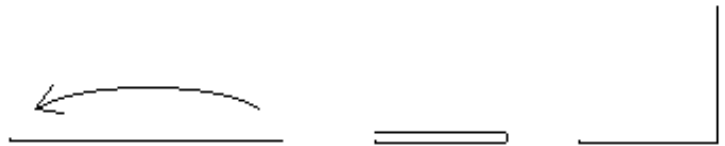


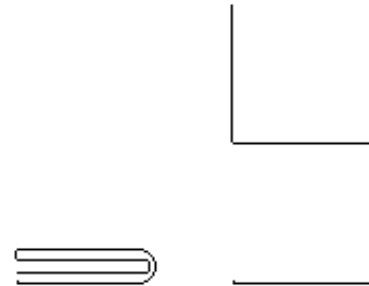
# La symétrie perturbée

14 avril 2020

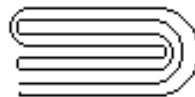
On plie une feuille de papier en deux puis on la déplie à angle droit



On répète l'opération une seconde fois



puis une troisième fois et on obtient ainsi une ligne brisée de  $2^3 = 8$  côtés.



Si on effectue  $n$  pliages on obtient alors une ligne brisée de  $2^n$  côtés, c'est un nombre qui grandit très vite avec  $n$ .

Un mobile parcourant la ligne brisée correspondante à un pliage simple effectuera un mouvement à gauche que l'on peut symboliser par la lettre **g**. Un double pliage consiste alors à encadrer le **g** du second pliage par le **g** du premier tirage et son symétrique **d**. Le mouvement du mobile peut être symboliser par le mot **g g d**. De même un troisième pliage consiste à encadrer **g** par **ggd** et son symétrique **gdd** (symétrie que l'on obtient en inversant la chaîne et remplaçant **g** par **d** et **d** par **g**) d'où la séquence **ggd g gdd**. Ainsi au quatrième pliage on obtient **ggddgdd g gddggdd**.

De façon générale, si  $m_n$  est le mot correspondant à  $n$  pliages alors  $m_n = m_{n-1}gs(m_{n-1})$ ,  $s$  étant l'opérateur de symétrie défini antérieurement. L'ajout de cet élément central  $g$  dans le mot  $m_ngs(m_n)$  perturbe la symétrie. Lorsque la feuille de papier est repliée un nombre important de fois, apparaît alors une courbe d'allure très complexe appelée « courbe du dragon ».

Si dans cette chaîne il n'y avait pas cet élément central **g** alors la suite des chaînes serait formée des éléments **g**, puis **gd** puis de nouveau par symétrisation **gdgd** puis **gdgdgdgd** et ce qui contrairement à ce qui précède ne produit qu'une alternance de **d** et de **g**.

Le programme **sp.java** génère une courbe du dragon sans mémorisation de la chaîne **ggddgddg...** correspondante. Il utilise deux procédures récursives croisées.

```
public void sn (int n, Graphics g) {
    if ( n > 0) {
        sn (n - 1, g);
        gauche (g);
        snm1 (n - 1, g);
    }
}
```

correspondant à (1)  $m_n = m_{n-1}gs(m_{n-1})$  et

```
public void snm1 (int n, Graphics g) {
    if ( n > 0) {
        sn (n - 1, g);
        droite (g) ;
        snm1 (n - 1, g) ;
    }
}
```

correspondant à  $s(m_n) = m_{n-1}ds(m_{n-1})$ ; en effet on peut remarquer que d'après (1) :  
 $s(m_n) = s(m_{n-1}gs(m_{n-1})) = s(s(m_{n-1})s(g)s(m_{n-1})) = m_{n-1}ds(m_{n-1})$  car  $s(g) = d$ ,  $s(ab) = s(b)s(a)$  et  $s(s(a)) = a$  pour tous mots  $a$  et  $b$ .

Remarque : un nombre important de pliages augmente le temps de tracé de façon sensible. La courbe prend alors une place énorme on doit alors diminuer le pas du tracé (qu'on peut même prendre égal à 0.1). Attention si le nombre de pliages dépasse 15, cela risque d'être long.

[ D'après "Principes de la symétrie perturbée" de Michel Mendes-France - l'ouvert n°49 de décembre 1987 - journal de l'A.P.M.E.P d'alsace et de l'I.R.E.M. de Strasbourg ]