ISM MPSI/PCSI, Informatique
Année 2024/2025

Corrigé du TP Informatique 25

Exercice 1

On saisit :

```
def arbre(n):
    if n==0:
        return [[[0,1j]]]
    else:
        res=arbre(n-1)
        aux=[]
        for branche in res[-1]: # on parcourt la derniere liste de branches
            a,b=branche
            c=b+(b-a)*2/3*np.exp(1j*np.pi/6)
            d=b+(b-a)*2/3*np.exp(-1j*np.pi/8)
            e=b+(b-a)*1/2*np.exp(-1j*np.pi/5)
            aux.append([b,c])
            aux.append([b,d])
            aux.append([b,e])
        res.append(aux)
        return res

test=arbre(7)
for liste_branche in test:
    for branche in liste_branche:
        deb,fin=branche
        plt.plot([deb.real,fin.real],[deb.imag,fin.imag],'b')
plt.show()
```

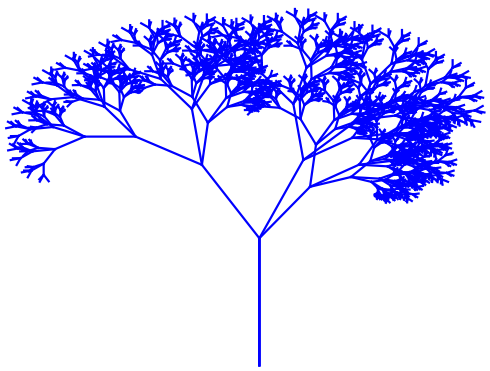

FIGURE 1 – Arbre récursif au bout de 7 étapes

Exercice 2

1. On saisit :

```
def ajout_triangle(Lx,Ly):
    [x1,x2]=Lx
    [y1,y2]=Ly
    x3=2/3*x1+1/3*x2
    y3=2/3*y1+1/3*y2
    x5=1/3*x1+2/3*x2
    y5=1/3*y1+2/3*y2
    x4=1/2*(x1+x2)+1/(2*np.sqrt(3))*(y1-y2)
    y4=1/2*(y1+y2)+1/(2*np.sqrt(3))*(x2-x1)
    return [x1,x3,x4,x5,x2],[y1,y3,y4,y5,y2]
```

2. On saisit :

```
def flocon(n,Lx,Ly):
    if n==0:
        return Lx,Ly
    else:
        Lx,Ly=ajout_triangle(Lx,Ly)
        Fx,Fy=[],[]
        for k in range(4):
            Zx,Zy=flocon(n-1,Lx[k:k+2],Ly[k:k+2])
            Fx+=Zx
            Fy+=Zy
        return Fx,Fy
```

3. On saisit :

```
n=5
xB,yB=[0,0]
xA,yA=[1,0]
tx,ty=flocon(n,[xB,xA],[yB,yA])
plt.plot(tx,ty,'b')
plt.show()
```

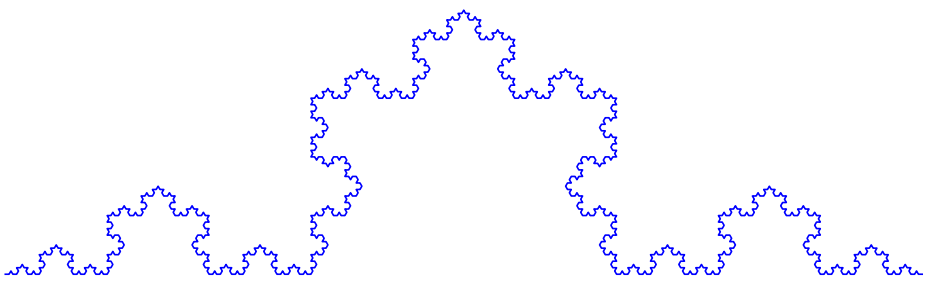

FIGURE 2 – Flocon de Koch

Exercice 3

1. Au rang n, on appelle la fonction au rang $n-1$. On récupère le dernier point obtenu, puis on parcourt en ordre décroissant tous les points, on calcule leur image par la rotation, et on les ajoute à la liste de points.

```
def dragon(n):
    if n==0:
        return [[0,0],[1,0]]
    else:
        L=dragon(n-1)
        [xc,yc]=L[-1] # centre de la rotation
        for k in range(len(L)-2,-1,-1):
            [x,y]=L[k]
            L.append([xc+y-yc,yc+xc-x])
        return L
```

2. Il suffit d'appeler la fonction précédente et de tracer les segments joignant les points obtenus.

```
L=dragon(n)
Lx=[L[k][0] for k in range(len(L))]
Ly=[L[k][1] for k in range(len(L))]
plt.plot(Lx,Ly,'b-')
plt.show()
plt.axis('equal')
plt.axis('off')
```

Par exemple, aux étapes 10 et 13, on obtient :

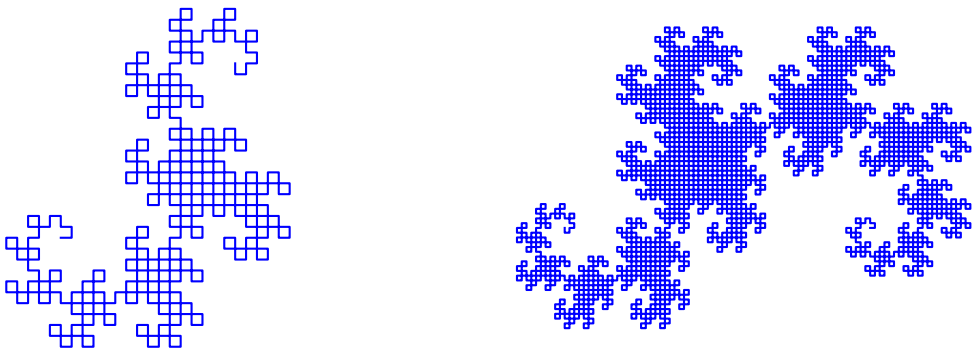

FIGURE 3 – Courbe du dragon