

# La récursivité de la tortue

Roger Cuppens(\*)

*Cet article fait suite à l'article La récursivité ou l'algorithmique sans boucles paru dans le numéro 513 où j'ai présenté les idées générales sur la récursivité et des exemples numériques. Dès 1967, pour initier des jeunes enfants dès l'école primaire à la programmation, Seymour Papert a imaginé un langage fonctionnel admettant la récursivité où une « tortue » trace des figures géométriques : le langage LOGO. Je reprends ici en les simplifiant à l'extrême les idées de Papert et aborde ainsi des notions fondamentales en géométrie telles que recouvrement, pavage et objets fractaux.*

## 1. L'univers de la tortue

Formellement, cet univers est un univers de fonctions d'une variable réelle avec deux primitives de base *avance* et *tourne*<sup>(1)</sup>. On obtient de nouvelles fonctions par concaténation de ces deux primitives. Pour simplifier l'écriture, on introduit aussi une fonction primitive *répète* qui à partir d'un entier naturel  $n$  et d'une fonction  $f$  consiste à concaténer  $n$  fois la fonction  $f$  avec elle-même.

Un tel univers n'a d'intérêt qu'avec son interprétation géométrique : dans un plan orienté, une tortue est un couple composé d'un point du plan et d'une orientation par rapport à une direction fixe du plan. Pour déplacer la tortue,

- *avance*( $l$ ) demande à la tortue d'avancer d'une longueur  $l$  dans la direction définie par son orientation en laissant une trace de ce déplacement ;
- *tourne*( $a$ ) demande à la tortue de tourner de l'angle dont la mesure (en degrés) est  $a$ .

Par exemple, l'instruction *répète*(4, [*avance*( $l$ ) *tourne*(90)]) demande à la tortue de se déplacer sur un carré de côté  $l$ , la position et l'orientation finales étant les mêmes qu'au départ. Nous écrivons donc

$$\text{carré}(l) = \text{répète}(4, [\text{avance}(l) \text{tourne}(90)])$$

qui donne bien un carré<sup>(2)</sup> :

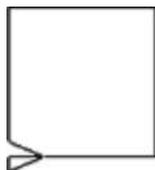
(\*) IRES de Toulouse

(1) Logo utilise quatre fonctions *av*, *re*, *td* et *tg* respectivement pour *avance*, *recule*, *tourne à droite* et *tourne à gauche*, mais admettant des valeurs négatives. Les considérations qui suivent sont donc destinées à des personnes maîtrisant l'orientation et les nombres négatifs. Pour les autres, il faudrait revenir aux fonctions de Logo.

(2) On peut obtenir cette figure (et les suivantes) avec le logiciel libre et gratuit *xlogo* en écrivant :

```
pour carré :l
  repete 4 [av :l tg 90]
fin
```

Un autre logiciel libre *Géotortue* utilise une syntaxe légèrement différente.

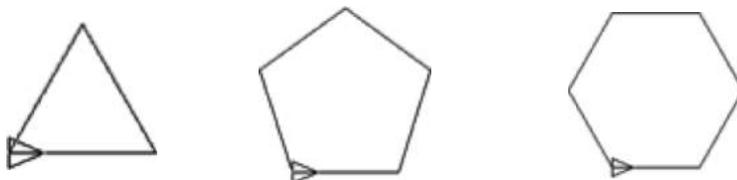


le petit triangle à gauche montrant la position finale de la tortue (qui est aussi ici la position initiale).

Plus généralement, avec la fonction

$$\text{poly}(n, l) = \text{répète}(n, [\text{avance}(l) \text{ tourne}(360/n)])$$

la tortue trace un polygone régulier (convexe) à  $n$  côtés de longueurs  $l$ .



*Remarque.* Dans cet univers, la tortue ne peut tracer que des courbes composées d'un nombre fini de segments.

## 2. Recouvrement des polygones réguliers

Nous commençons par le problème suivant :

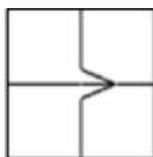
Si  $n$  est un entier, obtenir récursivement un recouvrement d'un carré de côté  $l$  par  $2^n$  carrés égaux.

Essayons pour  $n = 2$ .

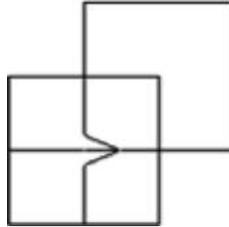
L'instruction  $\text{répète}(4, [\text{carré}(l/2)])$  ne donne qu'un seul carré (parcouru 4 fois) : il faut faire tourner chacun des quatre carrés. L'instruction

$$\text{répète}(4, [\text{carré}(l/2) \text{ tourne}(90)])$$

semble convenir.



Mais si on y regarde de plus près en traçant d'abord le carré de côté  $l$



on voit que parmi les quatre carrés un seul est intérieur au carré de départ : il faut de plus déplacer chacun des carrés. On arrive à l'instruction

$$\text{r\`ep\`ete}(4, [\text{carr\`e}(l/2) \text{avance}(l) \text{tourne}(90)])$$

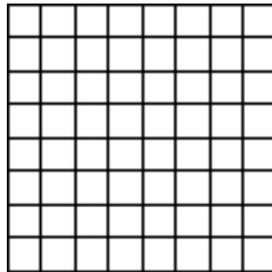
et par conséquent à la fonction

$$\text{recouv carr\`e}(n, l) =$$

$$\text{si}(n = 0) \text{ alors nil}$$

$$\text{sinon } [\text{r\`ep\`ete}(4, [\text{recouv carr\`e}(n - 1, l/2) \text{avance}(l) \text{tourne}(90)])]$$

qui donne bien le résultat voulu : pour  $n = 4$ , on obtient un échiquier<sup>(3)</sup> :



*Remarque.* La courbe limite quand  $n$  tend vers l'infini est dense dans le carré : nous dirons pour simplifier qu'elle « recouvre » le carré. Dans la réalité, l'ordinateur fournira pour des grandes valeurs de  $n$  un carré entièrement noir, puis la tortue restera immobile<sup>(4)</sup> pour des valeurs encore plus grandes de  $n$ .

On peut généraliser la fonction `recouv carré` à des polygones réguliers à  $p$  côtés, ce qui donne

$$\text{recouv poly}(n, p, l) =$$

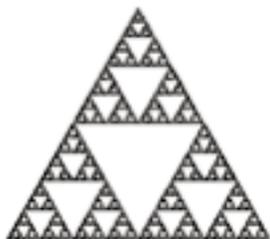
$$\text{si}(n = 0) \text{ alors nil}$$

$$\text{sinon } [\text{r\`ep\`ete}(p, [\text{recouv poly}(n - 1, p, l/2) \text{avance}(l) \text{tourne}(360 / p)])]$$

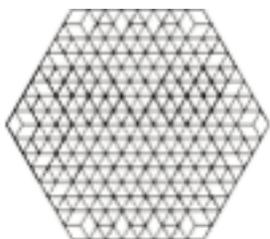
(3) Désormais, pour des questions d'esthétique, je cache la tortue dont la position initiale sera toujours le coin inférieur gauche de la figure avec une orientation horizontale.

(4) En réalité, elle tournera sur place sans se déplacer.

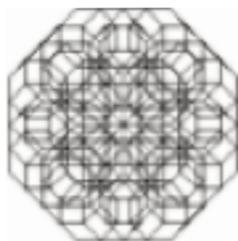
Pour  $p = 3$ , on obtient les premières figures d'un fractal classique, le tapis triangulaire de Sierpinski<sup>(5)</sup>



pour  $p = 6$ , un recouvrement de l'hexagone



et dans le cas général des « tapis » assez spectaculaires (ici  $n = 3, p = 8$ ).



#### 4. Les pavages du plan par des polygones réguliers

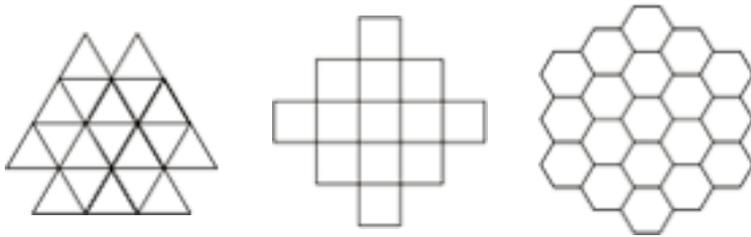
Les cas du carré recouvert par des carrés et de l'hexagone recouvert par des triangles et des hexagones donne envie de regarder ce qui se passe quand on intervient à l'extérieur du polygone. Pour passer de l'intérieur à l'extérieur il suffit de tourner « dans l'autre sens », ce qui finit par donner dans le cas général la fonction

$\text{pavpoly}(n, p, l) = \text{si } (n = 0) \text{ alors nil}$

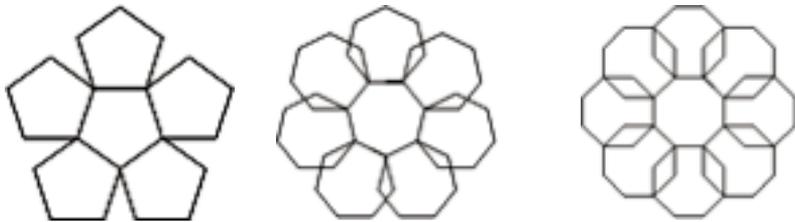
$\text{sinon } \left[ \text{répète} \left( p, \left[ \text{avance}(l) \text{ tourne} \left( (-1)^n \times 360 / p \right) \text{pavpoly}(n-1, p, l) \right] \right) \right]$

Avec cette fonction on obtient les cas classiques  $p = 3, p = 4$  et  $p = 6$  où l'on peut paver le plan respectivement par des triangles, des carrés et des hexagones

(5) Je reviendrai au paragraphe 6 sur ce tapis obtenu par Sierpinski en 1916.



Dans les autres cas ce n'est pas possible car dans le cas  $p = 5$  il n'y a pas recouvrement tandis que pour  $n > 6$ , il y a chevauchement :



#### 4. Les courbes fractales

La notion d'objet fractal a été introduite par Benoit Mandelbrot dans son livre *Les objets fractals – Forme, hasard et dimension*. Mais on en avait déjà obtenu dès la fin du 19<sup>e</sup> siècle et le début du 20<sup>e</sup> sous la forme de « monstres » mathématiques. Dans ce qui précède, nous avons vu apparaître l'un d'eux, le triangle de Sierpinski. Depuis diverses méthodes ont été introduites pour obtenir de tels objets.

Dans la suite, nous utiliserons la méthode suivante qui remonte à Mandelbrot (avec sa terminologie). Un *objet fractal* est la limite quand  $n$  tend vers l'infini d'une suite d'objets (ici des courbes)  $C_n$  définies récursivement et qui ne font que donner un aperçu de la complexité de l'objet limite. La courbe  $C_0$  est appelée *initiateur* et on obtient  $C_{n+1}$  à partir de  $C_n$  en remplaçant chaque segment composant  $C_n$  par un ensemble de segments appelé *générateur*. La courbe  $C_n$  est le  $n$ -ième *téragone* de la courbe limite.

*Remarque.* Le terme *téragone* désigne un polygone avec un grand nombre de côtés. Avec un ordinateur on ne peut obtenir que les premiers téragones.

#### 5. La courbe de von Koch

##### 5.1. Description de la courbe

Pour la courbe que von Koch a obtenu en 1904, l'initiateur est un segment de longueur  $l$  :



que l'on obtient avec la fonction  $k_0(l) = \text{avance}(l)$  et pour obtenir le générateur on coupe le segment en trois parties égales et on remplace le segment central par deux segments formant avec ce segment un triangle équilatéral :



On obtient cette courbe avec la fonction

$$k_1(l) = [k_0(l/3) \text{ tourne}(60) k_0(l/3) \text{ tourne}(-120) k_0(l/3) \text{ tourne}(60) k_0(l/3)],$$

ce qui donne pour les téragones de la courbe de von Koch l'algorithme :

$$\begin{aligned} \text{koch}(n, l) = & \text{si } (n = 0) \text{ alors } [\text{avance}(l)] \\ & \text{sinon } [\text{koch}(n-1, l/3) \text{ tourne}(60) \\ & \text{koch}(n-1, l/3) \text{ tourne}(-120) \\ & \text{koch}(n-1, l/3) \text{ tourne}(60) \text{koch}(n-1, l/3)] \end{aligned}$$

avec lequel on obtient par exemple pour  $n = 5$  :

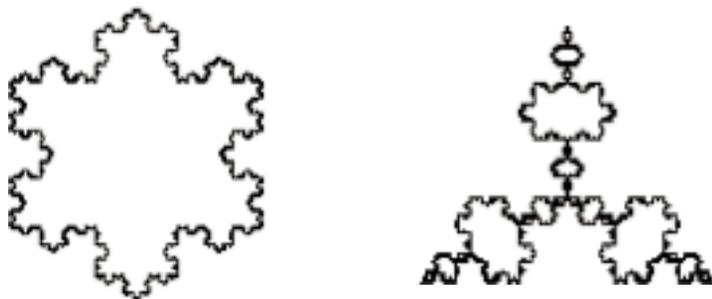


*Remarques.* 1. On voit que la courbe de von Koch n'a de tangente en aucun point.

2. On a ici une méthode générale pour obtenir les fonctions des fractales : on écrit la fonction décrivant le générateur et pour obtenir le  $n$ -ième téragone on remplace les instructions avance par le  $(n-1)$ -ième téragone

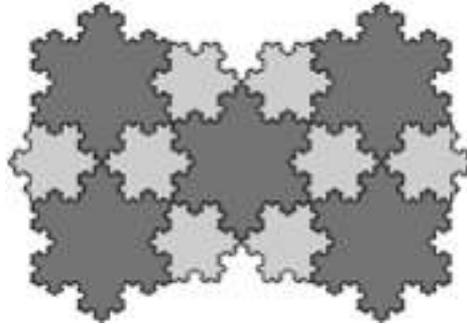
## 5.2. Le flocon et l'anti-flocon

Avec l'instruction répète( $3, [\text{koch}(n, l) \text{ tourne}(\pm 120)]$ ) (qui revient à dire que l'on prend comme initiateur un triangle équilatéral, le signe de la valeur de l'angle indiquant le sens de parcours), on obtient dans le cas « - » le flocon de von Koch et dans le cas « + » une belle courbe « en chandelier », l'anti-flocon :



*Remarques.* 1. Le flocon de von Koch est un exemple de courbe de longueur infinie délimitant une surface d'aire finie.

2. Comme le montre la figure suivante, on peut paver le plan avec des flocons de tailles différentes (le rapport des longueurs étant  $\sqrt{3}$ ).

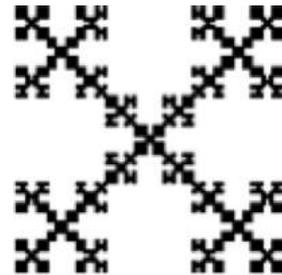
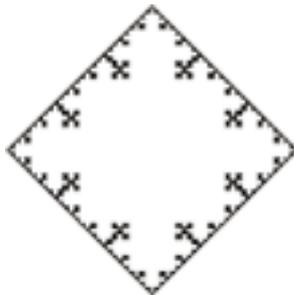


### 5.3. Une variante de la courbe de von Koch

On peut imaginer beaucoup de variantes de cette courbe, par exemple de remplacer dans le générateur les deux côtés du triangle équilatéral par trois côtés d'un carré :



et de prendre comme initiateur un carré : on obtient un tapis ou une croix suivant le sens de parcours de ce carré :



### 5.4. Complément<sup>(6)</sup>

On obtient un recouvrement du flocon de von Koch avec la courbe ayant pour initiateur un segment et pour générateur et premiers téragones les courbes :



(6) On trouvera sur le site de l'APMEP les détails des paragraphes indiqués (ici et dans la suite) comme compléments.

## 6. Le tapis triangulaire de Sierpinski

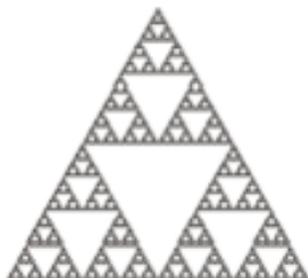
Nous avons vu apparaître le tapis de Sierpinski au paragraphe 2. On peut l'obtenir de plusieurs autres manières. L'une d'entre elles consiste à tracer la courbe ayant pour initiateur un triangle équilatéral et comme générateur la courbe



ce qui donne l'algorithme

$$\begin{aligned} \text{tapis}(n,l) &= \text{répète}(3, [\text{tapistr}(n,l) \text{ tourne}(120)]) \\ \text{tapistr}(n,l) &= \text{si}(n=0) \text{ alors } [\text{avance}(l)] \\ &\quad \text{sinon } [\text{tapistr}(n-1,l) \text{ tourne}(120) \\ &\quad \quad \text{tapistr}(n-1,l) \text{ tourne}(-120) \\ &\quad \quad \text{tapistr}(n-1,l) \text{ tourne}(-120) \\ &\quad \quad \text{tapistr}(n-1,l) \text{ tourne}(120) \\ &\quad \quad \text{tapistr}(n-1,l)] \end{aligned}$$

et le téragone



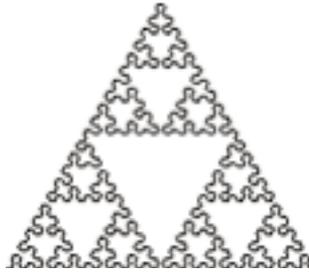
Une troisième méthode consiste à tracer la courbe ayant pour initiateur un segment et comme générateur la courbe



ce qui donne l'algorithme

$$\begin{aligned} \text{tapis}(n,l) &= \text{tapistr}(n,l,1) \\ \text{tapistr}(n,l,k) &= \text{si}(n=0) \text{ alors } [\text{avance}(l)] \\ &\quad \text{sinon } [\text{tourne}(60k) \text{ tapistr}(n-1,l,-k) \\ &\quad \quad \text{tourne}(-60k) \text{ tapistr}(n-1,l,k) \\ &\quad \quad \text{tourne}(-60k) \text{ tapistr}(n-1,l,-k) \text{ tourne}(60k)] \end{aligned}$$

et le téragone



*Remarque.* Les téragones obtenus dans les trois méthodes sont très différents. Néanmoins, avec un peu d'imagination, on peut concevoir que leurs limites sont les mêmes.

### Complément

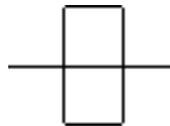
On obtient les téragones d'un tapis carré en prenant comme initiateur un segment et comme générateur la courbe



## 7. La courbe de Peano

La courbe introduite au paragraphe 2 est un exemple de courbe recouvrant un carré. Néanmoins on peut lui reprocher que beaucoup de segments sont parcourus plusieurs fois, ce qui nécessite beaucoup de temps et de mémoire. Dans ce paragraphe, on verra que pour d'autres raisons ce problème a été étudié il y a plus de 100 ans par les plus grands mathématiciens de l'époque.

En 1890, Peano a fourni une courbe que l'on peut obtenir en prenant pour initiateur un segment et pour générateur la courbe



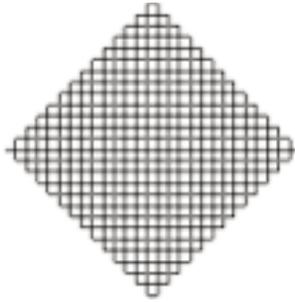
Elle correspond à l'algorithme

```

peano( $n, l$ ) = si ( $n = 0$ )  alors  [ $\text{avance}(l)$ ]
                        sinon  [ $\text{peano}(n-1, l/3)$  tourne(90)
                                répète( $3, [\text{peano}(n-1, l/3)$  tourne(-90)]])
                                répète( $3, [\text{peano}(n-1, l/3)$  tourne(90)]])
                                peano( $n-1, l/3$ ) tourne(-90)]peano( $n-1, l/3$ )

```

et donne le téragone



qui fournit bien à la limite une courbe recouvrant le carré, mais on remarquera que cette courbe a une infinité de points doubles. On peut éviter ceci en « arrondissant » le générateur :



mais ceci complique beaucoup la fonction sans apporter un nouvel éclairage !

### Complément

À la suite de Peano, plusieurs mathématiciens éminents ont fourni des courbes recouvrant des polygones :

- en 1891, Hilbert propose une courbe recouvrant un carré, sans point double, mais non fermée ;
- en 1912, Sierpinski propose une courbe fermée ayant la même propriété ;
- en 1905, Cesaro, puis Polya en 1913 proposent des courbes recouvrant un triangle rectangle isocèle.

### Conclusion

Il reste un sujet important à traiter, celui des recouvrements du plan par des surfaces fractales ou même par des surfaces polygonales non convexes. Ce domaine est suffisamment vaste pour faire l'objet d'un nouvel article. Mais on voit déjà tout l'intérêt de la démarche proposée pour construire et conjecturer des propriétés inattendues d'objets inhabituels. De là à les démontrer...