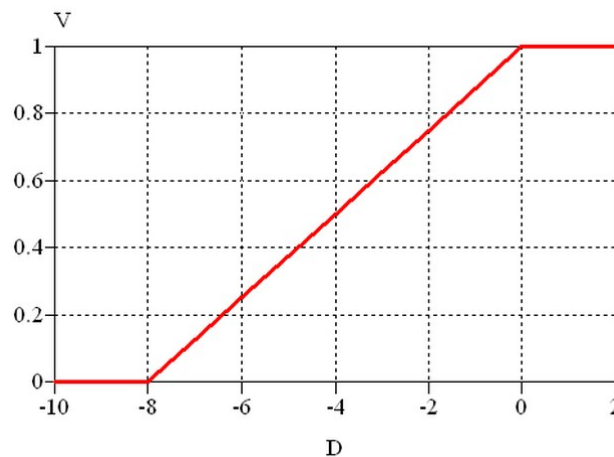
### 3. Les algorithmes de coloration

Les algorithmes qui nous intéressent sont contenus dans les deux fonctions `mdbc01` et `rgbc01`, que nous allons étudier maintenant.

#### 3.1. La luminosité : Fonction `mdbc01`

Cette fonction détermine la luminosité  $V$  du point considéré puis appelle la fonction `rgbc01` pour calculer la teinte  $H$  et la saturation  $S$ . Rappelons qu'il s'agit d'une version simplifiée d'un [algorithme](#) décrit par Robert Munafo.

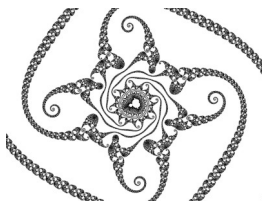
La luminosité  $V$  est déterminée par la variable `dscale` (ici notée  $D$ ) qui dépend à la fois de l'estimateur de distance et de la valeur de `DistFact` fixée par l'utilisateur ;  $V$  varie en fonction de  $D$  selon la courbe suivante :



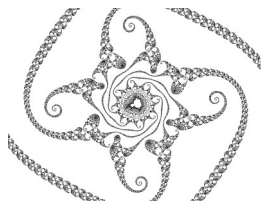
Les faibles valeurs de  $D$  correspondent aux points les plus proches de l'ensemble de Mandelbrot : la luminosité est alors minimale, ce qui permet de tracer la frontière de l'ensemble en noir. La luminosité augmente à mesure que l'on s'éloigne de l'ensemble.

Le paramètre `DistFact` permet de modifier la valeur de  $D$  pour obtenir des images plus claires (`DistFact`  $>$  0) ou plus sombres (`DistFact`  $<$  0).

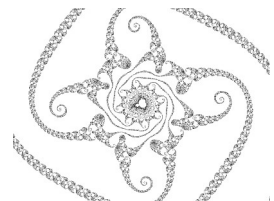
Les figures suivantes montrent l'effet de l'augmentation de `DistFact` de -2 à +2 sur les images en noir et blanc (obtenues en fixant à zéro la valeur de `ColorFact`).



`DistFact` = -2



`DistFact` = 0

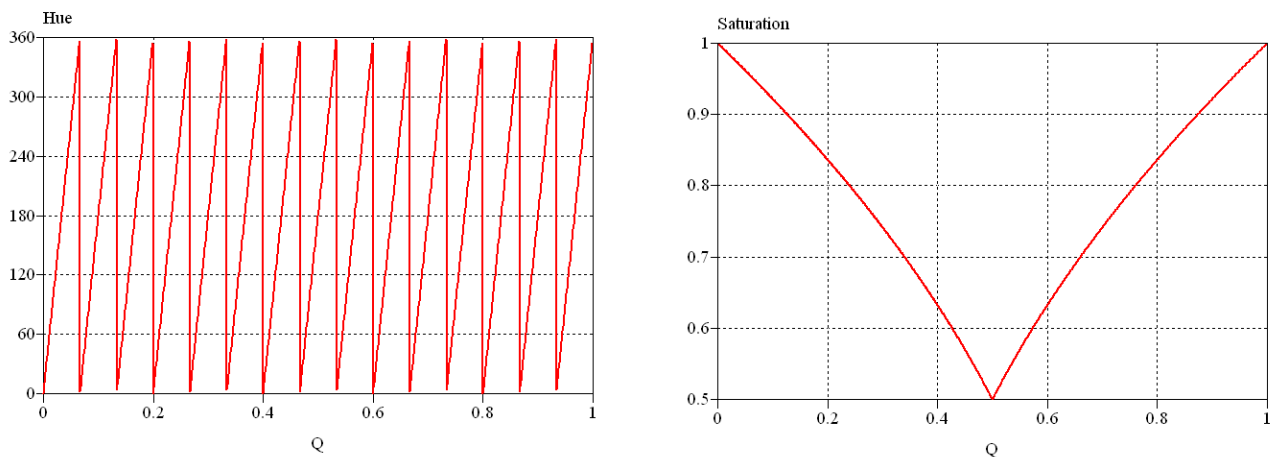


`DistFact` = +2

### 3.2. La teinte et la saturation : Fonction `rgbcol`

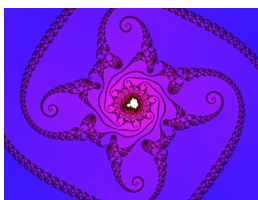
Ces deux valeurs sont déterminées par une variable **q** (définie dans `mdbc01`) qui dépend à la fois du *continuous dwell* et du paramètre **ColorFact** (pris en valeur absolue, soit **AbsCol**). Ce dernier fonctionne comme un pourcentage. Il permet de calibrer la valeur de **q** pour la situer dans l'intervalle [0, 1]. Pour une image donnée, **q** varie habituellement sur une gamme de quelques dixièmes.

Connaissant **Q** le programme calcule deux variables **angle** et **radius** pour définir les valeurs de H et S sur le disque de couleurs. Après quelques transformations supplémentaires, on obtient des valeurs de H et S qui varient en fonction de **q** selon les courbes suivantes :

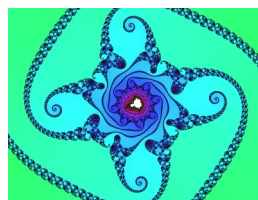


Pour la teinte (H), un spectre complet correspond à une variation  $\Delta q \approx 0,066$ . En ajustant la valeur de **ColorFact** on peut amplifier ou réduire la valeur de  $\Delta q$  pour avoir une gamme de couleurs plus ou moins étendue.

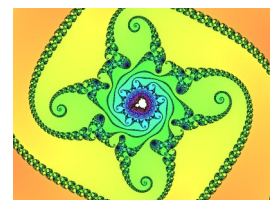
Les figures suivantes montrent l'effet de la variation de **ColorFact** de -4 à -6 sur les images en couleurs (obtenues en fixant **DistFact** à -2). La valeur **ColorFact** = -8 correspond à l'image du § 2.



**ColorFact** = -2



**ColorFact** = -4



**ColorFact** = -6

Pour la saturation (S), les valeurs calculées sont toujours supérieures à 0,5 (pour éviter d'avoir des images trop sombres) et varient selon une fonction racine carrée plutôt que linéaire.

Si **ColorFact** est positif, on assombrit une bande sur deux (selon que le nombre d'itérations **iter** est pair ou impair) en diminuant à la fois **V** et **S (radius)**. La figure suivante montre l'effet obtenu avec **ColorFact** = +8. On constate que non seulement les bandes « impaires » sont assombries, mais que la luminosité globale de l'image est diminuée (suite à la diminution de **V**).



#### **4. Conclusion**

Cet article nous a permis de détailler les algorithmes utilisés dans *Crocodile BASIC* pour tracer les figures fractales. Nous en verrons d'autres exemples dans les prochains articles de cette série.