

# Des fractals utiles...

par Jacques Bélair et Christiane Rousseau

Même si elle a connu une certaine faveur dans les milieux scientifiques, la théorie des objets fractals demeurait surtout un sujet de curiosité, ou, au mieux, fournissait une description commode de certains phénomènes. Plus récemment, l'utilisation de la théorie des systèmes dynamiques a permis la mise au point d'une technique de compression d'images, qui offre aussi la possibilité de générer très simplement des objets fractals. Dans cette communication, nous exposons la base de cette application, et l'illustrons à l'aide de quelques exemples. Nous croyons que, tout comme la théorie de l'itération unidimensionnelle [Bull. AMQ, Octobre 87, p. 9-20], ces exemples peuvent être présentés à des (bons) étudiants de niveau collégial.

## 1. Introduction

L'étude des formes géométriques d'objets analytiques irréguliers a connu un regain de popularité au cours des dernières années. Les efforts de B. Mandelbrot ont porté fruit, et beaucoup de scientifiques se sont mis à l'étude de ces objets, naguère qualifiés de marginaux, et le plus souvent réduits au rang de contre-exemples. Parallèlement, et non indépendamment, l'essor de la dynamique non linéaire a permis l'introduction des notions de chaos déterministe et d'attracteurs étranges.

Les méthodes que nous exposons ici se situent à la conjonction de ces deux domaines. Il s'agit d'une utilisation de la théorie de l'itération pour la représentation d'images. Si l'on veut reproduire un paysage urbain, on se débrouille assez bien avec des segments de droite. Au contraire, lorsque ces images sont des scènes naturelles, par exemple, un arbre, une feuille, une forêt ou le tracé d'un fleuve, une condensation significative de l'information nécessaire à la reconstruction de ces images exige l'emploi d'objets beaucoup plus complexes, les fractals.

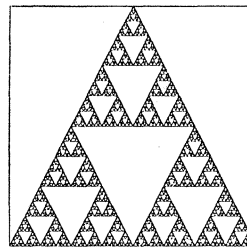
Une application plus connue de la géométrie fractale est la génération d'images irrégulières par des méthodes stochastiques, pour la représentation de paysages. Dans ce cas, on peut, à un coût relativement modeste, générer des détails très fins du profil d'une montagne, par exemple, ou de la morphologie d'une surface planétaire. C'est précisément ce qui a été employé dans la réalisation des films «Stark Trek II» et «The Last Starfighter». Ce que nous représentons ici est une solution au problème inverse: à partir d'une image donnée, déterminer une représentation comprimée qui permette de la générer efficacement.

Mentionnons que notre exposé est bref à dessein, et ne traite que de cas particuliers de la théorie. Pour un exposé plus exhaustif, on consultera le livre récent de M. Barnsley [1] ou son article dans [2, pp. 219-242].

## 2. Un exemple particulier: le tapis de Sierpinski

La méthode repose sur l'itération de quelques fonctions affines bien choisies. Le tapis de Sierpinski, représenté à la Figure 1, est un fractal très connu. (voir, par exemple, [2, p.282]). On le construit généralement suivant un algorithme purement géométrique: partant de la surface d'un triangle, on enlève de son intérieur un triangle dont les sommets sont les points milieux de chacun des côtés du triangle initial. On répète ensuite cette procédure, d'abord pour chacun des trois triangles restant, puis, de façon récurrente, sur chacun des  $3^j$  triangles,  $j = 1, 2, \dots$  Après un nombre infini de découpage, la Figure 1 est obtenue.

**Figure 1.** Les figures 1 à 3 montrent différentes constructions du tapis de Sierpinski. Ici, la méthode géométrique classique du retrait successif de un, puis 3, puis 9,... triangles du grand triangle.



Voyons maintenant une façon directe, et plus facilement programmable sur un ordinateur, de le tracer. La méthode consiste, à partir du point  $(x_0, y_0) = (0,0)$ , à choisir au hasard, avec probabilité  $\frac{1}{3}$  chacune, une transformation T parmi

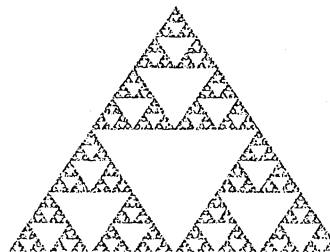
$$T_1(x,y) = \left(\frac{x}{2}, \frac{y}{2}\right),$$

$$T_2(x,y) = \left(\frac{x}{2} + \frac{y}{2}, \frac{y}{2}\right),$$

$$T_3(x,y) = \left(\frac{x}{2} + \frac{1}{4}, \frac{y}{2} + \frac{1}{2}\right),$$

à tracer le point  $(x_1, y_1) = T(x_0, y_0)$ , et à répéter le même algorithme à partir de  $(x_1, y_1)$ . Si on est suffisamment patient, on obtiendra ainsi le tapis de Sierpinski (Figure 2).

**Figure 2.** Résultat de l'itération successive du point (0,0) sous les transformations  $T_1$ ,  $T_2$  et  $T_3$  de la section 2, chacune étant choisie avec probabilité  $\frac{1}{3}$

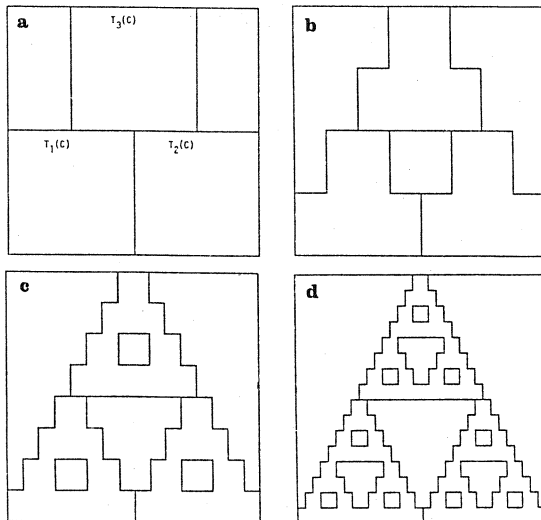


Pour comprendre ce phénomène, regardons ce qui se passe si au lieu de partir de l'origine, on part d'un point quelconque  $(x_0, y_0)$  dans le carré unité  $C = [0,1] \times [0,1]$ . L'image du point  $(x_0, y_0)$  est dans la réunion  $C_1 = T_1(C) \cup T_2(C) \cup T_3(C)$  (voir Figure 3a). À l'étape suivante, l'image du point  $(x_1, y_1)$  est contenue dans la réunion  $C_2 = T_1(C_1) \cup T_2(C_1) \cup T_3(C_1)$ , et ainsi de suite (voir Figures 3b, 3c, et 3d). On voit qu'après 10 étapes les itérés sont tellement proches du tapis de Sierpinski que l'oeil ne peut plus distinguer la différence: ce tapis est un **attracteur** pour le **système de fonctions itérées**  $(T_1, T_2, T_3)$  avec probabilités  $(P_1 = \frac{1}{3}, P_2 = \frac{1}{3}, P_3 = \frac{1}{3})$ .

Avant de passer à la généralisation, regardons les ingrédients essentiels:

1. Les applications  $T_1, T_2$  et  $T_3$  sont toutes trois **contractantes**: dans ce cas-ci, elles divisent les distances par 2, ce qui garantit que les itérées finiront sur un **attracteur**.
2. Cet attracteur,  $S$ , a la propriété que  $S = T_1(S) \cup T_2(S) \cup T_3(S)$ .
3. Les probabilités  $P_1, P_2, P_3$  sont proportionnelles aux aires respectives de  $T_1(S), T_2(S), T_3(S)$ .

**Figure 3.** Images successives du carré unité  $[0,1] \times [0,1]$  sous l'union des transformations  $T_1, T_2$  et  $T_3$  de la section 2.



### 3. La méthode générale

Pour coder l'objet  $A$  à reproduire, on doit l'écrire comme une réunion  $A = T_1(A) \cup T_2(A) \cup \dots \cup T_n(A)$ , où les transformations  $T_i$  sont des transformations affines contractantes et les probabilités  $P_i$  sont choisies proportionnelles aux surfaces des  $T_i(A)$ . Par exemple, dans le cas de la fougère de la Figure 4d, on prend les 4 transformations

$$\begin{aligned}
 T_1(x,y) &= (0, .16y), \\
 T_2(x,y) &= (.2x - .26y, .23x + .22y + .16), \\
 T_3(x,y) &= (-.15x + .28y, .26x + .24y + .044), \\
 T_4(x,y) &= (.85x + .04y, -.04x + .85y + .16),
 \end{aligned}$$

avec probabilités  $P_1 = .01, P_2 = .07, P_3 = .07$  et  $P_4 = .85$ ,

et on regarde l'attracteur dans le carré  $[-.5, .5] \times [0, 1]$ . L'image de  $T_1$  est la queue de la fougère, l'image de  $T_2$  (respectivement  $T_3$ ) est la branche inférieure gauche (respectivement droite) de la fougère, et l'image de  $T_4$  est la fougère privée de sa queue et de ses 2 branches inférieures. On voit sur les Figures 4a à 4c le traçage successif des points qui donneront la fougère de la Figure 4d.

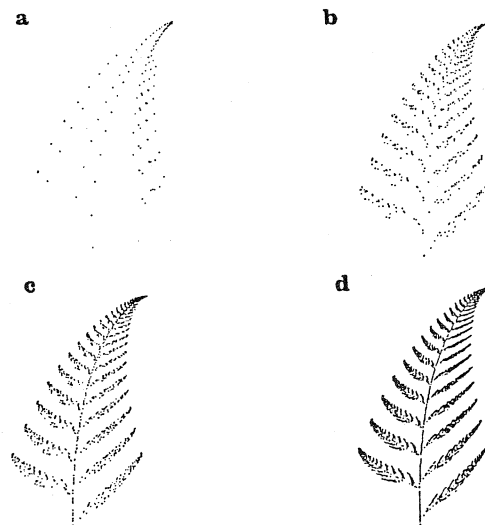
Une question demeure: comment trouver les transformations  $T(x,y)$ ? Une telle transformation peut s'écrire

$T(x,y) = (ax + by + e, cx + dy + f)$   
 où  $a, b, c, d, e$  et  $f$  sont des constantes; pour les déterminer, il suffit de connaître l'image sous  $T$  de trois points. Les relations  $T(x_j, y_j) = (X_j, Y_j), j = 1, 2, 3$ , nous donnent 6 équations aux 6 inconnues  $a, b, c, d, e$ , et  $f$ . Les applications sont contractantes lorsque  $a^2 + b^2 < 1$  et  $c^2 + d^2 < 1$ .

La méthode ne semble permettre de représenter que des objets auto-similaires, c'est-à-dire semblables à eux-mêmes lorsqu'on les regarde à la loupe. Comme les objets naturels sont rarement exactement auto-similaires et comme il est pratiquement impossible de définir les transformations appropriées avec une précision absolue, on aimerait jouir d'un résultat de «stabilité dans l'approximation»: si les transformations choisies approchent celles recherchées, alors l'attracteur obtenu  $B$  approche l'ensemble désiré  $A$ . Une telle stabilité est garantie par le théorème «de collage» (en français même dans les textes américains), exposé dans la prochaine section.

Les valeurs prises par les probabilités  $P_i$  influencent moins l'ensemble  $A$  lui-même que la fréquence de visite de ses parties: plus  $P_k$  est élevée, pour une valeur fixée de  $k$ , et plus  $T_k(A)$  est visitée souvent par les itérées de  $x_0$ . Dans la pratique, on a donc avantage à choisir une valeur de  $P_k$  qui soit proportionnelle à la surface relative de  $T_k(A)$  par rapport à celle de  $A$ .

**Figure 4.** Images générées par l'itération des transformations  $T_1, T_2, T_3, T_4$  de la section 3.



#### 4. Les mathématiques sous-jacentes

La base de l'algorithme est une extension des théorèmes de points fixes. Une version classique, et presque élémentaire d'un tel théorème est la suivante:

**Théorème A.** Soit  $f$  une fonction continue de l'intervalle  $[0,1]$  dans lui-même, qui est contractante: pour tout  $x,y$  dans  $[0,1]$ ,  $|f(x)-f(y)| \leq \tau |x-y|$ , pour une constante  $\tau$  comprise entre 0 et 1. Alors  $f$  possède un unique point fixe,  $f(x) = x$ .

La preuve de ce théorème, telle qu'originellement présentée par Picard dans un contexte particulier, repose sur un schème itératif: partant d'un point  $x_0$  quelconque dans l'intervalle  $[0,1]$ , on définit la suite  $(x_n)$ , dite des approximations successives, par  $x_{n+1} = f(x_n)$ , et on montre que la limite de cette suite existe et est précisément un point fixe de la fonction  $f$ . On appelle  $f$  une contraction si elle satisfait les hypothèses du théorème A.

Plutôt que de considérer une seule fonction, on a affaire à un ensemble de transformations  $T_1, \dots, T_m$ , affublées chacune d'une probabilité  $P_i$  telle que  $0 < P_i < 1$  et leur somme soit un. On se restreint ici au plan euclidien, même si ces notions se traitent dans un cadre plus général. Rappelons d'abord deux notions de distance, pour les points et les ensembles du plan.

**Définition A i.** Pour un point  $(x,y)$  du plan, on note  $\|(x,y)\| = \sqrt{x^2+y^2}$  sa distance à l'origine, aussi appelée sa norme;  
ii. Deux sous-ensembles  $X,Y$  du plan sont proches (leur distance  $H(X,Y)$  est inférieure à un certain  $\epsilon > 0$ ) si, pour tout point  $x$  de  $X$ , il existe un point  $y$  de  $Y$  tel que  $\|x-y\| < \epsilon$ , et réciproquement.

**Définition B.** On appelle système de fonctions itérées (SFI) le couple formé de  $m$  transformations affines contractantes  $T_1, \dots, T_m$  et de  $m$  nombres réels  $P_1, \dots, P_m$  satisfaisant  $0 < P_i < 1$ ,  $\sum_{i=1}^m P_i = 1$ . Soit  $s$  la plus petite constante telle que pour tous points  $x$  et  $y$  de  $\mathbb{R}^2$ ,  $\|T_i(x)-T_i(y)\| \leq s \|x-y\|$ . Cette constante est dite constante de Lipschitz de  $T_i$ .

Le théorème d'existence d'un attracteur est le suivant:

**Théorème B** Soit  $(T_1, \dots, T_m; P_1, \dots, P_m)$  un SFI; alors il existe un unique sous-ensemble  $A$ , non-vide, fermé et borné de  $\mathbb{R}^2$  tel que  $A = \bigcup_{i=1}^m T_i(A)$ , appelé attracteur du SFI.

Cet ensemble peut s'obtenir de deux façons:

(i) Soit  $A_0$  un sous-ensemble quelconque mais borné du plan, et, pour  $m > 0$ , posons  $A_n = \bigcup_{i=1}^m T_i(A_{n-1})$ ; alors la suite d'ensembles  $(A_0, A_1, \dots)$  converge vers  $A$ , c'est-à-dire que la distance  $H(A, A_n)$  tend vers 0.

(ii) Soit  $x_0$  un point quelconque de  $\mathbb{R}^2$ ; pour  $n \geq 0$ , choisir pour  $x_{n+1}$  un des points  $T_1(x_n), \dots, T_m(x_n)$  dont les probabilités respectives sont  $P_1, \dots, P_m$ ; alors, avec probabilité 1, l'ensemble limite des points  $(x_0, x_1, \dots)$  est  $A$ .

L'analogie SFI-contraction est claire au vu de l'algorithme i. En pratique, puisqu'il est généralement plus efficace d'itérer des points que des ensembles, c'est la seconde méthode qui est utilisée. De plus, puisque l'ensemble  $A$  est la limite des itérées du point de départ, on ne trace les points obtenus qu'à partir d'un certain rang (en pratique, à partir du dixième rang).

Le théorème de «stabilité dans l'approximation» est le suivant:

**Théorème C (de collage)** Soit un SFI, avec constante de Lipschitz  $s$  et  $\epsilon > 0$ ; si  $B$  est un sous-ensemble de  $\mathbb{R}^2$  tel que  $H(B, \bigcup_{i=1}^m T_i(B)) < \epsilon$ , alors  $H(A, B) < \epsilon / (1 - s)$ , où  $A$  est l'attracteur du SFI.

En mots, le théorème précédent assure que si pour un SFI donné, on trouve un ensemble bien approché par les réunions de son image sous les transformations, alors cet ensemble approche bien l'attracteur du SFI, qui est exactement la réunion de ses images.

#### 5. Applications et implémentation

La méthode exposée ici permet la représentation d'images, même très complexes, par les paramètres définissant quelques transformations affines et leurs probabilités associées. C'est une méthode beaucoup plus efficace que la discrétisation directe, consistant à établir un treillis sur l'image et à emmagasiner chaque pixel. De plus, l'absence de régularité dans l'objet à représenter ne complique en rien l'implémentation de la méthode: nulle part n'utilise-t-on d'approximations par des droites ou des arcs de cercles, par exemple. C'est pour ces raisons que les militaires américains ont manifesté un net intérêt financier pour cette approche de la compression d'images.

En terminant, nous donnons en Appendice un programme facilement altérable, en langage BASIC, qui permet de générer des ensembles à l'aide de transformations affines. Les paramètres définissant ces fonctions apparaissent aux lignes 20 et 30. Nous laissons au lecteur la découverte de l'attracteur engendré par les paramètres utilisés ici.

#### Bibliographie

- [1] M. BARNSLEY. *Fractals Everywhere*. Academic Press, New York, 1988.
- [2] H.-O. PEITGEN, D. Saupe, éd. *The Science of Fractal Images*, Springer, New York, 1988.

---

## Appendice

```
10 DIM a(2), b(2), c(2), d(2), e(2), f(2), p(2)
20 DATA .917, -.224, .332, .867, .159, -.029, .8
30 DATA -.178, -.153, .143, -.191, .93, .18, 1
40 FOR j=1 TO 2
50 READ a(j), b(j), c(j), d(j), e(j), f(j), p(j)
60 NEXT j
70 x=0
80 y=0
90 FOR n=1 TO 20000
100 pk=RND
```

```
110 IF pk<=p(1) THEN k=1 ELSE k=2
120 newx = a(k)*x + b(k)*y + e(k)
130 newy = c(k)*x + d(k)*y + f(k)
140 x = newx
150 y = newy
160 IF n > 10 THEN LINE (x*300, 260-y*220)-(x*300,
    260-y*220)
170 NEXT n
180 END
```

---