Spirale polygonale
Algorithme

1 Spirale polygonale

1.1 Définition

Définition 1 : Une spirale polygonale est engendrée par une suite de points M_n d'affixe z_n tels que : $z_{n+1} = ke^{i\theta} z_n$ avec z_0 donné.

- k : coefficient de réduction ($0 < k < 1$)
- θ : angle de la rotation.
- n : nombre de rotations

Remarque : Une spirale polygonale ressemble à une coquille d'escargot.

Exemple : Soit (z_n) définie sur $\mathbb{C}$ par : $z_0 = 8$ et $z_{n+1} = \left(\dfrac{3}{4} + \dfrac{\sqrt{3}}{4} i\right) z_n$.

Déterminer le coefficient de réduction et de l'angle de rotation.

On pose $a = \dfrac{3}{4} + \dfrac{\sqrt{3}}{4} i$, on a alors : $k = |a| = \dfrac{\sqrt{9+3}}{4} = \dfrac{\sqrt{12}}{4} = \dfrac{2\sqrt{3}}{4} = \dfrac{\sqrt{3}}{2}$

$$\left.\begin{array}{l} \cos\theta = \dfrac{3}{4} \div \dfrac{\sqrt{3}}{2} = \dfrac{\sqrt{3}}{2} \\ \sin\theta = \dfrac{\sqrt{3}}{4} \div \dfrac{\sqrt{3}}{2} = \dfrac{1}{2} \end{array}\right\} \Rightarrow \arg(a) = \theta = \frac{\pi}{6}\ [2\pi]$$

La suite (z_n) est donc définie par $z_0 = 8$ et $z_{n+1} = \dfrac{\sqrt{3}}{2} e^{i\frac{\pi}{6}} z_n$

1.2 Algorithme

Déterminer un algorithme permettant à l'aide de la suite des points $M_n(z_n)$,

- de tracer les segments $[M_0M_1]$, $[M_1M_2]$, $[M_2M_3]$, …, $[M_{n-1}M_n]$
- de tracer les segments : $[OM_1]$, $[OM_2]$, $[OM_3]$,…, $[OM_n]$
- puis de déterminer la somme ℓ_n des longueurs de ces segments :
$$\begin{aligned} \ell_n &= M_0M_1 + M_1M_2 + M_2M_3 + \cdots + M_{n-1}M_n \\ &= |z_1 - z_0| + |z_2 - z_1| + |z_3 - z_2| + \cdots + |z_n - z_{n-1}| \end{aligned}$$

On peut proposer la fonction spirale(n) en Python .

- le nombre complexe i se rentre avec $1j$

- On crée la fonction exponentielle complexe $\mathrm{e}(x)$ puis la fonction $\mathrm{f}(z)$ qui donne z_{n+1} en fonction de z_n.
- On définit les paramètres de repère, puis pour tracer le point M_n on sépare z en partie réelle X et partie imaginaire Y.
- u permet de stocker l'ancienne valeur de z pour calculer la somme $s = \ell_n$.
- On relie les points entre eux et l'on donne enfin la valeur de $s = \ell_n$

spirale($8+0j$, 50)
$\to \quad s = 29,834$

spirale($8+0j$, 100)
$\to \quad s = 29,856$

spirale($8+0j$, 200)
$\to \quad s = 29,856$

spirale($8+0j$, 500)
$\to \quad s = 29,856$

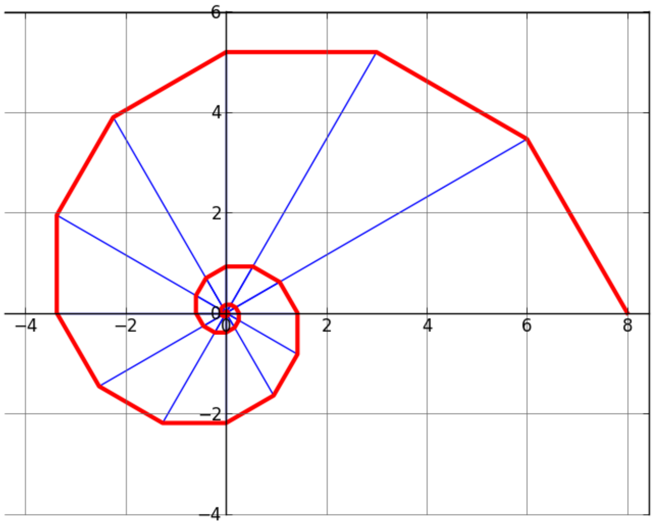
Conjecture : $\lim\limits_{n\to+\infty} \ell_n = 29,856$

```python
from math import*
import matplotlib.pyplot as plt

def e(x):
    return cos(x)+1j*sin(x)

def f(z):
    k=sqrt(3)/2
    t=pi/6
    return k*e(t)*z

def spirale(z,n):
    plt.xlim(-6,10)
    plt.ylim(-4,6)
    plt.grid(linestyle="-",color="0.4")
    X=[z.real]
    Y=[z.imag]
    s=0
    for i in range(1,n+1):
        u=z
        z=f(z)
        X.append(z.real)
        Y.append(z.imag)
        plt.plot([0,z.real],[0,z.imag],color="b")
        s=s+abs(z-u)
    plt.plot(X,Y,linewidth=3,color="r")
    plt.show()
    return s
```



1.3 Calcul de la somme

Calcul de la limite de $\ell_n = |z_1 - z_0| + |z_2 - z_1| + |z_3 - z_2| + \cdots + |z_n - z_{n-1}|$

On pose $u_n = |z_{n+1} - z_n|$, on a alors $\ell_n = u_0 + u_1 + u_2 + \cdots + u_{n-1}$

Montrons que la suite (u_n) est géométrique :

$u_{n+1} = |z_{n+2} - z_{n+1}| = |ke^{i\theta}z_{n+1} - ke^{i\theta}z_n| = |ke^{i\theta}||z_{n+1} - z_n| = ku_n$

$\forall n \in \mathbb{N}$, $\dfrac{u_{n+1}}{u_n} = k$, la suite (u_n) est géométrique de raison $q = k$
de premier terme $u_0 = |z_1 - z_0|$

ℓ_n est la somme des n premiers termes de la suite géométrique u_n, donc :

$\ell_n = u_0 \times \dfrac{1-k^n}{1-k}$ or $\lim\limits_{n\to+\infty} k^n = 0$ car $0 < k < 1$ donc $\lim\limits_{n\to+\infty} \ell_n = \dfrac{u_0}{1-k}$

Si $z_0 = 8$ et $z_{n+1} = \left(\dfrac{3}{4} + \dfrac{\sqrt{3}}{4}i\right) z_n$, on a alors :

$$|z_1 - z_0| = \left|8\left(\frac{3}{4} + \frac{\sqrt{3}}{4}i\right) - 8\right| = |6 + 2\sqrt{3}i - 8| = |2 + 2\sqrt{3}i| = \sqrt{4+12} = 4$$

$$\lim_{n\to+\infty} \ell_n = \frac{u_0}{1 - \dfrac{\sqrt{3}}{2}} = \frac{8}{2-\sqrt{3}} = 8(2+\sqrt{3}) \approx 29{,}856$$

1.4 Calculatrice Ti-83 python

ÉDITEUR : SPIRALE
LIGNE DU SCRIPT 0011

```
from math import *
import ti_plotlib as plt

def e(x):
  return cos(x)+1j*sin(x)

def f(z):
  k=sqrt(3)/2
  t=pi/6
  return k*e(t)*z

def spirale(z,n):
  plt.cls()
  plt.color(0,0,0)
  plt.pen("thin","solid")
  plt.window(-6,10,-4,6)
  plt.grid(1,1,"solid")
  plt.axes("axes")
  X=[z.real]
  Y=[z.imag]
  s=0
  for i in range(1,n+1):
    u=z
    z=f(z)
    X.append(z.real)
    Y.append(z.imag)
    plt.line(0,0,z.real,z.imag,"")
    s=s+abs(z-u)
  plt.pen("medium","solid")
  plt.color(250,0,0)
  plt_plot(X,Y,"o")
  plt.show_plot()
  return s
```