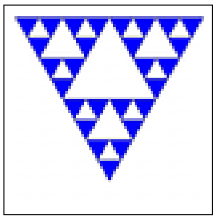


Algorithmique : les triangles de Sierpiński

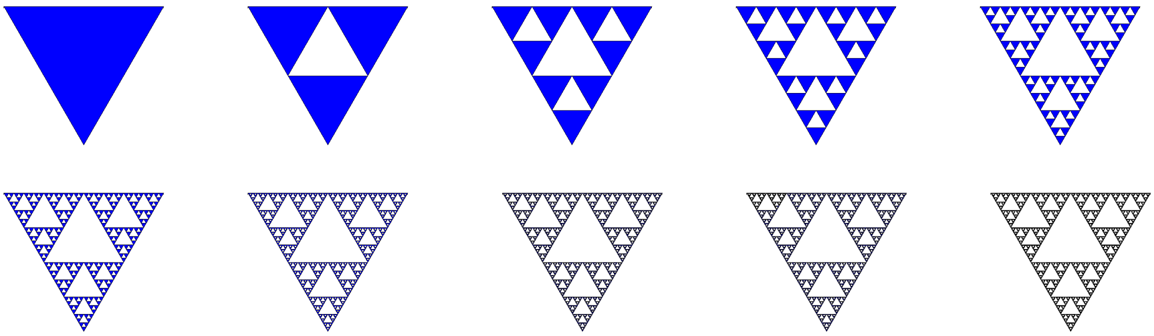
TS

Fiche Élève

Auteur : RM

La fractale de Sierpiński est la limite pratiquement impossible à imaginer d'une suite de triangles dans lesquels on a découpé des trous triangulaires de plus en plus nombreux suivant une certaine règle . On les appelle les triangles de Sierpiński (bien que ce soit des triangles à trous), notés ci-dessous TS_1, TS_2, $\ldots$, TS_N, $\ldots$

Ci-dessous, on est parti d'un triangle équilatéral (le triangle-source TS_0 en bleu). Suivent les 9 premiers triangles de la suite de Sierpiński :



On a l'impression que les derniers triangles sont égaux, mais il n'en est rien : chaque triangle s'obtient à partir du précédent en faisant de nouveaux trous. Voici un algorithme de construction des triangles de Sierpiński tel qu'il figure dans l'article « Triangle de Sierpiński » de l'encyclopédie en ligne « Wikipedia ».

1 - Commencer à partir d'un triangle quelconque du plan. Le triangle canonique de Sierpiński se construit à partir d'un triangle équilatéral ayant une base parallèle à l'axe des abscisses.

2 - Tracer les trois segments qui joignent deux à deux les milieux des côtés du triangle, ce qui délimite 4 nouveaux triangles.

3 - Enlever le petit triangle central. Il y a maintenant trois petits triangles qui se touchent deux à deux par un sommet, dont les longueurs des côtés sont la moitié de celles du triangle de départ (obtenue par une homothétie de rapport 1/2), et dont l'aire est divisée par 4.

4 - Recommencer à la deuxième étape avec chacun des petits triangles obtenus.

L'objet de ce travail est de programmer un algorithme « scilab » de tracé des premiers triangles de Sierpiński.

Première partie : tracé du premier triangle de Sierpiński

On pourra utiliser l'algorithme « PremierTrinski » dont voici le listing :

Listing 1 – Premier triangle de Sierpinski

```
clear;
clf;
X=[-1,1,0]';// Abscisses de A, B, C, en colonne.
Y=[sqrt(3),sqrt(3),0]';// Ordonnées de A, B, C, en colonne.
plot2d([X;X(1,1)],[Y;Y(1,1)],[-1.4,-0.4,1.4,2]);
// On a ajouté les coordonnées de A pour fermer le triangle ;
```

```
// [−1.4,−0.4,1.4,2] définit la portion du plan allouée au graphique.
a=gca(); // Commande technique.
orthonorme; // Repère orthonormé imposé.
xfpolys(X,Y,2); // Coloriage de ABC en bleu (identifiant 2).
a.axes_visible="off"; // On efface les axes (commande technique).
Xx=[X(2,1);X(3,1);X(1,1)]; // En vue du calcul des abscisses des milieux.
X1=(X+Xx)/2; // Abscisses de C1, A1, B1 (milieux de AB, BC, CA).
Yy=[Y(2,1);Y(3,1);Y(1,1)]; // En vue du calcul des ordonnées des milieux.
Y1=(Y+Yy)/2; // Ordonnées de C1, A1, B1.
xfpolys(X1,Y1,8);
//Tracé de A1B1C1 et coloriage en blanc (identifiant 8).
```

1 - Démontrer que ABC est bien un triangle équilatéral.

2 - Expliquer le calcul des coordonnées des milieux C_1, A_1, B_1 de AB, BC et CA.

3 - Exécuter l'algorithme « PremierTrinski ».

Précision - La commande « xfpolys(U,V,n) » où U contient des abscisses en colonne, V le même nombre d'ordonnées, en colonne et n un entier trace le polygone défini par U et V (en fermant la ligne polygonale si nécessaire) et colorie l'intérieur de ce polygone dans la couleur identifiée par n.

Deuxième partie : tracé des triangles de Sierpiński

L'algorithme ci-dessous permet d'automatiser le procédé qui a permis, dans la première partie, de passer de TS_0 à TS_1. Plus précisément, on part d'un triangle (quelconque) ABC colorié en bleu, on le découpe en 4 triangles comme ci-dessus en stockant leurs coordonnées : chacun d'eux est décrit par 2 colonnes à 3 éléments, la colonne des abscisses de ses 3 sommets et la colonne de leurs ordonnées. Ce stockage constitue les sorties X_1 et Y_1.

Listing 2 – Fonction "Sierpinski"

```
function [X1,Y1]=Sierpinski(X,Y)
  // Entrées : X et Y, abscisses et ordonnées de A, B, C.
  // (X et Y sont des tableaux à trois lignes et 1 colonne).
  // A, B et C désignent ici 3 points non alignés quelconques.
  Xx=[X(2,1);X(3,1);X(1,1)];
  XX=(X+Xx)/2; // Abscisses des milieux C1,A1,B1 de AB, BC, CA.
  Yy=[Y(2,1);Y(3,1);Y(1,1)];
  YY=(Y+Yy)/2; // Ordonnées de C1,A1,B1.
  X1=[X(1,1),XX(1,1),XX(3,1),XX(3,1);
  XX(1,1),X(2,1),XX(2,1),XX(2,1);
  XX(3,1),XX(2,1),X(3,1),XX(1,1)]; // X1 : tableau à 3 lignes  4 colonnes.
  Y1=[Y(1,1),YY(1,1),YY(3,1),YY(3,1);
  YY(1,1),Y(2,1),YY(2,1),YY(2,1);
  YY(3,1),YY(2,1),Y(3,1),YY(1,1)]; // Y1 : tableau à 3 lignes  4 colonnes.
// Sorties : X1 et Y1.  Ces tableaux définissent 4 petits triangles égaux
// qui recouvrent ABC comme TS1 recouvre TS0.
endfunction;
```

4 - Indiquer quels triangles sont décrits par les premières (respectivement deuxièmes, troisièmes et quatrièmes) colonnes de X_1 et Y_1.

5 - Le tracé du $N^{\text{ème}}$ triangle de Sierpiński peut se faire à l'aide de l'algorithme « NiemeTrinski » suivant :

Listing 3 – Nème triangle de Sierpinski

```
clf;
N=input('N=');// Entrée du rang du triangle de Sierpinski désiré.
X=[-1,1,0]';
Y=[sqrt(3),sqrt(3),0]';// X et Y définissent le triangle équilatéral
// initial.
plot2d([X;X(1,1)],[Y;Y(1,1)],[-1.4,-0.4,1.4,2]);
// Tracé du triangle équilatéral initial préalablement fermé.
// [-1.4,-0.4,1.4,2] définit la portion du plan allouée au graphique.
a=gca();
orthonorme;
xfpolys(X,Y,2);// Remplissage en bleu du triangle initial.
a.axes_visible="off";
for i=1:N
  // On construit successivement les N premiers triangles de Sierpinski.
  Xbis=[];
  Ybis=[];
  for j=1:size(X,"c")
    // Le nombre de colonnes de X est le nombre de petits triangles bleus
    // contenus dans le triangle de Sierpinski de l'étape précédente.
    [X1,Y1]=Sierpinski(X(:,j),Y(:,j));
    // Découpe du j ème petit triangle en
    // 4 triangles plus petits à l'aide de la fonction "Sierpinski".
    xfpolys(X1(:,4),Y1(:,4),8);
    // Les quatrièmes colonnes de X1 et Y1
    // définissent le triangle central qui doit disparaître
    (= être colorié en blanc, identifiant 8).
    [Xbis,X1(:,1:3)];// Stockage des abscisses des sommets des 3
    // triangles restants (en bleu).
    Ybis=[Ybis,Y1(:,1:3)];// Idem pour les ordonnées.
  end;// Fin de la construction du i ème triangle de Sierpinski.
  X=Xbis;// X a maintenant 3*size(X,"c") colonnes et 3 lignes.
  Y=Ybis;// Idem.
end;
```

Vérifier et exécuter cet algorithme pour N=1, 3, 5 et 7.

Rappel : Ayant ouvert l'application « scilab » et son éditeur de texte, on pourra ouvrir à partir de celui-ci les algorithmes « Fonctinski » (qui est une fonction « scilab ») et « NiemeTrinski ». Alors, on chargera le premier algorithme dans « scilab », puis on exécutera dans « scilab » le second (voir Barre des menus>Exécuter ?).

6 - Calculer le périmètre p_N de la zone bleue du N ème triangle de Sierpiński (c'est à dire la somme des périmètres des triangles bleus composant TS_N). Pour quelles valeurs de N ce périmètre est-il plus grand que 10^9 ?

7 - Calculer l'aire a_N de la zone bleue du N ème triangle de Sierpiński (c'est à dire la somme des aires des triangles bleus le composant). Pour quelles valeurs de N cette aire est-elle plus petite que 10^{-9} ?

8 - Pour quelles valeurs de N a-t-on $p_N \geq 10^9$ et $a_N \leq 10^{-9}$? N'est-ce pas curieux ?