



Algorithmique : un fractale aléatoire

Seconde

Fiche Élève

Auteur : R M

On considère, dans le plan rapporté à un repère orthonormé $(O; \vec{i}, \vec{j})$, 3 points distincts A_1 , A_2 et A_3 définis par leurs coordonnées, respectivement (x_1, y_1) , (x_2, y_2) et (x_3, y_3) .

1 - À quelle condition les points A_1 , A_2 et A_3 ne sont pas alignés ? Vérifier que cette condition est satisfaite si l'on choisit

$$(x_1, y_1) = (-7.5, -3.8), \quad (x_2, y_2) = (9.81, -6.19), \quad (x_3, y_3) = (8.15, 14.51)$$

Les points A_1 , A_2 et A_3 forment donc un triangle non aplati noté (T) . On choisit maintenant un point quelconque M défini par ses coordonnées x_M et y_M , par exemple

$$x_M = 4 \quad \text{et} \quad y_M = 2$$

En fait, le point M va bouger. Ce que nous venons de définir est sa position initiale.

2 - Le point M se déplacera de sa position actuelle, quelle qu'elle soit, à la suivante comme suit :

- ✓ On tire au hasard l'un des sommets du triangle (T) .
- ✓ On prend le milieu du segment qui joint la position actuelle de M à ce sommet.
- ✓ Ce milieu sera la nouvelle position de M .

Nous voulons répéter ce procédé 10000 fois, tracer le triangle (T) , la position initiale de M et les 10000 positions suivantes pour voir ce qu'il se passe, à l'aide d'un logiciel de calcul comme « scilab ».

Par exemple, au début du processus, la position actuelle de M est sa position initiale. Si le sommet tiré est A_2 , le milieu du segment MA_2 est la nouvelle position de M , et ainsi de suite.

Passons donc à l'écriture de l'algorithme. La saisie des données ne pose pas de problème :

Listing 1 – source scilab

```
// Un fractale aléatoire
clear ;
clf ;
// Entrée des données :
Absc=[-7.5,9.81,8.15]; // Abscisses de A1, A2 et A3.
Ordo=[-3.8,-6.19,14.51]; // Ordonnées de A1, A2 et A3. Ces points distincts
ne doivent pas être alignés.
xM=4; // Abscisse de M.
yM=2; // Ordonnée de M.
coM=[xM,yM]; // Conservation des coordonnées initiales de M.
```

2.a - Comment peut-on tirer au hasard un sommet de (T) à l'aide de la commande « tirage_entier » ?

2.b - Programmer les 10000 déplacements de M à l'aide d'une boucle « pour ». On prendra soin de stocker les abscisses et les ordonnées de chaque nouvelle position calculée, afin de préparer les tracés.

2.c - Programmer les tracés demandés.

2.d - Vérifier votre algorithme en le comparant à l'algorithme « Fractale aléatoire » téléchargeable ci-dessous. Expliquer notamment la commande du tracé de (T) .

Listing 2 – Fractale aléatoire

```
// Un fractale aléatoire
clear;
clf;
// Entrée des données :
Absc=[-7.5,9.81,8.15];// Abscisses de A1, A2 et A3.
Ordo=[-3.8,-6.19,14.51];// Ordonnées de A1, A2 et A3. Ces points distincts ne
doivent pas être alignés.
xM=4;// Abscisse de M.
yM=2;// Ordonnée de M.
coM=[xM,yM];// Conservation des coordonnées initiales de M.
//Traitement des données :
X=[];// Début du stockage des abscisses des positions successives de M,
position initiale non comprise.
Y=[];// Début du stockage des ordonnées des positions successives de M,
position initiale non comprise.
// Traitement des données : programmation du mouvement de M.
for i=1:10000, // On va faire apparaître 10000 positions de M,
position initiale non comprise.
    j=tirage_entier(1,1,3);// Tirage au hasard d'un sommet de (T).
    xM=(xM+Absc(1,j))/2;// Abscisse de la nouvelle position de M.
    X=[X,xM];// Stockage de la nouvelle valeur de xM.
    yM=(yM+Ordo(1,j))/2;// Ordonnée de la nouvelle position de M.
    Y=[Y,yM];// Stockage de la nouvelle valeur de yM.
end
// Sorties :
orthonorme;// Cette commande impose que le repère soit orthonormé.
plot(X,Y,"r+")// Tracé des différentes positions du point M à l'aide de
signes + de couleur rouge.
plot(coM(1,1),coM(1,2),"b.");// Tracé de la position initiale de M à l'aide
d'un point bleu.
plot([Absc,Absc(1,1)],[Ordo,Ordo(1,1)],"k")// Tracé de (T) en noir.
```

4 - Exécuter l'algorithme. Commenter le résultat.

