Classe : TS, spécialité Maths

Durée : 1h30

Thèmes :
_ Matrices et opérations
_ programmation (fonctions, tracé de courbes, boucle for, instruction conditionnelle.

Date : Décembre 2017

Transformations de points dans le plan à l'aide de matrice

Programmer et dessiner des fractales

(TP adapté du manuel Math'x, enseignement de spécialité, édition 2012, page 137)

Logiciels :
GeoGebra (Partie A)
Python (Partie B, C et D)

Objectifs :
_ Découvrir que les matrices peuvent représenter des transformations géométriques. Certaines transformations sont connues (partie A avec GeoGebra : Symétries), et d'autres sont plus complexes (Partie B, C et D avec Python).
_ Nous allons voir comment la répétition de transformations, intégrées dans des algorithmes puis programmées avec Python, permet de tracer des figures géométriques appelées fractales.

Enoncé :
Comment modéliser sur ordinateur, à l'aide de matrices, différentes plantes pour créer des décors numériques par exemple ?

Partie A : Exemples de transformations connues représentées par des matrices

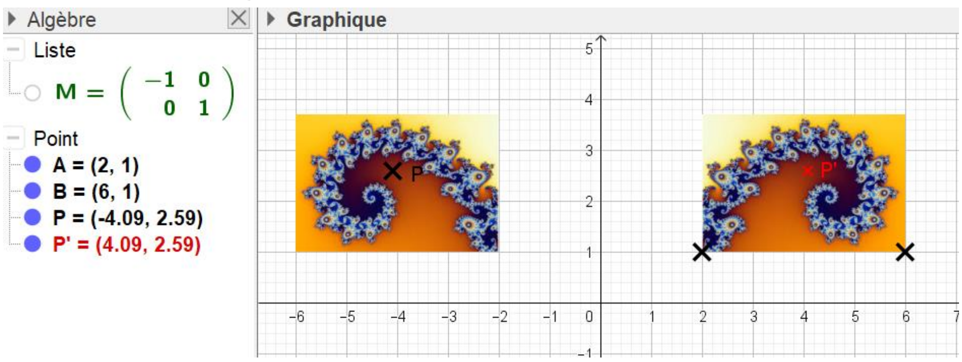
1. Ouvrir le logiciel GeoGebra.
   a. Insérer une image de votre choix, par défaut nommée *image1* par le logiciel.
   b. Entrer A=(2 ,1) comme coordonnées du coin inférieur gauche et choisir B=(6 ,1) celles du coin inférieur droit, afin de régler la position et la taille de l'image.

2. Une première transformation

a) Entrer la matrice $M = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}$ en entrant dans la zone de saisie **M={{-1,0},{0,1}}.**

b) Transformer votre « image1 » par la matrice M en écrivant dans la zone de saisie AppliquerMatrice[M,image1].
Quelle transformation semble-t-elle avoir été appliquée à l'image de départ ?

c) Créer un point P dans le plan. On transforme le point P en un point P' par la matrice M, en écrivant dans la zone de saisie P'=M*P.
On obtient alors un écran analogue à l'écran ci-dessous :



d) Déplacer le point P sur l'image et observer P'. Que peut-on remarquer sur les coordonnées de P et P' ?

e) Soit $P(x\,;y)$ et $P'(x'\,;\;y')$. L'écriture P'=M*P siginifie alors que : $\begin{pmatrix} x' \\ y' \end{pmatrix} = M \times \begin{pmatrix} x \\ y \end{pmatrix}$

Exprimer x' et y' en fonction de x et y et justifier l'effet de la matrice M sur l'image.

3. D'autres transformations connues

En modifiant la matrice M sur le logiciel, reconnaître successivement les transformations associées aux matrices $M = \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix}$, puis $M = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$ et justifier à l'aide d'un produit matriciel.

Partie B : Autres transformations et représentation de la fractale « dragon de Heighway »

On prend dans cette partie : $M = \begin{pmatrix} 0{,}5 & -0{,}5 \\ 0{,}5 & 0{,}5 \end{pmatrix}$.

1. On considère le point P''(x'' ; y'') tel que : $\begin{cases} x'' = 0{,}5x - 0{,}5y + 1 \\ y'' = 0{,}5x + 0{,}5y - 3 \end{cases}$.

   Montrer que le système est équivalent au produit matriciel : $\begin{pmatrix} x'' \\ y'' \end{pmatrix} = M \times \begin{pmatrix} x \\ y \end{pmatrix} + V$;

   où M est la matrice ci-dessus et V une matrice colonne à préciser.

2. **La courbe du dragon (ou dragon de Heighway)**

   On construit une suite de points de la façon suivante :

- Le premier point est O(0 ; 0).
- A partir de ce point O, chaque point est obtenu à partir de son prédécesseur en appliquant une transformation $f_1$ou f_2 . Les transformations f_i sont associées à une relation matricielle de la forme $\begin{pmatrix} x'' \\ y'' \end{pmatrix} = M_i \times \begin{pmatrix} x \\ y \end{pmatrix} + V_i$.
- Pour chaque nouveau point, on choisit au hasard et de façon équiprobable l'une des deux transformations $f_1$ou f_2 suivantes :

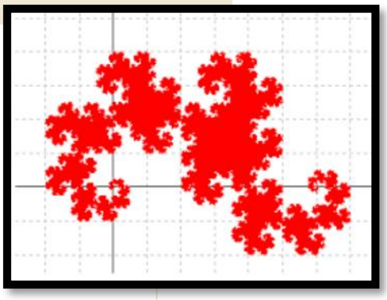
  Pour f_1 : $M_1 = \begin{pmatrix} 0{,}5 & -0{,}5 \\ 0{,}5 & 0{,}5 \end{pmatrix}$ et $V_1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$; pour f_2 : $M_2 = \begin{pmatrix} -0{,}5 & -0{,}5 \\ 0{,}5 & -0{,}5 \end{pmatrix}$ et $V_2 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$

a) Ecrire un algorithme qui crée une suite de 10 000 points.

b) Pour programmer cet algorithme sur Python, on peut définir deux fonctions pour représenter les deux transformations f_1 et f_2 , qui ont la même probabilité d'être appliquées. On prend comme point initial P(0,0) et on le transforme 10 000 fois pour obtenir la création d'une fractale.

Expliquer les variables et les fonctions du programme suivant, écrit en langage Python :

```
import matplotlib.pyplot as plt
from random import*

nbpoints=10000
# point initial
p = (0, 0)

def transformation_1(p):
    x = p[0]
    y = p[1]
    x1=0.5*x-0.5*y
    y1=0.5*x+0.5*y
    return x1,y1

def transformation_2(p):
    x = p[0]
    y = p[1]
    x1=-0.5*x-0.5*y+1
    y1=0.5*x-0.5*y
    return x1,y1

def transforme(p):
    # Choix aléatoire (avec équiprobabilité) entre les 2 transformations de fonctions
    tirage=random()
    if tirage <1/2 :
        x, y = transformation_1(p)
    else :
        x, y = transformation_2(p)
    return x, y

def construction(p, nbpoints):
    x = [p[0]]
    y = [p[1]]
    for i in range(nbpoints):
        p = transforme(p)
        x.append(p[0])
        y.append(p[1])
   # Graphique
    plt.plot(x, y, 'o')
    plt.title('Dragon de Heighway')
    plt.show()

construction(p, nbpoints)
```



c) Ecrire ce programme et l'exécuter pour obtenir le dragon de Heighway.

Partie C : La fougère de Barnsley (1998)

Le Mathématicien anglais Michael Barnsley a décrit comment, à partir d'un point, créer des figures en forme de fougères en répétant un grand nombre de fois une transformation simple sur ce point.

- Le premier point est O(0 ; 0).
- A partir de ce point O, chaque point est obtenu à partir de son prédécesseur en appliquant une transformation f_1 avec une probabilité de 0,01 ; f_2 avec une probabilité de 0,85 ; f_3 avec une probabilité de 0,07 ; ou f_4 avec une probabilité de 0,07.
- Les transformations f_i sont associées à une relation matricielle de la forme $\begin{pmatrix} x'' \\ y'' \end{pmatrix} = M_i \times \begin{pmatrix} x \\ y \end{pmatrix} + V_i$.

Pour f_1 : $M_1 = \begin{pmatrix} 0 & 0 \\ 0 & 0{,}16 \end{pmatrix}$ et $V = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$; pour f_2 : $M_2 = \begin{pmatrix} 0{,}85 & 0{,}04 \\ -0{,}04 & 0{,}85 \end{pmatrix}$ et $V = \begin{pmatrix} 0 \\ 1{,}6 \end{pmatrix}$

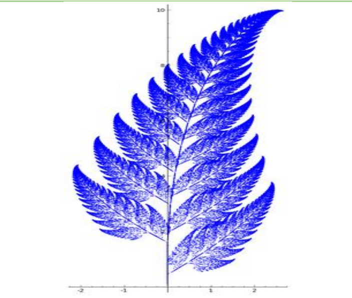
Pour f_3 : $M_3 = \begin{pmatrix} 0{,}2 & -0{,}26 \\ 0{,}23 & 0{,}22 \end{pmatrix}$ et $V = \begin{pmatrix} 0 \\ 1{,}6 \end{pmatrix}$; pour f_4 : $M_2 = \begin{pmatrix} -0{,}15 & 0{,}28 \\ 0{,}26 & 0{,}24 \end{pmatrix}$ et $V = \begin{pmatrix} 0 \\ 0{,}44 \end{pmatrix}$

Pour construire cette forme géométrique qui ressemble à une fougère :

1. Ecrire les 4 relations matricielles sous forme de systèmes.
2. Modifier le programme de la partie B pour tracer la fougère de Barnsley.
3. Modifier le nombre de points tracés et constater les effets sur la fougère.

Remarque : Chacune des 4 transformations est responsable de la création d'une partie de la fougère.

f_1 participe à créer la tige. f_2, la plus probable, crée la tige et les feuilles, f_3 et f_4 créent les feuilles de gauche et de droite de la fougère.



Partie C : Une fractale en forme d'arbre

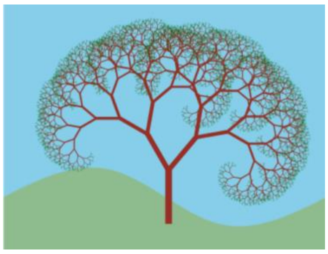
On reprend le processus d'un grand nombre de répétitions, cette fois-ci de 3 transformations géométriques, équiprobables.

On pose c = 0,255 ; r = 0,75 ; q = 0,625 ; $A_1 = -\frac{\pi}{8}$; $A_2 = \frac{\pi}{5}$.

Pour f_1 : $M_1 = \begin{pmatrix} 0 & 0 \\ 0 & c \end{pmatrix}$ **et** $V = \begin{pmatrix} 0,5 \\ 0 \end{pmatrix}$

Pour f_2 : $M_2 = \begin{pmatrix} r\cos(A_1) & -r\sin(A_1) \\ r\sin(A_1) & r\cos(A_1) \end{pmatrix}$ **et** $V = \begin{pmatrix} 0,5 - 0,5r\cos(A_1) \\ c - 0,5r\sin(A_1) \end{pmatrix}$

Pour f_3 : $M_3 = \begin{pmatrix} q\cos(A_2) & -r\sin(A_2) \\ q\sin(A_2) & r\cos(A_2) \end{pmatrix}$ **et** $V = \begin{pmatrix} 0,5 - 0,5q\cos(A_2) \\ 0,6c - 0,5q\sin(A_2) \end{pmatrix}$.



1. Ecrire les 3 relations matricielles sous forme de systèmes.
2. Modifier le programme de la partie B pour tracer la nouvelle fractale.
3. Modifier le nombre de points tracés et constater les effets sur l'arbre.