FRACTALES

Danis Yesenia Gómez Giraldo

Un fractal es un objeto geométrico cuya estructura básica es: fragmentada, en ocasiones aparentemente irregular y se repite a diferentes escalas

Para generar un fractal matemático, o tener una aproximación a un fractal natural se hace un algoritmo de recursividad

¿QUE ES LA RECURSIVIDAD?

proceso en el que una función se llama a sí misma. Así en el programa recursivo, se proporciona la solución al caso base y la solución del problema más grande se expresa en términos de problemas más pequeños.

SE DESTACAN TRES FORMAS DE ALGORITMOS DE RECURSIVIDAD PARA GENERAR ESPACIOS FRACTALES

- **Sistemas de funciones iteradas:** *Unos conjuntos se reemplazan recursivamente por su imagen bajo un sistema de aplicaciones*: el conjunto de Cantor, la alfombra de Sierpinski, el triángulo de Sierpinski, la curva de Peano, la curva del dragón, **el copo de nieve de Koch** o la Esponja de Menger, son algunos ejemplos.
- **Fractales de algoritmos de Escape**: *Definidos por una relación de recurrencia en cada punto del espacio* (por ejemplo, el plano complejo): el conjunto de Mandelbrot, **conjunto de Julia**, y el fractal de Lyapunov.
- **Fractales aleatorios**: *Generados por procesos estocásticos, no deterministas*: el movimiento browniano,el vuelo de Lévy, los paisajes fractales o los árboles brownianos. Estos últimos son producidos por procesos de agregación por difusión limitada..

COPO DE NIEVE DE KOCH

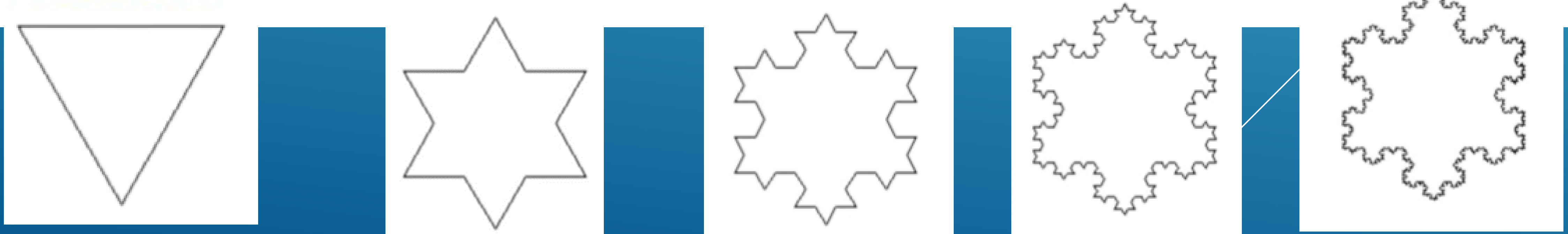
Es una curva cerrada continua pero no diferenciable en ningún punto, descrita por el matemático sueco ***Helge von Koch*** en 1904 en un artículo titulado «Acerca de una curva continua que no posee tangentes y obtenida por los métodos de la geometría elemental

Su construcción : Se toma un segmento, se lo divide en tres partes iguales, se remplaza la parte central por dos partes de igual longitud haciendo un ángulo de 60 grados. Luego, con los cuatro segmentos, se procede de la misma manera, lo que da lugar a 16 segmentos más pequeños en la segunda iteración. Y así sucesivamente.

PROGRAMA DE RECURRENCIA POR SISTEMAS DE FUNCIONES ITERADAS PARA COPO DE NIEVE DE KOCH

```
#creando la funcion para la curva de Von Kosh
def curvaVonkoch(xi, yi, xf, yf, n):

    if n == 0:
        create_line(xi,yi,xf,yf,n)
    elif n > 0:
        x1=xi+(xf-xi)/3.0
        y1=yi+(yf-yi)/3.0

        x3=xf-(xf-xi)/3.0
        y3=yf-(yf-yi)/3.0

        x2=(x1+x3)*cos(pi/3)-(y3-y1)*sin(pi/3)
        y2=(y1+y3)*cos(pi/3)+(x3-x1)*sin(pi/3)

        #llamado de la funcion dentro de la misma función pasando
        #los valores acabados de crear aplicacion de la recurcion
        curvaVonkoch(xi, yi, x1, y1, n-1)
        curvaVonkoch(x1, y1, x2, y2, n-1)
        curvaVonkoch(x2, y2, x3, y3, n-1)

    return
```

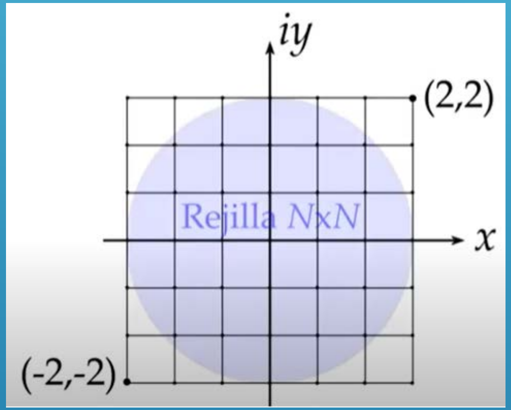
```python
#función para crear copos de  von koch
def copoVonkoch(lado, n):
    #generando primero los lados del triangulo
    x_vertice1=0
    y_Vertice1=0

    x_vertice2=lado*cos(2*pi/3)
    y_vertice2=lado*sin(2*pi/3)

    x_vertice3=lado*cos(pi/3)
    y_vertice3=lado*sin(pi/3)

    #llamando la curva del Von Koch
    curvaVonkoch(x_vertice1, y_Vertice1, x_vertice2, y_vertice2, n)
    curvaVonkoch(x_vertice2, y_Vertice2, x_vertice3, y_vertice3, n)
    curvaVonkoch(x_vertice3, y_Vertice3, x_vertice1, y_vertice1, n)

  return
```



CONJUNTOS DE JULIA

Los conjuntos de Julia, así llamados por el matemático Gaston Julia, son una familia de conjuntos fractales que se obtienen al estudiar el comportamiento de los números complejos al ser iterados por una función de la siguiente forma:

F= $z^2 + c$.



$$z_0 = z$$

$$z_1 = F(z_0) = z_0^2 + c$$

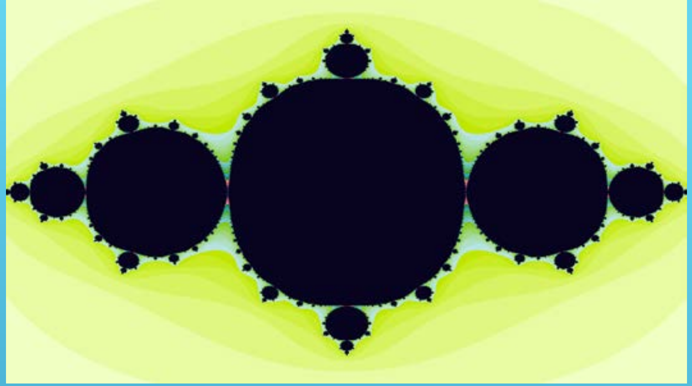
$$z_2 = F(z_1) = z_1^2 + c$$

Así los puntos de Julia determinan los espacios en el plano complejo donde dado un valor c y el conjunto de todos los puntos iniciales z_0, la sucesión converge o se mantiene acotada .

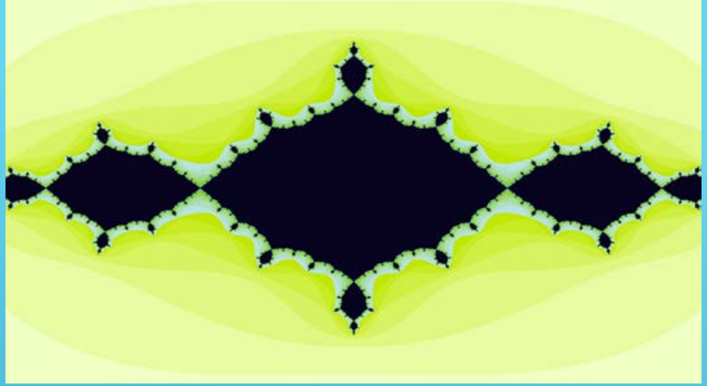
PROGRAMA DE RECURRENCIA POR ALGORITMOS DE ESCAPE PARA SET DE JULIA

```
# configurar las variables de acuerdo con
#la ecuación para crear el fractal
cX, cY =  0 , 0
moveX, moveY = 0.0, 0.0
maxIter = 255
#Desplazamiento dentro del plano complejo bajo los conjuntos de Julia
for x in range(a):
    for y in range(h):
        #coordenadas  en la region compleja donde se establece
        #imaginaria para zy y real para zx
        zx = 1.5*(x - a/2)/(0.5*zoom*a) + moveX
        zy = 1.0*(y - h/2)/(0.5*zoom*h) + moveY
        i = maxIter

        #aplicando sucesion para conjuntos de Julia con   el c dado
        while zx*zx + zy*zy < 4 and i > 1:
            tmp = zx*zx - zy*zy + cX #aplicacion
            zy,zx = 2.0*zx*zy + cY, tmp
            i -= 1
```

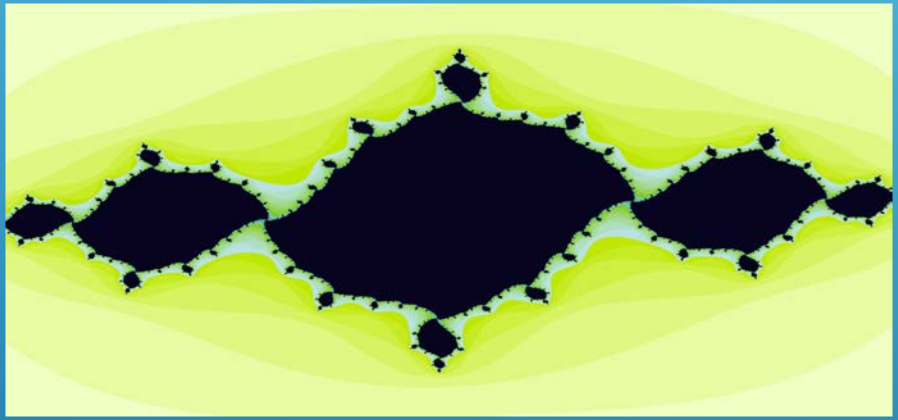
0, 0



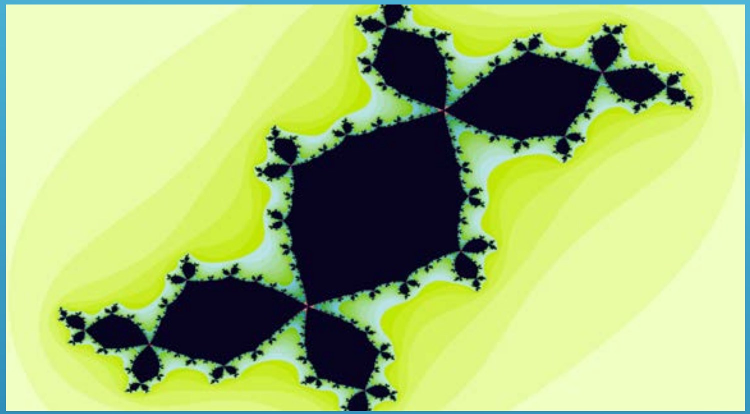
-0.8, 0



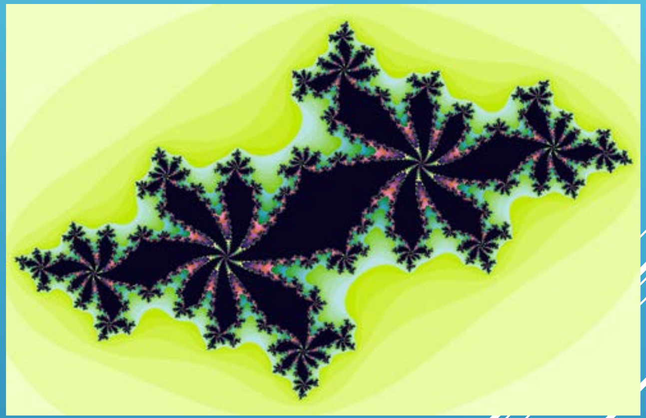
-1.1, 0



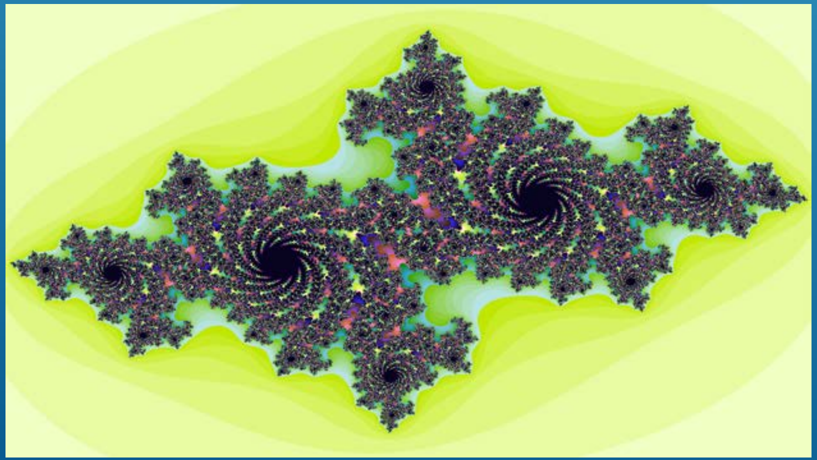
-1, 0.1



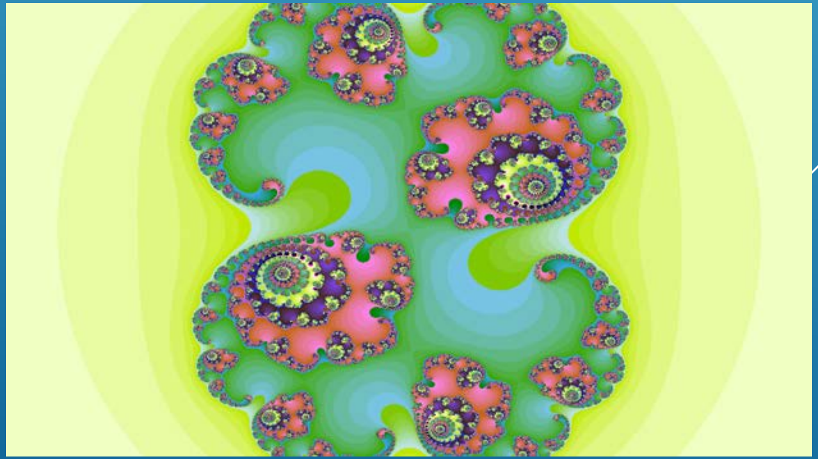
-0.12, 0.75



-0.624 , 0.435



-0.7, 0.27015



0.285, -0.01