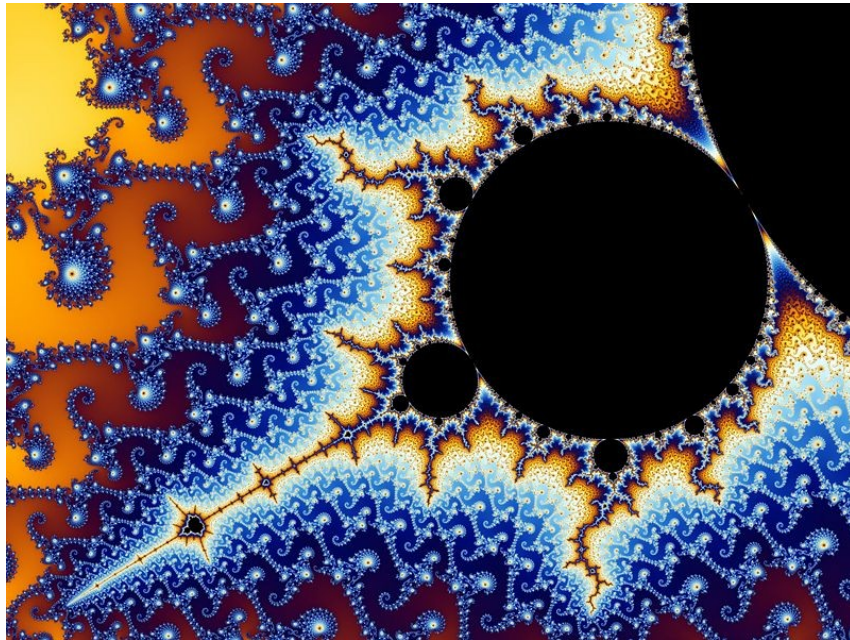


Projet de Physique P6
STPI/P6/2015 – 047

Analyse des propriétés d'objets fractals 2D et 3D



Étudiants :

Claire MOLINIER

Josselin BOST

Quentin VALLEE

Victor BONIOU

Betty CROCHEMORE

Malak TAJ

Sandratra RASENDRASOA

Enseignant-responsable du projet :

Jérôme YON

Date de remise du rapport : 12/06/2015

Référence du projet : STPI/P6/2015 – 047

Intitulé du projet : Analyse des propriétés d'objets fractals 2D et 3D.

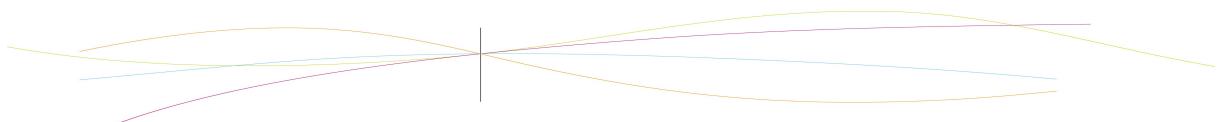
Type de projet : *calcul et modélisation.*

Objectifs du projet : L'objectif premier de notre projet était de découvrir les objets fractals et de comprendre la notion de dimension fractale. Pour cela, la finalité du projet consistait à générer un objet fractal « nouveau » et à mettre en œuvre une méthode pour analyser cet objet.

Mots-clefs du projet : *objet fractal, dimension fractale, modélisation, méthode des boites.*

TABLE DES MATIÈRES

| | |
|---|----|
| Introduction..... | 5 |
| Méthode et organisation du travail..... | 6 |
| 1.Qu'est ce qu'une fractale ?..... | 7 |
| 1.1.Définitions | 7 |
| 1.2.Quelques exemples | 8 |
| 2.Génération d'un objet fractal..... | 9 |
| 2.1.Cas général..... | 9 |
| 2.2.Génération d'une courbe de Koch..... | 10 |
| 2.3.Génération d'un « nouvel » objet fractal..... | 12 |
| 3.Détermination de la dimension fractale..... | 14 |
| 3.1.Comment définir la dimension fractale ?..... | 14 |
| 3.2.La méthode des boîtes..... | 16 |
| 3.2.1.Théorie de la méthode des boîtes..... | 16 |
| 3.2.2.Application de la méthode des boîtes ImageJ..... | 17 |
| 3.2.3.Descriptif du logiciel..... | 17 |
| 3.2.4.Macros et fonctions..... | 17 |
| 3.2.5.Principe de la macro BoxCounting()..... | 18 |
| 3.2.6.Comparaison de l'algorithme utilisé avec Fraclac..... | 18 |
| 3.2.7.Perspectives..... | 19 |
| 3.3.Analyse de notre objet..... | 19 |
| Conclusion..... | 21 |
| Bibliographie..... | 22 |
| Annexes..... | 23 |
| Programmes réalisés..... | 23 |
| Documents divers..... | 27 |



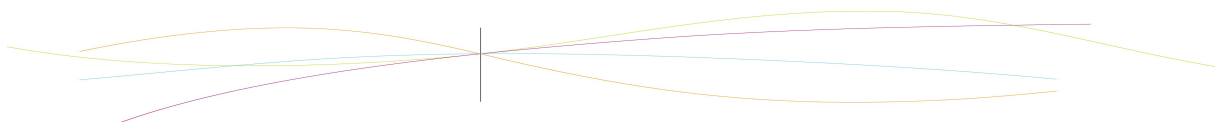
Introduction

Les objets fractals sont, pour le grand public, des créations belles mais un peu étrange, réalisées par des mathématiciens et des informaticiens. En effet, il suffit de faire une recherche dans un moteur de recherche pour trouver des images presque psychédélics illustrant les fractales. Et excepté pour qualifier ces images, il est rare que le terme « fractale » soit connu et utilisé à bon escient.

Cette expression a été inventée en 1974, par Benoit Mandelbrot, dans son livre « Les Objets fractals – Forme, hasard et dimension », il y développe la théorie de la rugosité. Cette théorie énonce qu'un objet fractal est similaire quelque soit l'échelle à laquelle on l'observe. Cette idée est généralement celle qui vient à l'esprit, lorsque l'on parle de fractale. Il est intéressant de noter que la nature crée des objets qui suivent cette définition. Ainsi, la fougère, ou certaines espèces de choux, tel que le chou de Romanesco, présentent des caractères qui sont similaires à grande et petite échelle. Même nos poumons ou notre système sanguin suivent des propriétés fractales, la trachée se scinde en deux conduits, qui se rescinde chacun en deux, etc.

Nos connaissances, au début du projet, ne dépassaient pas ce point. Ce projet nous a donc permis d'en apprendre beaucoup sur les fractales, tout en nous ouvrant à d'autres connaissances, principalement informatiques. En effet, lors de ce projet, nous avons utilisé de façon très importante les logiciels Scilab et Imagej.

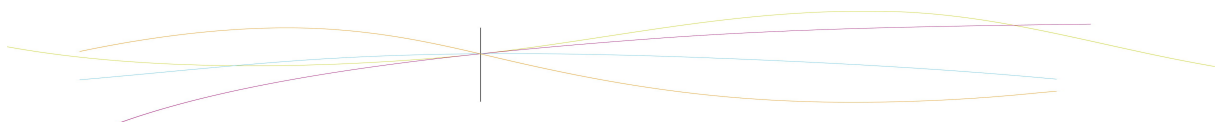
Ce rapport est donc un compte rendu qui va revenir sur les connaissances que nous avons accumulées sur les fractales et sur le travail que nous avons effectué. C'est à dire : la création d'objets fractals grâce à des algorithmes codés avec le logiciel scilab et l'étude d'objet fractal grâce à la méthode des boites réalisé sous Imagej. Ce rapport sera conclu par l'analyse de nos résultats et des perspectives d'évolution sur la base de ces résultats.



Méthode et organisation du travail

Lors de ce projet, Monsieur Yon nous a laissé libres quant au choix du sujet sur le thème des fractales, nous devions seulement ne pas balayer tout le sujet et prendre un exemple précis pour le traiter à « fond ». Dans un premier temps, nous avons donc dû nous familiariser avec le sujet, les fractales. En effet, nous n'avions presque aucune connaissance sur ce thème. Nous nous sommes donc rendus compte que le sujet était très vaste et très « mathématique ». Nous avons alors décidé de diviser le sujet en deux parties, une première partie mathématique et modélisation d'objet fractal puis une seconde partie axée sur les applications physiques des fractales. Nous avons alors fait de nombreuses recherches pour comprendre les lois mathématiques qui régissent ces objets et pour trouver des applications physiques concrètes. Nous avons alors décidé de nous concentrer sur un mur antibruit. Ce mur, grâce à une forme fractale, permettrait de diminuer considérablement le passage des ondes sonores. Nous pensions générer un objet fractal de la même forme que le mur antibruit puis tester l'efficacité du mur, suivant la dimension fractale ou le nombre d'itérations, en comptant le nombre de « rebond » de l'onde sur le mur. Nous avons alors soumis notre projet à Monsieur Yon qui a estimé que cela serait difficilement réalisable. En effet, nous n'étions déjà pas sûrs de savoir construire nous même un objet fractal, et même une fois fait, nous aurions beaucoup de problèmes à simuler l'envoi d'une onde sur l'objet.

Suite à cela, nous sommes repartis de zéro et nous avons essayé de trouver un sujet plus abordable dans le temps qui nous était imparti. Nous avons décidé de conserver l'idée de faire deux groupes pour que ce soit plus simple pour travailler. Nous avons abandonné l'idée de faire une application physique car cela s'apparentait trop à de la recherche documentaire, ce qui nous intéressait moins. Nous avons quand même conservé l'idée de générer un objet fractal de nous même, puis nous avons décidé de créer un programme pour retrouver la dimension d'un objet qu'on sait fractal, suivant la méthode des boîtes. Nous nous sommes alors séparés en deux et chaque groupe a commencé à travailler sur son sujet. Dans le premier groupe, composé de Betty Crochemore, Claire Molinier, Victor Boniou et Quentin Vallée, nous avons décidé de travailler sur la génération d'un objet fractal, l'étoile de Koch. Pour cela, nous avons décidé de travailler avec le logiciel scilab puisqu'il apparaissait comme le plus adapté pour générer un dessin. Dans le second groupe, composé de Malak Taj, Sandratra Rasendrasoa et Josselin Bost, nous avons décidé de travailler sur la méthode des boîtes. Cependant, nous avons choisi de travailler sur le logiciel imageJ et de changer la résolution des images pour trouver la dimension fractale d'un objet, et plus particulièrement de celui fourni par le premier groupe.



1. QU'EST CE QU'UNE FRACTALE ?

1.1. Définitions

Un objet fractal peut se définir de plusieurs manières différentes. La première apparition du mot fractal date de 1974, cette notion a été introduite par Benoît Mandelbrot, dans *La théorie de la rugosité*. Il explique qu'« une fractale désigne des objets dont la structure est invariante par changement d'échelle. ». La définition d'un objet fractal serait alors un dessin qui se répète à l'infini. En effet, la particularité d'un objet fractal est son autosimilarité, c'est à dire que peu importe où on zoom sur l'objet, on retrouvera toujours le même motif. Pour se rendre compte de ce principe, on peut observer un chou romanesco. En effet, ce légume présent dans la nature est un des objets fractals les plus connus. Quand on regarde le chou dans son ensemble, on n'observe pas immédiatement un schéma qui se répète. Cependant, quand on observe le dessin plus attentivement, on remarque que le même motif se répète à différentes échelles.

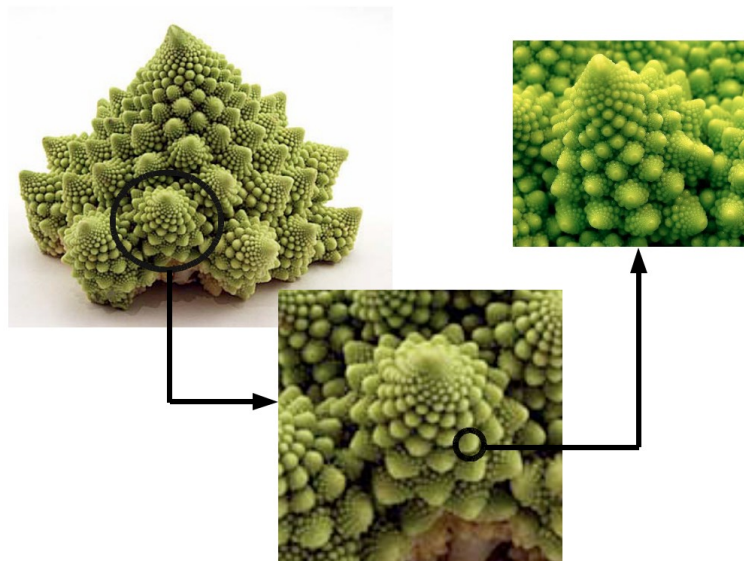
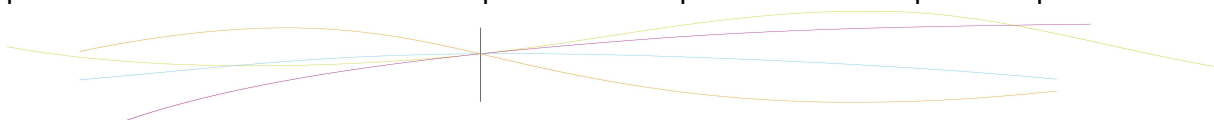


Illustration 1: Chou romanesco.

Cependant, on peut aussi définir un objet fractal comme un objet mathématique « dont la création ou la forme ne trouve ses règles que dans l'irrégularité ou la fragmentation. », Larousse. On comprend alors que l'autosimilarité du dessin peut s'expliquer par des règles mathématiques complexes telle que l'homothétie interne. Ce terme compliqué est l'explication mathématique de l'autosimilarité. En effet, une homothétie peut se définir par la formule $\vec{AM'} = k \cdot \vec{AM}$ où AM est un vecteur représentant l'image que l'on veut reproduire et k, le facteur de réduction/augmentation souhaité. Il permet donc de reproduire le même motif, segment par segment, d'un objet fractal.

De plus, par définition, pour qu'un objet soit fractal, il faut qu'il suive une loi de puissance en fonction des différents paramètres du problème comme par exemple le nombre



d'itération, le périmètre ou encore l'aire du dessin. Cette loi nous permet ainsi de connaître la dimension fractale de l'objet. On obtient donc la formule : $d = \frac{\ln(N)}{\ln(\text{echelle})}$ avec N le paramètre étudié. Nous reviendrons en détail sur cette notion dans la suite du rapport.

Les objets fractals sont très nombreux et différents, on en retrouve certains dans la nature alors que d'autres sont simplement des dessins avec une forme spécifique.

1.2. Quelques exemples

L'ensemble de Mandelbrot a été découvert par Gaston Julia et Pierre Fatou au début du XXème siècle. Cependant, ce n'est qu'en 1980 qu'il est pour la première fois représenté. Benoit Mandelbrot arrive à le visualiser par ordinateur et présente ses recherches sur les polynômes quadratiques, qui permettent de créer cette figure fractale. Malgré la représentation de l'ensemble, les travaux de Mandelbrot restent vagues et, ce n'est que 4 ans plus tard, que deux mathématiciens Adrien Douady et de John H. Hubbard établissent les propriétés fondamentales de cet objet.

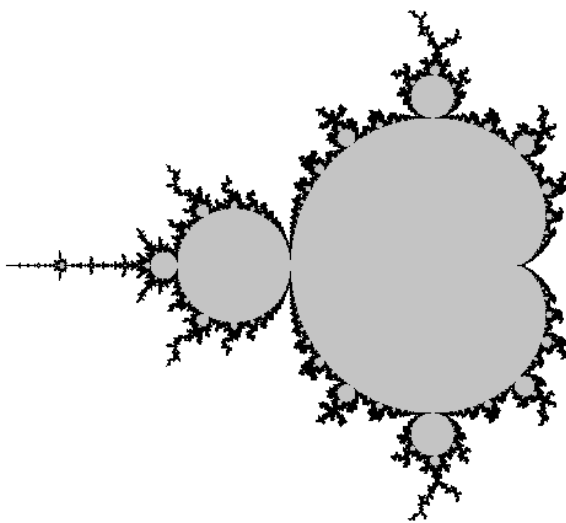
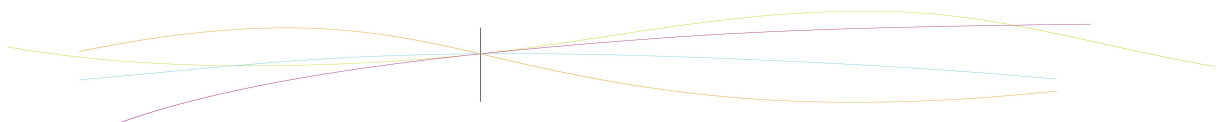


Illustration 2: Ensemble de Mandelbrot

L'ensemble de Mandelbrot se définit comme l'ensemble des points c du plan complexe pour lesquels la suite complexe

$$\begin{aligned} z_0 &= 0 \\ z_{n+1} &= z_n^2 + c \end{aligned} \text{ est bornée.}$$

Cependant, les fractales ne sont pas que des objets théoriques, ils sont aussi présents dans la nature. L'un des exemples les plus connus est la géométrie de la côte bretonne. En effet, on peut observer que selon l'échelle qu'on prend, la côte bretonne peut avoir une taille différente. En faisant varier cette échelle et en mesurant la longueur de la côte pour chacune d'elle, on obtient une loi de puissance. Cela prouve donc que la côte bretonne a une géométrie fractale. Cette forme spécifique a de nombreuses répercussions, notamment sur la force des vagues ! En effet, une force sera d'autant plus puissante que la surface sur laquelle elle s'exerce sera petite. Or, nous avons vu que la forme fractale de la côte bretonne permettait d'avoir une surface très grande. Cela explique alors que les vagues soient peu fortes en Bretagne.



Nous avons donc bien vu qu'il existe différents objets fractals dans la nature alors que d'autres sont des objets purement mathématiques générés sur ordinateur. On peut alors se demander si on peut créer des objets fractals simples ? Et si cela est possible comment s'y prendre.

2. GÉNÉRATION D'UN OBJET FRACTAL

2.1. Cas général

Dans un premier temps, nous allons prendre l'exemple de la construction d'un objet fractal 3D : l'éponge de Menger. Elle est aussi appelée éponge de Menger-Sierpinski et est une extension du tapis de Sierpinski en 3 dimensions. Elle fut décrite pour la première fois par le mathématicien autrichien Karl Menger en 1926.

L'éponge de Menger a pour forme initiale un cube de côté a . A la première itération, on découpe le cube initial en 27 cubes égaux de côté $a/3$. Puis, on enlève tous les cubes « centraux », c'est à dire qu'on ne garde que les arêtes du cube initial, il ne nous reste donc plus que 20 cubes. Ensuite, on répète cette opération dans chaque nouveau cube. Le volume diminue donc au cours des itérations alors que l'aire augmente. Théoriquement, on pourrait obtenir un objet d'aire infinie et de volume nul ! Le volume diminue de $\frac{20}{27}$ à chaque itération donc il vaudra $\left(\frac{20}{27}\right)^n V_0$ à la n -ième itération. Au contraire, l'aire augmente de $\frac{4}{3}$ à chaque itération.

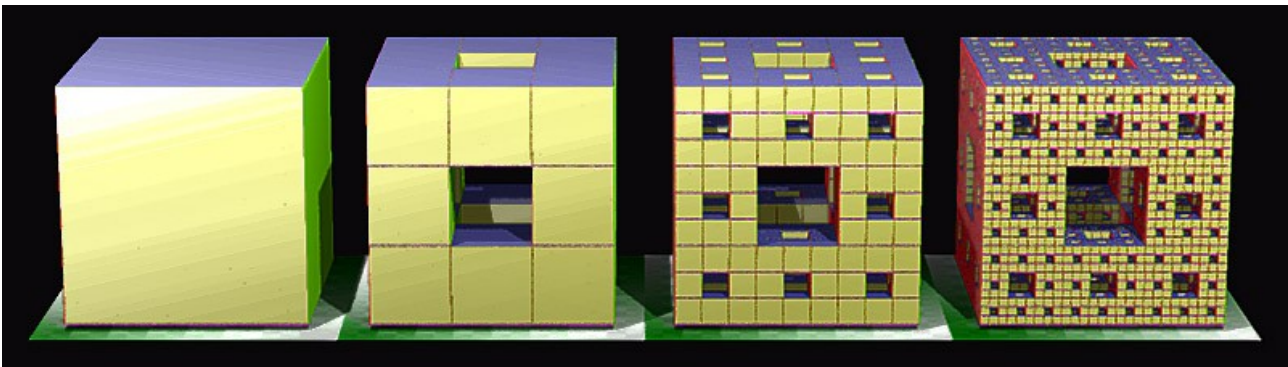
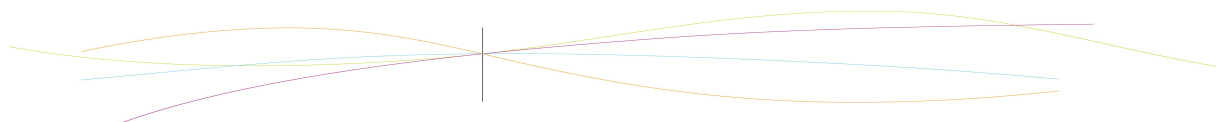


Illustration 3 : premières itérations de la construction d'une éponge de Menger.

Dans un second temps, prenons l'exemple du flocon de Koch. Cette forme fractale est l'une des premières courbes fractales à avoir été décrite. Elle a été inventée par le mathématicien suédois Helge von Koch en 1904, c'est à dire avant même l'apparition du terme fractal !

La construction d'un flocon de Koch est très simple. Initialement, la figure n'est qu'un triangle équilatéral de périmètre défini. Ensuite, à l'itération 1, on divise chaque segment en 3



et au centre de chacun, on construit un nouveau triangle équilatéral de côté $\frac{1}{3}$ du segment. A chaque itération on répète la même opération. Ainsi, si le périmètre initial est de 1, à l'itération n le périmètre sera de $\left(\frac{4}{3}\right)^n$.

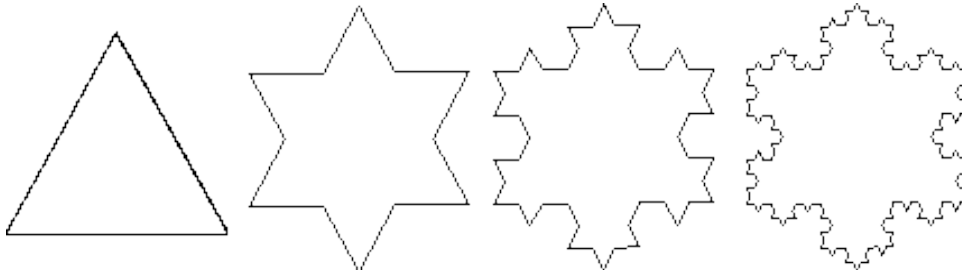


Illustration 4: premières itérations de la construction du flocon de Koch

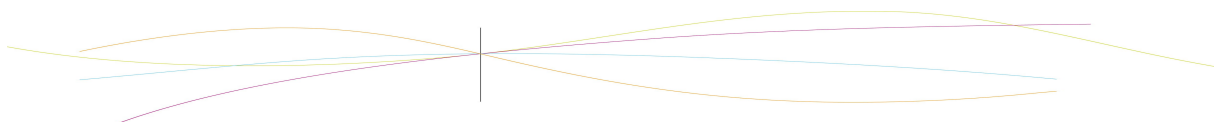
Nous venons donc de voir deux formes fractales faciles à construire à la main. C'est pour cela que nous avons choisi de générer le flocon de Koch pour notre projet.

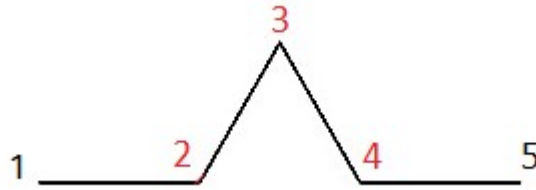
2.2. Génération d'une courbe de Koch

Dans un premier temps, nous avons décidé de générer une courbe de Koch. La courbe de Koch est un côté du flocon de Koch que nous venons de voir. Nous avons décidé de faire notre propre programme pour générer une courbe de Koch grâce au logiciel graphique Scilab. Nous avons vu dans la première partie comment créer cet objet. Ces étapes sont plutôt simples et faciles à reproduire à la main, seulement on a alors la limite de l'échelle, nous ne pouvons pas dessiner des objets trop petits. C'est pourquoi l'utilisation d'un logiciel graphique est intéressante, puisque nous pouvons faire un grand nombre d'itérations et zoomer sur la courbe de Koch pour vérifier l'auto similarité.

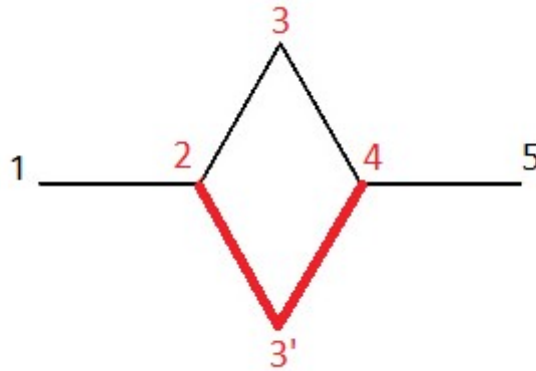
Pour cela, nous avons créé un programme avec le logiciel Scilab (disponible en annexe). Le principe est le suivant :

- L'appel de la fonction de la courbe de Koch se fait en donnant les coordonnées des deux points du segment de base, n représente le nombre d'itération et c représente le pourcentage de chance d'avoir la nouvelle itération à l'intérieur ou à l'extérieur de la courbe, car il est possible d'apporter un caractère aléatoire à la courbe de Koch.
- Ensuite, nous cherchons tous les points à relier pour former la figure de l'itération suivante, on a donc besoin de 5 points comme ci-dessous.





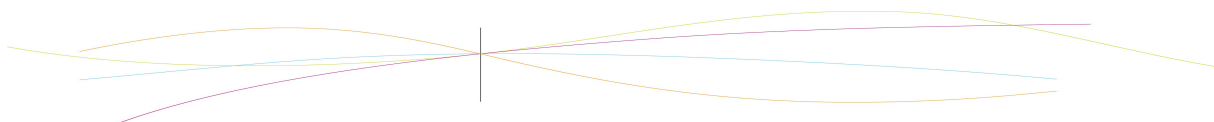
- Pour déterminer si la nouvelle itération se fera à l'intérieur ou à l'extérieur de la courbe, il suffit de changer le point 3, on aura une valeur positive ou négative du coefficient a , qui déterminera si le point 3 se trouvera au-dessus ou en-dessous du segment. On aura donc deux configurations possibles avec le point 3 ou 3' :



Si l'on modifie la valeur c en entrée de la fonction, on modifie alors la probabilité d'avoir l'une ou l'autre des configurations, à $c=0,5$ on a autant de chance d'avoir un point en haut qu'en bas, pour un $c=0$ on aura une courbe de Koch ordinaire et pour $c=1$ on aura la courbe de Koch retournée.

- Une fois que l'on a compris comment marche une itération, on refait la même chose pour chaque segment formé. Ici, on utilisera la récursivité, qui nous évitera d'avoir trop de points à stocker (pour une itération on a déjà 5 points à stocker, pour 2, cela monte à 17 points et ainsi de suite sans récursivité !).

- Enfin, il faut relier les points calculés, mais nous n'allons le faire que lorsque $n=0$, c'est à dire qu'à l'échelle la plus petite rencontrée. Si l'on ne rajoute pas cette condition, le programme va nous tracer en plus de la courbe de Koch, tous les traits intermédiaires de construction que nous ne désirons pas voir apparaître.



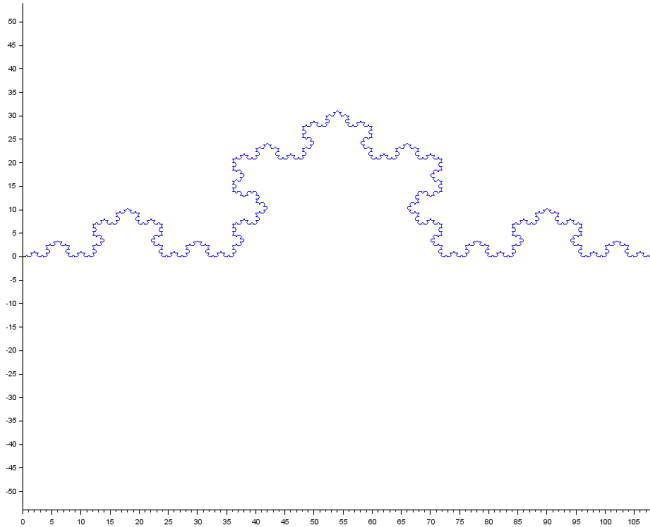


Illustration 5: Courbe de Koch classique.

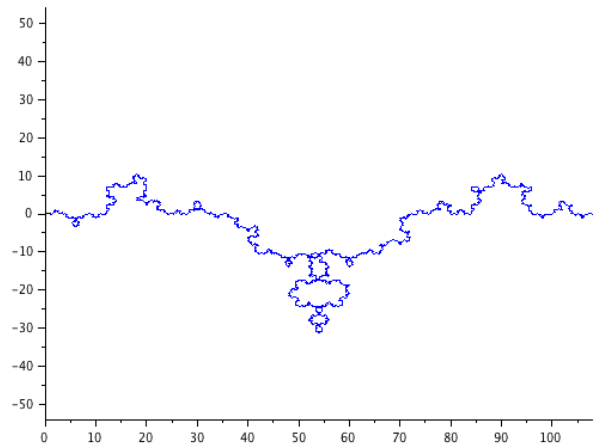
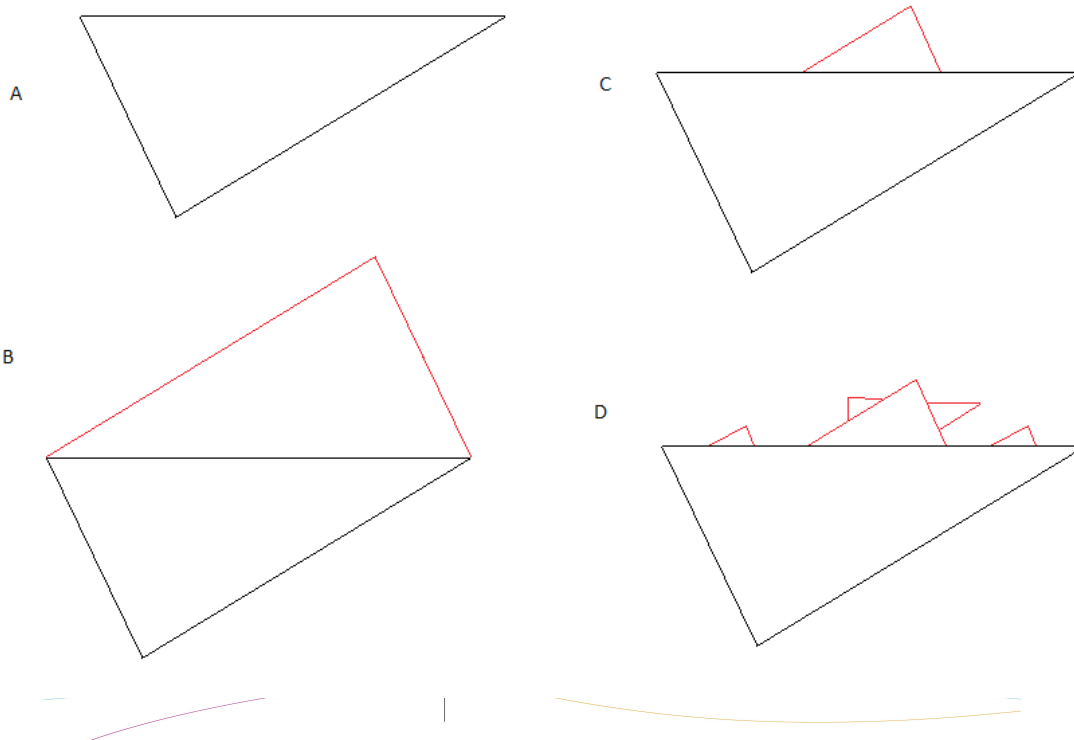


Illustration 6: Courbe de Koch aléatoire.

On a observé qu'en ajoutant de l'aléatoire, les courbes de Koch prennent des formes très diversifiées et beaucoup plus chaotique, qui peuvent rappeler le contour d'une côte maritime par exemple. Au premier abord, il est difficile d'imaginer qu'elles ont été générées en suivant un algorithme très précis et autosimilaire. On peut donc en conclure que l'aspect fractale d'une forme ou d'un objet n'est pas forcément évident.

2.3. Génération d'un « nouvel » objet fractal

Dans un second temps, nous avons décidé de générer un flocon de Koch, il nous a suffit de créer 3 courbes de Koch en forme de triangle. Avec ce changement, nous voulions

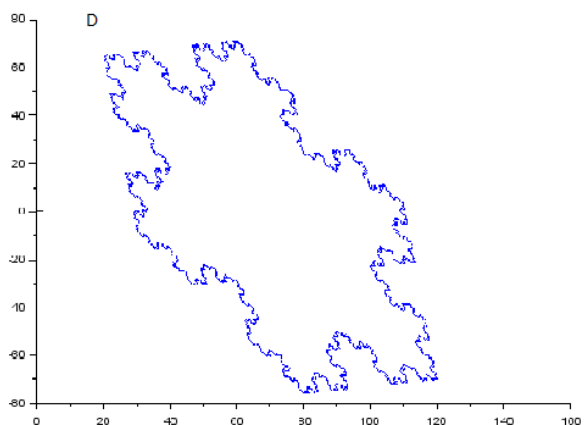
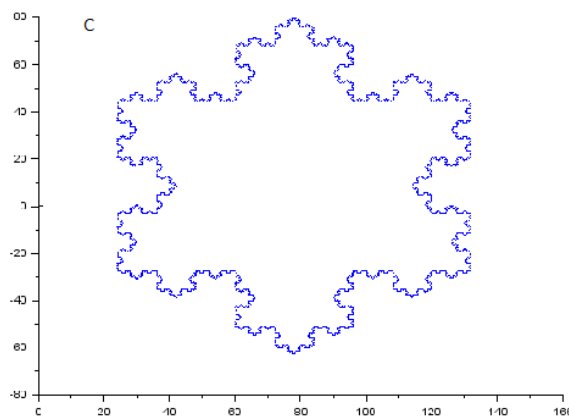
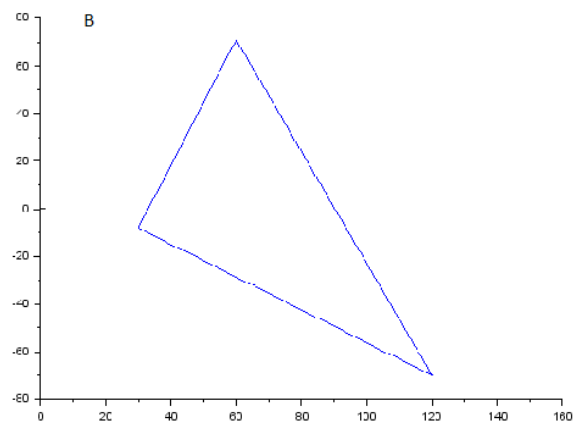
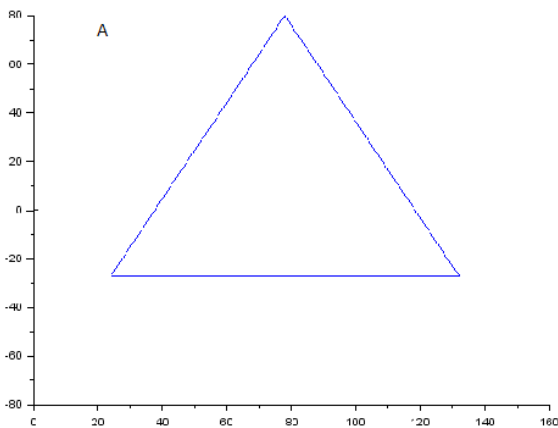


voir s'il était possible de générer un « nouvel » objet fractal, c'est à dire ne plus prendre un triangle équilatéral comme forme initiale mais un triangle quelconque.

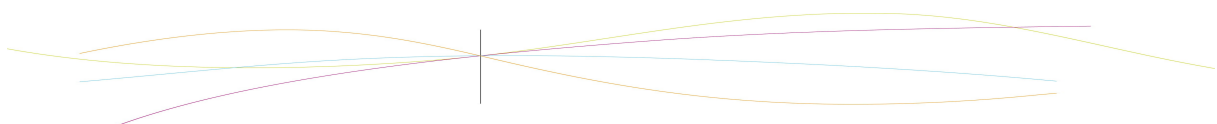
Les étapes des itérations sont les suivantes :

- Nous partons de la figure A, qui est un triangle quelconque.
- Nous traçons la symétrie centrale du triangle par rapport au côté sur lequel nous faisons l'itération, ici le plus grand côté, ce qui nous donne la figure B.
- Ensuite, nous faisons une réduction du nouveau triangle formé par un rapport de 3, ce qui forme la figure C.
- Puis on réitère ces mêmes étapes sur chaque nouveau segment formé, l'itération 2 est représentée par la figure D.
- On refait également les mêmes étapes pour chaque côté du triangle.

Le principe de construction reste donc exactement le même que pour le flocon de Koch, seule la forme de l'objet initial change. Sur la figure suivante, on peut donc observer la différence entre le flocon de Koch classique et notre objet.



À gauche, on observe un triangle équilatéral et la généralisation du flocon de Koch à l'itération 4. Alors qu'à droite, on observe un triangle quelconque et la généralisation du flocon de Koch à l'itération 4. Nous voyons donc que l'objet fractal que nous avons créé est



une généralisation du flocon de Koch car si l'on suit les étapes avec un triangle équilatéral, nous obtenons le flocon de Koch. Un triangle quelconque nous donne un flocon de Koch renversé.

Le code utilisé est à peu près similaire à celui de la courbe de Koch (utilisation de la récursivité, choix des itérations...).

Dans cette partie, nous venons donc de voir comment générer un « nouvel » objet fractal. Cependant, nous ne sommes pas encore sûr que ce soit un objet fractal ! En effet, l'autosimilarité du dessin ainsi que sa construction très similaire au flocon de Koch nous font penser que c'est un objet fractal mais nous ne l'avons pas encore prouvé !

3. DÉTERMINATION DE LA DIMENSION FRACTALE

Pour prouver que notre objet est fractal, nous devons vérifier qu'il suit une loi de puissance, c'est à dire que nous devons déterminer sa dimension fractale. Pour cela, nous avons décidé de mettre en place une méthode de détermination de la dimension d'un objet fractal : la méthode des boîtes. Dans un premier temps, nous allons vous définir la dimension fractale d'un objet.

3.1. Comment définir la dimension fractale ?

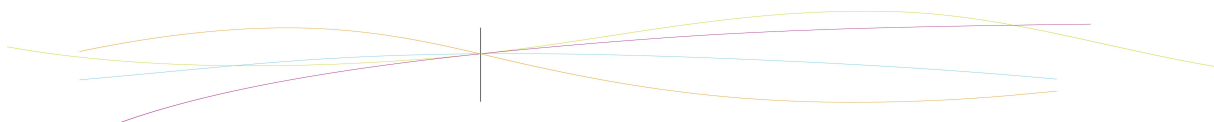
Comment pouvons-nous mesurer un élément fractal ? Les objets du quotidien se définissent par leur dimension, une ligne droite est de dimension 1, une surface de 2 et un volume de 3. Cependant comment définir la dimension de ces objets complexes fractals. La dimension constitue l'une des particularités d'un objet fractal car celle-ci ne s'exprime plus par un entier mais par un réel qui peut varier entre 0 et 3. Plusieurs mathématiciens et physiciens se sont penchés sur la question, il existe donc plusieurs définitions de la dimension fractale. Nous avons ici choisi de définir la dimension fractale grâce à l'autosimilarité de celle-ci et le rapport d'homothétie qui régit leur construction.

Pour découvrir la notion de dimension fractale définie par homothétie de rapport λ , partons d'exemples pour différentes dimensions :

- Pour un objet de dimension 1 :

Par une homothétie de rapport 2, c'est-à-dire lorsque la longueur du segment est multiplié par 2, on peut remarquer que l'objet final contient $2=2^1$ fois l'objet de départ. La dimension est de 1.

Voici des exemples pour différents λ :





$N = 1$
 $\lambda = 1$



$N = 2$
 $\lambda = 2$

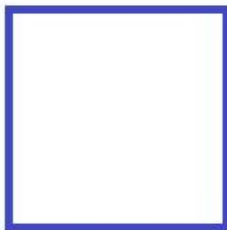


$N = 3$
 $\lambda = 3$

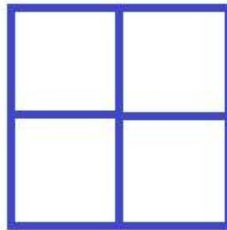
- Pour un objet de dimension 2 :

Par homothétie de rapport 2 toujours, on remarque que l'objet final contient cette fois $4=2^2$ fois l'objet initial. Ici, la dimension est de 2.

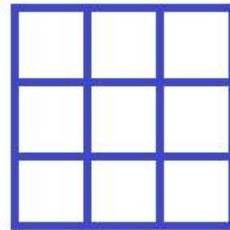
Voici des exemples pour différents λ :



$N = 1$
 $\lambda = 1$



$N = 4$
 $\lambda = 2$

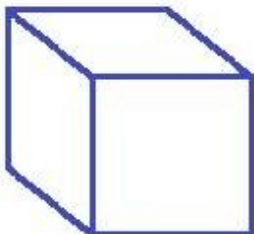


$N = 9$
 $\lambda = 3$

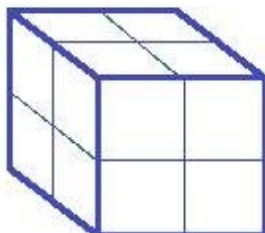
- Pour un objet de dimension 3 :

Par homothétie de rapport 2, l'objet final contient maintenant $8=2^3$ fois l'objet initial. Enfin, ici, la dimension est de 3.

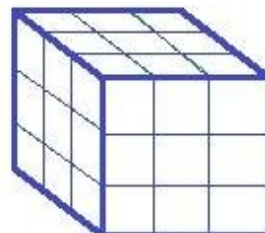
Voici des exemples pour différents λ :



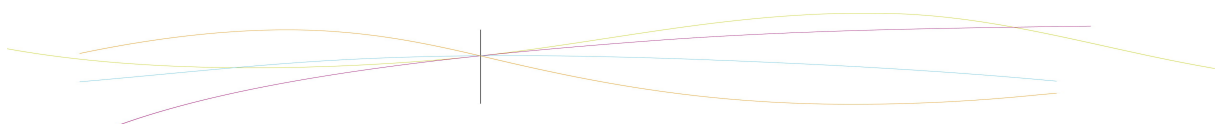
$N = 1$
 $\lambda = 1$



$N = 8$
 $\lambda = 2$



$N = 27$
 $\lambda = 3$



- Généralisation :

Si l'on généralise cette définition pour un objet quelconque de dimension d , pour une homothétie de rapport λ , alors notre objet initial se retrouve λ^d fois dans l'objet final.

Si l'on résonne dans le sens inverse.

Soit un objet fractal.

On retrouve dans cet objet N fois le nombre d'homotétie de rapport λ , on a donc : $N = \lambda^d$

$$d = \frac{\ln N}{\ln(\lambda)}$$

Par équivalence, on obtient :

Selon cette définition, il est possible de calculer la dimension de tout objet en connaissant le nombre de fois où l'on retrouve une homothétie de rapport λ dans un objet N . On peut ainsi assimiler λ à un rapport d'échelle et N au nombre de fois où l'on retrouve le motif de l'objet fractal à cette échelle, on rejoint ainsi la définition de dimension fractale à la particularité des objets factals qui est l'autosimilarité.

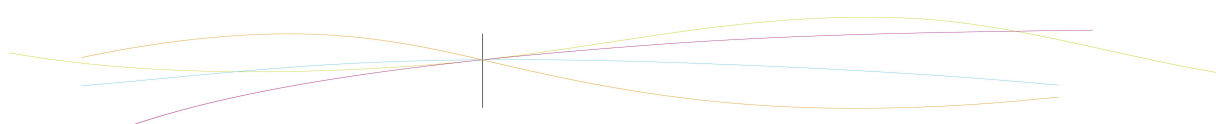
3.2. La méthode des boîtes

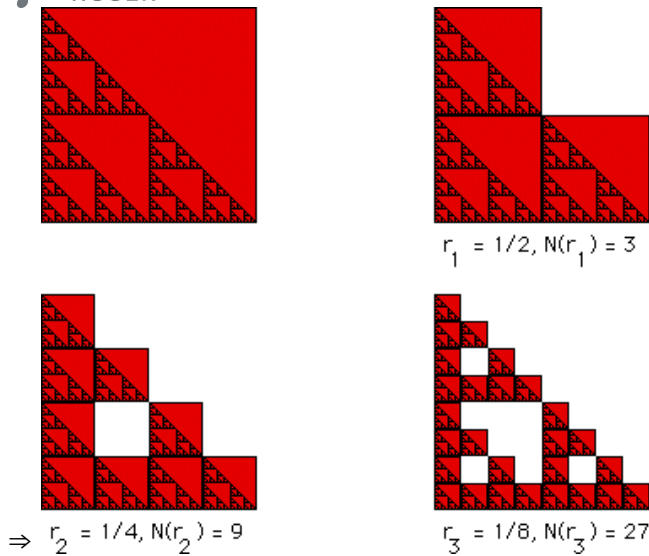
3.2.1. Théorie de la méthode des boîtes

Le principe de calcul de notre dimension fractale est de recouvrir l'ensemble A (l'image) avec des boîtes de même taille puis de faire varier la longueur ϵ du côté de la boîte et de compter à chaque variation le nombre minimal de boîtes $N(\epsilon)$ nécessaires pour couvrir entièrement A .

On l'obtient par la relation suivante : voir document sur la dimension fractale en annexe.

Soit l'ensemble A un ensemble fini de points, lorsque ϵ devient plus petit qu'un certain seuil, dans chaque boîte il n'y a qu'un seul point, et ainsi le nombre de boîtes est le même que le nombre de points et la dimension de comptage de boîtes tend vers 1 si ϵ se rapproche encore plus de 0. Dans le cas où l'ensemble A est un réseau constitué de segments, lorsque ϵ devient plus petit qu'un certain seuil ϵ_c , toute boîte de taille ϵ recouvrant un segment ne couvre aucun autre segment en dehors des points de concours entre les segments. Or pour recouvrir un segment de longueur L par des boîtes de taille ϵ , il faut L/ϵ boîtes. Ainsi le nombre de boîtes nécessaires pour recouvrir le réseau évolue en $1/\epsilon$ et ainsi la dimension de comptage de boîtes tend vers 1 si ϵ se rapproche encore plus de 0. Il apparaît donc que le choix de la gamme de valeurs que l'on retient pour ϵ est décisif. Concrètement, une fois la famille $(\epsilon_1, \epsilon_2, \dots, \epsilon_n)$ choisie, on effectue la régression des $\text{Log}N(\epsilon_i)$ par les $-\text{Log}\epsilon_i$. On obtiendra ainsi DB , dimension de comptage de boîtes de la fractale pour la famille $(\epsilon_1, \epsilon_2, \dots, \epsilon_n)$.





3.2.2. Application de la méthode des boîtes ImageJ

Nous avons décidé de mettre en pratique cette théorie par nous-même, en revanche, ayant choisi le département ASI, nous avons essayé d'appliquer la méthode des boîtes classique à notre domaine et ceci en utilisant les pixels en tant que boîtes. En effet, à l'aide du logiciel ImageJ, nous avons compté le nombre de pixels (=boîtes) occupés par un objet fractal donné sur une image.

Comme expliqué plus haut, le changement de la taille des boîtes est essentiel pour le bon fonctionnement de la méthode. Or, il est impossible de changer la taille des pixels. Pour y remédier, c'est la taille de l'image que nous avons changé : il suffit de réduire la taille de l'image plusieurs fois, de récupérer le nombre de pixels occupés pour chaque nouvelle image puis de tracer la courbe de régression. En dépit de la simplicité de la méthode, l'appliquer sur une machine s'est avéré bien plus complexe : il a fallu une bonne maîtrise des logiciels utilisés et de bonnes connaissances en algorithmique.

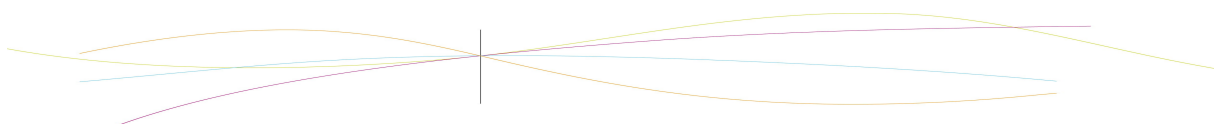
Dans cette partie, nous tenterons d'expliquer notre démarche.

3.2.3. Descriptif du logiciel

ImageJ est un programme de traitement d'image permettant d'analyser, éditer ou transformer une image en 8-bit, 16-bit et 32-bit. Il offre de nombreux avantages, comme le fait de pouvoir lire de nombreux formats d'images. Il permet de faire plusieurs opérations, notamment des calculs de l'aire d'une image et du nombre de pixels d'une sélection de l'utilisateur.

L'une des caractéristiques les plus intéressantes du programme est l'usage des « macros », un ensemble d'instructions permettant d'exécuter une tâche spécifique ou de renvoyer un résultat. La création d'une macro demande une connaissance des différentes fonctions fournies dans le programme.

Pour notre projet, ImageJ nous a été utile pour appliquer la méthode des boîtes aux images réalisées par le groupe. Pour cela, nous avons été amenés à créer un algorithme (sous forme de macros) nous permettant d'obtenir le nombre de pixels occupés par l'image à différentes résolutions.



Il existe plusieurs logiciels de traitement d'image capables de réaliser les mêmes prouesses que ImageJ mais, notre choix s'est porté sur celui-ci car il est libre (accessible à tous, portable...) et la documentation fournie avec permet de maîtriser ce logiciel et de comprendre de manière détaillée comment il fonctionne, il permet d'ajouter de nouvelles fonctionnalités et se conforme ainsi à tous nos besoins.

3.2.4. **Macros et fonctions**

La macro *Box counting* fait appel à plusieurs fonctions que nous avons défini au préalable, afin de réaliser les différentes mesures sur l'image.

La fonction *ObtenirPixelBetN()* permet d'obtenir le nombre de pixels d'une sélection, les résultats étant ensuite stockés dans des tableaux qui seront utilisés par la suite.

La fonction *ChangerResolution(i)* est utilisée afin de modifier le nombre de pixels d'une image, en réduisant ou augmentant la résolution.

La fonction *Seuillage()* applique un procédé de seuillage sur l'image sélectionnée, ce qui consiste à mettre à zéro tous les pixels ayant un niveau de gris inférieur à une certaine valeur et à la valeur maximale les pixels ayant une valeur supérieure. Ainsi le résultat du seuillage est une image binaire (pixels noirs et blancs).

3.2.5. **Principe de la macro BoxCounting()**

- On récupère dans différents tableaux les informations sur l'image sélectionnée (niveaux de gris, nombre de pixels, etc).
- On stocke dans une variable globale le nombre de pixels ayant un intérêt pour notre étude.
- On modifie la résolution de l'image afin de réduire le nombre de pixels de cette dernière.
- On réitère le procédé pour obtenir le nombre de pixels à chaque itération.

La variable *ValeurHisto* est un tableau contenant le nombre de pixels « intéressants » à chaque étape.

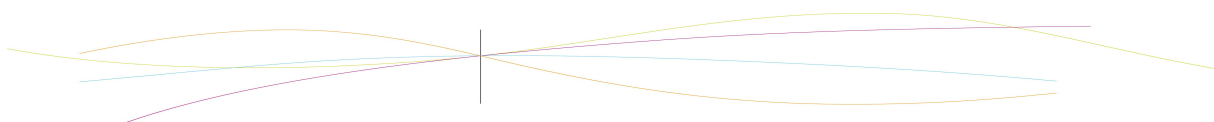
La variable *nomImage* représente le nom de l'image à analyser.

La variable *nbPixel* contient le nombre de pixels pour chaque niveau de couleurs(ou niveau de gris).

3.2.6. **Comparaison de l'algorithme utilisé avec Fraclac**

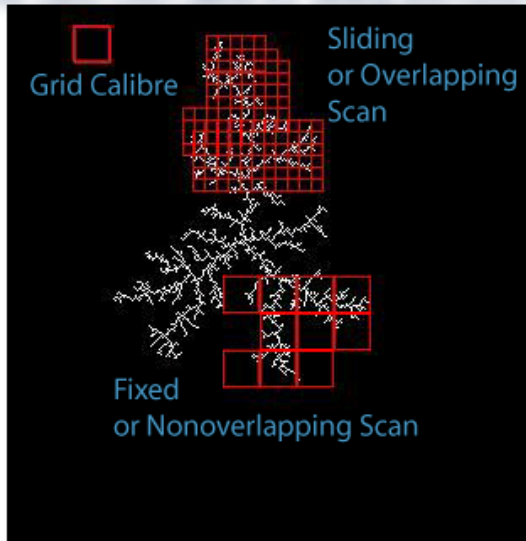
Fraclac est un plug-in adapté à ImageJ permettant de faire des analyses élaborées sur de nombreux types d'objets fractals. Facile à prendre en main, avec une documentation claire et concise, il nous a permis dans un premier temps de faire nos premières analyses sur des images créées par le groupe chargé de l'élaboration des fractales.

Ce logiciel permet d'appliquer « littéralement » la méthode des boîtes, de manière assez intuitive, en créant des sélections rectangulaires dont on fait varier la taille pour ensuite réaliser une régression et obtenir la dimension fractale.



En effet, il comporte plusieurs options telles que l'analyse d'image en 3D, en niveau de gris, la réalisation de régression linéaire permettant d'obtenir la dimension fractale instantanément... Ce logiciel présente un très grand nombre d'options, certaines dont le fonctionnement reste assez complexe et difficile à décrypter.

The Same Grid Calibre for Sliding and Fixed Scans



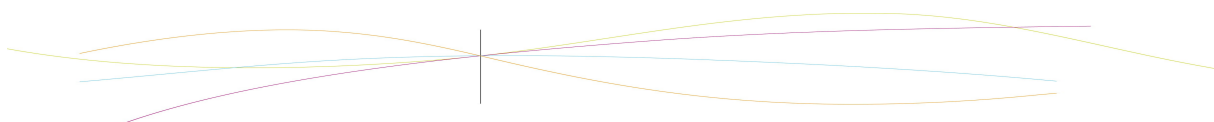
Cependant, nous avons dans un premier temps appliqué le principe de leur algorithme afin d'analyser nos images, après quelques essais infructueux, il nous est apparu difficilement réalisable en si peu de temps, le logiciel étant très élaboré. Bien que notre algorithme soit moins précis que celui de FracLac, nous l'avons fait par nous même et bien sur en comprenant son fonctionnement.

3.2.7. Perspectives

Concernant les perspectives d'amélioration de ce programme, nous avons pour objectif d'essayer d'adapter notre algorithme pour pouvoir analyser une image et trouver sa dimension fractale en fonction des niveaux de gris de l'image et non plus seulement en fonction des pixels noirs et blancs. Cela aurait ouvert notre programme à un plus grand domaine d'analyse, et aurait permis de conserver de l'information de l'image fractale et d'éviter la perte due au seuillage effectué dans notre programme. Nous avons pu observer cette méthode car le plugin fracLac permettait également d'utiliser la méthode des boites en fonction des niveaux de gris.

Nous avons également pensé à améliorer la précision de notre algorithme en essayant de réduire l'intervalle de réduction de la taille de l'image car celle-ci est réduite de moitié à chaque itération et cela ne permet pas d'effectuer beaucoup d'itérations. Cela aurait donc permis d'effectuer plus d'itérations et ainsi d'obtenir plus de points pour tracer la courbe, la dimension fractale obtenue n'en aurait été que plus précise.

Nous venons donc de voir comment déterminer la dimension d'un objet fractal. Grâce à cette méthode, nous allons donc pouvoir analyser notre objet et voir s'il est bien fractal.



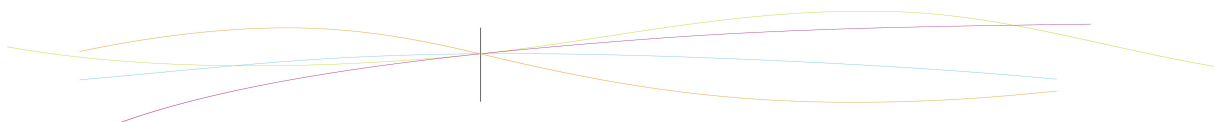
3.3. Analyse de notre objet

Après s'être assuré que notre algorithme ne comptait aucune erreur de syntaxe, nous avons repris l'image d'un objet fractal que le groupe a généré : la courbe de Koch, puis nous avons testé notre algorithme dessus. La courbe finale obtenue était linéaire et décroissante ce qui démontrait bien que l'objet était fractal, en revanche, le coefficient obtenu, qui correspond à la dimension fractale, était de 2.15 alors que le résultat attendu était de $(\ln 4 / \ln 3 = 1,26)$.

Nous avons donc décidé de tester notre algorithme sur des images d'objets fractals trouvés sur internet et dont la dimension fractale est connue et bien précise comme le carré qui a pour dimension fractale 2. Avec notre algorithme, on obtient 2.005 ce qui prouve, qu'à 0.005 près, notre algorithme est fonctionnel. Un autre objet fractal que nous avons testé est l'ensemble de Mandelbrot de dimension fractale égale à 2, le résultat fourni par notre algorithme est très similaire au résultat théorique : 2.06 (comparaison des pentes lors de la régression en annexe).

Étant sûr que notre algorithme fonctionnait, nous avons repris nos analyses avec nos images mais cette fois-ci avec le flocon de Koch, le résultat obtenu est toujours supérieur à 2. Nous savons qu'un objet fractal de dimension supérieur à 2 représente une surface. Ce qui peut expliquer l'incohérence de nos résultats puisque la courbe de Koch et le flocon de Koch représentent des périmètres. Ainsi, on peut dire que notre algorithme ne peut être appliqué qu'à des surfaces, le test fait sur le carré en est l'exemple.

D'autre part, il s'est avéré que les images utilisées n'étaient pas assez grandes pour réaliser un nombre d'itération suffisant pour déterminer avec précision la dimension fractale lors de la régression. En effet, nous avons, au maximum, réalisé 10 itérations avec les images données. Ce qui s'est révélé insuffisant pour estimer précisément la dimension fractale: l'échelle des axes étant logarithmique, une légère variation peut provoquer un changement énorme dans l'estimation de la dimension fractale. C'est une autre source d'incohérence possible de nos résultats, sachant qu'un objet idéal n'est pas limité par les pixels alors que nous le sommes.



Conclusions et Perspectives

Conclusions sur le travail réalisé

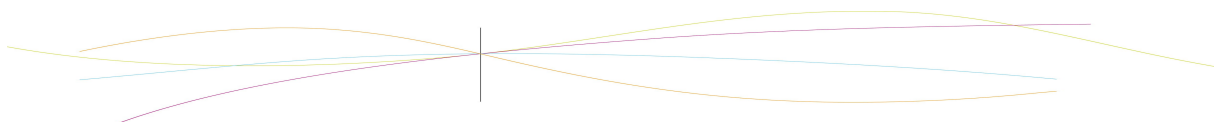
Pour conclure ce projet semestriel sur les fractales, nous sommes heureux de pouvoir dire que les objectifs que nous nous étions fixés au début du projet sont atteints. En effet, nous avons pu développer un programme qui réalise une étoile de Koch et la dessine en temps réel. De même, nous avons pu utiliser le logiciel Imagej afin de calculer la dimension fractale d'une fractale, comme nous le souhaitions. Nous avons donc pu grâce à ce projet accroître nos connaissances sur les fractales, en général et sur quelques exemples, et en manipuler grâce à l'informatique. Il n'y a qu'un seul objectif que nous n'avons pas pu atteindre. Nous aurions souhaité pouvoir modifier la dimension fractale des objets que nous générions. Nous n'y avons pas franchement réussi, nous avons certes deux programmes qui génèrent des objets avec des dimensions fractales différentes mais nous ne pouvons pas modifier facilement la dimension des objets que nous créons.

Conclusions sur l'apport personnel de cette UV de projet

Ce projet nous a permis de travailler et d'apprendre à travailler en groupe tout en étant autonome. C'est à dire que nous n'avons pas d'impératifs pour travailler, mais nous devons tout de même travailler, individuellement, afin que notre projet avance, en tant que groupe. C'est un projet de groupe, il ne peut donc pas y avoir de personnes qui ne comprennent rien à l'ensemble du projet, même si l'on peut avoir des lacunes sur certains points spécifiques. Voici ce à quoi nous avons dû aboutir en tant que groupe pendant ce semestre et que nous avons tous réussi à atteindre. Ce projet nous a donc appris à travailler de façon autonome au sein d'un groupe, tout en étant supervisé. Nous avons su éviter les risques liés au fait de ne pas avoir de chef déclaré au sein de notre groupe. C'est à dire que nous avons travaillé ensemble, et nous ne sommes pas partis dans des directions opposées.

Perspectives pour la poursuite de ce projet

Un regret que nous avons est de ne pas avoir pu dépasser le stade de la simulation informatique. Si nous avons pu étudier les propriétés physiques d'objets fractals, ce projet aurait pu prendre une toute autre dimension. Si cela semble très complexe, la réalisation et/ou l'étude d'un mur anti-bruit, par exemple, auraient été très intéressantes. Il est possible aussi de donner une dimension un peu plus mathématique et biologique au projet, en s'intéressant, par exemple, aux capacités et limites des objets fractals dans la nature (tel que les poumons). Ainsi, nous pensons que si ce projet devait être reconduit, il doit être orienté soit vers la génération d'objets fractals plus complexes (tel que l'ensemble de Mandelbrot) ou d'objets fractals dont on pourrait contrôler la dimension fractale ; ou encore, vers l'étude, comme je l'ai dit, des objets fractals naturels.



Bibliographie

Partie 1.

http://fr.wikipedia.org/wiki/Ensemble_de_Mandelbrot (valide à la date du 07/05/2015)

http://math.univ-lille1.fr/~bodin/fichiers/capes_mandelbrot.pdf (valide à la date du 07/05/2015)

<http://www.mathcurve.com/fractals/mandelbrot/mandelbrot.shtml> (valide à la date du 07/05/2015)

<http://geoffreyhistoire.pagesperso-orange.fr/fractales/g%E9ographie.html> (valide à la date du 07/05/2015)

<https://www.irisa.fr/symbiose/people/asiegel/Presentations/Neuchatel.pdf> (valide à la date du 11/05/2015)

Alain Le Méhauté, « Les géométries fractales », Edition Hermès, 1990

Jaques Dubois et Jean Chaline, « Le monde des fractales », ellipses, 2006

Partie 2.

<http://villemin.gerard.free.fr/Wwwgvm/Suite/FracMeng.htm> (valide à la date du 06/05/2015)

http://fr.wikipedia.org/wiki/Éponge_de_Menger (valide à la date du 06/05/2015)

Partie 3.

[http://fractalesland.free.fr/Nouveau%20dossier/autre%20cadre%20\(dimension%20fractale\).htm](http://fractalesland.free.fr/Nouveau%20dossier/autre%20cadre%20(dimension%20fractale).htm) ,valide à la date du 20/05/2015

<https://www-fourier.ujf-grenoble.fr/~demailly/manuscripts/fractales-print.pdf> ,valide à la date du 21/05/2015

<http://www.futura-sciences.com/magazines/mathematiques/infos/dossiers/d/mathematiques-fractales-234/page/3/> ,valide à la date du 21/05/2015

<http://perso.ens-lyon.fr/quentin.griette/d/dh.pdf> ,valide à la date du 21/05/2015

<http://perso.ens-lyon.fr/quentin.griette/d/dh.pdf> ,valide à la date du 21/05/2015

Michael Barnsley, « Fractals Everywhere », Academic Press, 1998

<http://rsb.info.nih.gov/ij/docs/intro.html> ,valide à la date du 28/05/2015

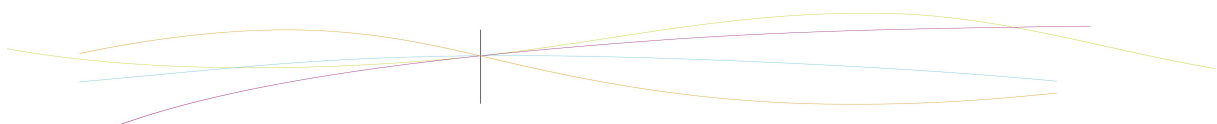
<http://rsb.info.nih.gov/ij/developer/macro/functions.html> ,valide à la date du 28/05/2015

http://cmci.embl.de/documents/ijcourses#macro_programming_in_imagej ,valide à la date du 04/06/2015

<http://rsb.info.nih.gov/ij/plugins/fractalac/FLHelp/Introduction.htm#software> ,valide à la date du 28/05/2015

<http://rsb.info.nih.gov/ij/plugins/fractalac/FLHelp/BoxCounting.htm> ,valide à la date du 04/06/2015

<http://cybergeogeo.revues.org/26440>,valide à la date du 04/06/2015



Annexe

1. Programmes réalisés

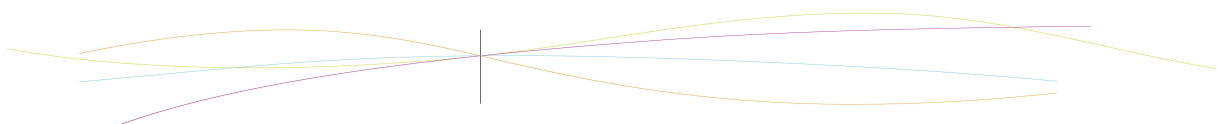
- Tracer la courbe de Koch

```

function r=Koch(xvo, yvo, n, c)

plot2d(0,0,-1,"010","",[0,-54,108,54])
xmp(1)=xvo(1);
xmp(5)=xvo(2);
xmp(2)=xmp(1)+(xmp(5)-xmp(1))/3;
xmp(4)=xmp(1)+2*((xmp(5)-xmp(1))/3);
ymp(1)=yvo(1);
ymp(5)=yvo(2);
ymp(2)=ymp(1)+(ymp(5)-ymp(1))/3;
ymp(4)=ymp(1)+2*((ymp(5)-ymp(1))/3);
a=rand();
if a<c then
a=-1;
end;
if c<a then
a=1;
end;
xmp(3)=(xmp(2)+xmp(4))*cos(%pi/3)-a*(ymp(4)-ymp(2))*sin(%pi/3);
ymp(3)=(ymp(2)+ymp(4))*cos(%pi/3)+a*(xmp(4)-xmp(2))*sin(%pi/3);
if n==0 then
plot(xmp,ymp);
end;
if n>0 then
n=n-1;
xv1=[xmp(1),xmp(2)];
yv1=[ymp(1),ymp(2)];
Koch(xv1,yv1,n,c);
xv2=[xmp(2),xmp(3)];
yv2=[ymp(2),ymp(3)];
Koch(xv2,yv2,n,c);
xv3=[xmp(3),xmp(4)];
yv3=[ymp(3),ymp(4)];
Koch(xv3,yv3,n,c);
xv4=[xmp(4),xmp(5)];
yv4=[ymp(4),ymp(5)];
Koch(xv4,yv4,n,c);
end;
r=0;
endfunction
clf
  
```

xmp représente l'abscisse des points à relier pour tracer la courbe de Koch.
ymp représente l'ordonnée des points à relier pour tracer la courbe de Koch.



xv0, xv1, xv2, xv3 et xv4 sont les systèmes d'abscisses des deux points aux extrémités du segment.

yv0, yv1, yv2, yv3 et yv4 sont les systèmes d'ordonnées des deux points aux extrémités du segment.

- Tracer notre objet fractal.

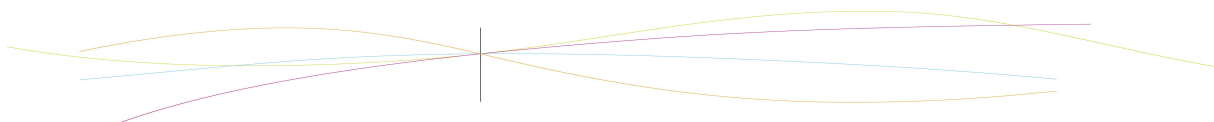
```

function r=Fract(x0,y0,x1,y1,x2,y2,n)
plot2d(0,0,-1,"010","",[0,-80,160,80])

//séparer chaque segments en 3
//x(AB)
xmp(1)=x0;
xmp(5)=x1;
xmp(2)=xmp(1)+(xmp(5)-xmp(1))/3;
xmp(4)=xmp(1)+2*((xmp(5)-xmp(1))/3);
xmp(3)=xmp(2)+(x1-x2)/3
//y(AB)
ymp(1)=y0;
ymp(5)=y1;
ymp(2)=ymp(1)+(ymp(5)-ymp(1))/3;
ymp(4)=ymp(1)+2*((ymp(5)-ymp(1))/3);
ymp(3)=ymp(2)+(y1-y2)/3;
if n=0 then
plot(xmp,ymp);
end;
//points lateraux
xv1=xmp(1)+(x2-xmp(1))/3;
yv1=ymp(1)+(y2-ymp(1))/3;
xv2=xmp(5)+(x2-xmp(5))/3;
yv2=ymp(5)+(y2-ymp(5))/3;
if n>0 then
n=n-1;
//1
Fract(xmp(2),ymp(2),xmp(3),ymp(3),xmp(4),ymp(4),n);
//11
Fract(xmp(1),ymp(1),xmp(2),ymp(2),xv1,yv1,n);
//2
Fract(xmp(3),ymp(3),xmp(4),ymp(4),xmp(2),ymp(2),n);
//22
Fract(xmp(4),ymp(4),xmp(5),ymp(5),xv2,yv2,n);
end;
r=0;
endfunction
clf

```

```
Fract(30,-8,60,71,120,-70,4)
```



- Méthode des boîtes.

```

var ValeurHisto = newArray(7);
var Resolution = newArray(7);
var nomImage = "Flocon_rempli.png";
var NbPixel = newArray(256);

macro "OuvrirImage" {
//charger l'image à étudier
    open("C:\Users\SASsa\Downloads\Flocon_rempli.png");
}

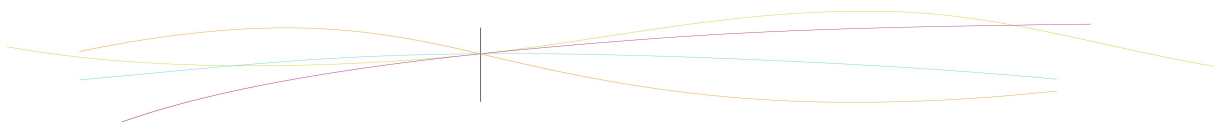
macro "BoxCounting" {
//méthode des boîtes
    ObtenirPixelBetN(); //obtenir tableau pixels
    getStatistics(area); //obtenir total pixels
    Resolution[0] = area;
    ValeurHisto[0]=area-NbPixel[255];
    ChangerResolution(0);
    for (i=1;i<7;i++) {
        ChangerResolution(i);
        ObtenirPixelBetN(); //obtenir tableau pixels
        getStatistics(area); //obtenir total pixels
        Resolution[i] = area;
        ValeurHisto[i]=area-NbPixel[255];
    }
    for (i=0;i<7;i++){
        print("ValeurHisto"+i+"="+ValeurHisto[i]);
    }
}

function ObtenirPixelBetN() {
//obtenir le nb de pixels blanc et noir de la sélection

    nBins = 256;
    run("Clear Results");
    row = 0;
    getHistogram(values, counts, nBins);
    for (i=0; i<nBins; i++) {
        setResult("Value", row, values[i]);
        setResult("Count", row, counts[i]);
        NbPixel[i] = counts[i];
        row++;
    }
    updateResults();
}

function ChangerResolution(i) {
//change la résolution de l'image
    if (i<1) {
        selectWindow(nomImage);
        BinaireEtSeuil();
    }
}

```

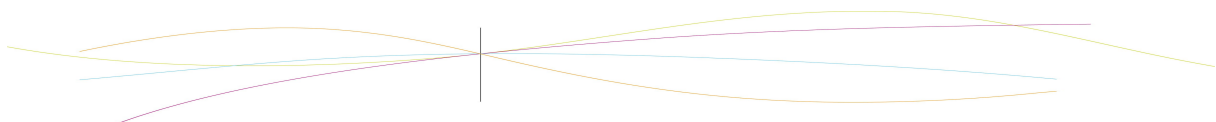


```

run("Scale...", "x=0.5 y=0.5 width=439 height=255 interpolation=Bilinear
average create title="+nomImage+"_1");
selectWindow(nomImage+"_1");
run("Scale...", "x=0.5 y=0.5 width=439 height=255 interpolation=Bilinear
average create title="+nomImage+"_2");
BinaireEtSeuil();
}
else {
selectWindow(nomImage+"_"+2*i);
BinaireEtSeuil();
getSelectionBounds(x, y, largeur, hauteur);
run("Scale...", "x="+pow(2,-1)+" y="+pow(2,-1)+" width="+((largeur/2)+
height+"+(hauteur/2)+" interpolation=Bilinear average create title="+nomImage+"_"+(2*i+1));
selectWindow(nomImage+"_"+2*i+1);
getSelectionBounds(x, y, largeur, hauteur);
run("Scale...", "x=1 y=1 width="+largeur+" height="+hauteur+"
interpolation=Bilinear average create title="+nomImage+"_"+2*(i+1));
BinaireEtSeuil();
}
}

function BinaireEtSeuil() {
//permet de faire le seuillage
setOption("BlackBackground", false);
run("Make Binary");
}

```



2. Document divers

- explication de la dimension fractale.

Méthode des boîtes

On cherche à déterminer la dimension fractale d'un objet à partir de la relation de proportionnalité entre la surface et l'échelle (à l'aide de la dimension fractale).
On pose $Surface = S$, $\Omega = échelle_{image}$ et D_f la dimension fractale.

On a

$$S = \Omega^{D_f}$$

$$\ln(S) = D_f \ln(\Omega)$$

$$d \ln S = D_f d \ln(\Omega)$$

$$D_f = \frac{d \ln(S)}{d \ln(\Omega)} = \frac{d \ln(P \times e)}{d \ln(\Omega)} = \frac{d(\ln(P) + \ln(e))}{d \ln(\Omega)}$$

avec P le périmètre de l'image et e l'aire d'un pixel ($e = 1$).

On applique à la méthode des boîtes

$$S = N_{boîtes} \times S_{boîte}$$

où $N_{boîtes}$ le nombre de boîtes et $S_{boîte}$ la surface

$$S = N_{boîtes} \times a^{D_f}$$

à l'échelle de la boîte

$$d \ln(S) = d \ln(N_{boîtes}) + D_f d \ln(a)$$

or S ne varie pas dans le temps

$$0 = d \ln(N_{boîtes}) + D_f d \ln(a)$$

$$\implies D_f = - \frac{d \ln(N_{boîtes})}{d \ln(a)}$$

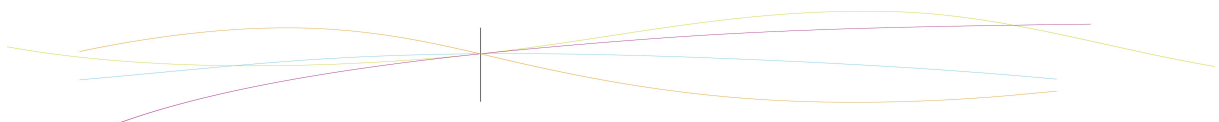
Dimension fractale au niveau du périmètre:

$$P_{total} = N_{btige} \times longueur_{tige}$$

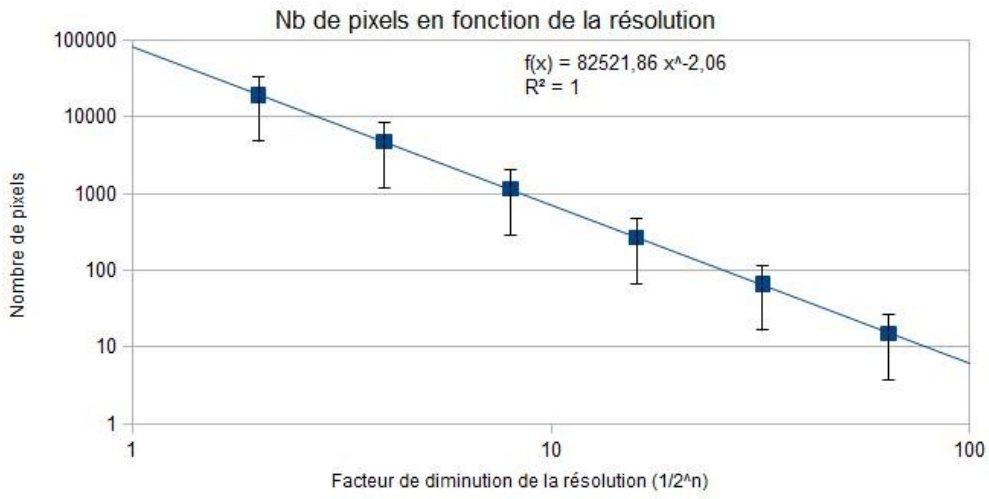
$$d \ln(P_{total}) = d \ln(N_{btige}) + D_f d \ln(longueur_{tige})$$

$$0 = d \ln(N_{btige}) + D_f d \ln(longueur_{tige})$$

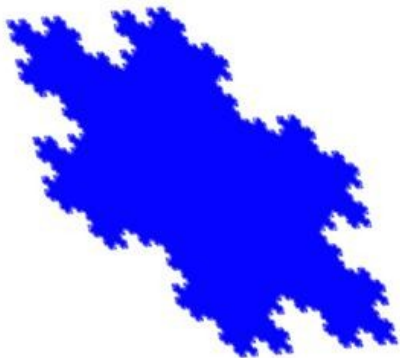
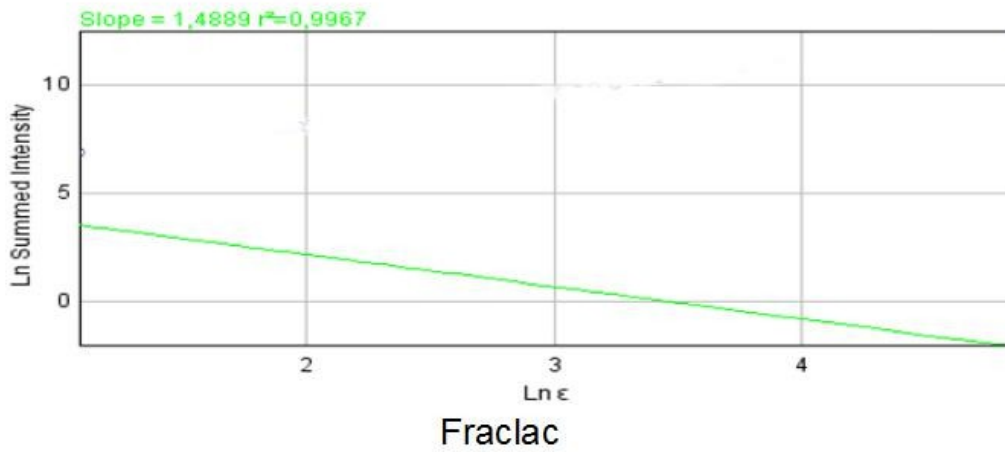
$$\implies D_f = - \frac{d \ln(N_{btige})}{d \ln(longueur_{tige})}$$



- comparaison des résultats.



Mesures expérimentales



Df obtenu = $2,06 \pm 0,86$

Flocon de Koch

