

Examen Fractales

March 24, 2025

Nombre y Apellidos: Miriam Bernat Jiménez

0.0.1 Ejercicio 1: Verificar el ejercicio 2 b o c dibujando la solución con el algoritmo determinista de tus prácticas entregadas.

Código común:

```
[ ]: import matplotlib.pyplot as plt

# Implementamos la función para el algoritmo determinista
def fractal_determinista(sfi, pasos, semilla):
    """
    sfi: lista de funciones (SFI)
    pasos: número de iteraciones
    semilla: tupla (x0, y0) con el punto inicial
    """
    # Conjunto inicial con la semilla
    puntos = [semilla]

    for _ in range(pasos):
        nuevos_puntos = []
        # A cada punto actual le aplicamos TODAS las funciones del SFI
        for p in puntos:
            for funcion in sfi:
                nuevos_puntos.append(funcion(p))
        # Actualizamos la lista de puntos con los nuevos
        puntos = nuevos_puntos

    return puntos

# Nuestro main para ejecutar
def ejecucion(sfi, pasos, semilla):
    """
    sfi: lista de funciones (SFI)
    """
    # Ejecutamos el algoritmo determinista
    puntos_resultantes = fractal_determinista(sfi, pasos, semilla)

    # Separamos en listas las coordenadas x e y
```

```

x_vals, y_vals = zip(*puntos_resultantes)

# Graficamos el conjunto de puntos
plt.figure(figsize=(5, 5))
plt.scatter(x_vals, y_vals, s=1, color='blue')
plt.title(f"Fractal determinista con {pasos} pasos")
plt.axis('equal')    # Misma escala en ambos ejes
plt.show()

```

2B:

[27]: *# Definimos las transformaciones según la tabla dada*

```

def f1(p):
    x, y = p
    return (1/8 * x, 1/8 * y)

def f2(p):
    x, y = p
    return (1/8 * (x + 1), 1/8 * y)

def f3(p):
    x, y = p
    return (1/8 * (x + 1), 1/8 * (y + 1))

def f4(p):
    x, y = p
    return (1/8 * (x), 1/8 * (y + 1))

def f5(p):
    x, y = p
    return (1/8 * (x), 1/8 * (y + 7))

def f6(p):
    x, y = p
    return (1/8 * (x + 1), 1/8 * (y + 7))

def f7(p):
    x, y = p
    return (1/8 * (x), 1/8 * (y + 6))

def f8(p):
    x, y = p
    return (1/8 * (x + 1), 1/8 * (y + 6))

def f9(p):
    x, y = p
    return (1/8 * (x + 6), 1/8 * (y + 7))

```

```

def f10(p):
    x, y = p
    return (1/8 * (x + 7), 1/8 * (y + 7))

def f11(p):
    x, y = p
    return (1/8 * (x + 6), 1/8 * (y + 6))

def f12(p):
    x, y = p
    return (1/8 * (x + 7), 1/8 * (y + 6))

def f13(p):
    x, y = p
    return (1/8 * (x + 6), 1/8 * (y + 1))

def f14(p):
    x, y = p
    return (1/8 * (x + 7), 1/8 * (y + 1))

def f15(p):
    x, y = p
    return (1/8 * (x + 6), 1/8 * (y))

def f16(p):
    x, y = p
    return (1/8 * (x + 7), 1/8 * (y))

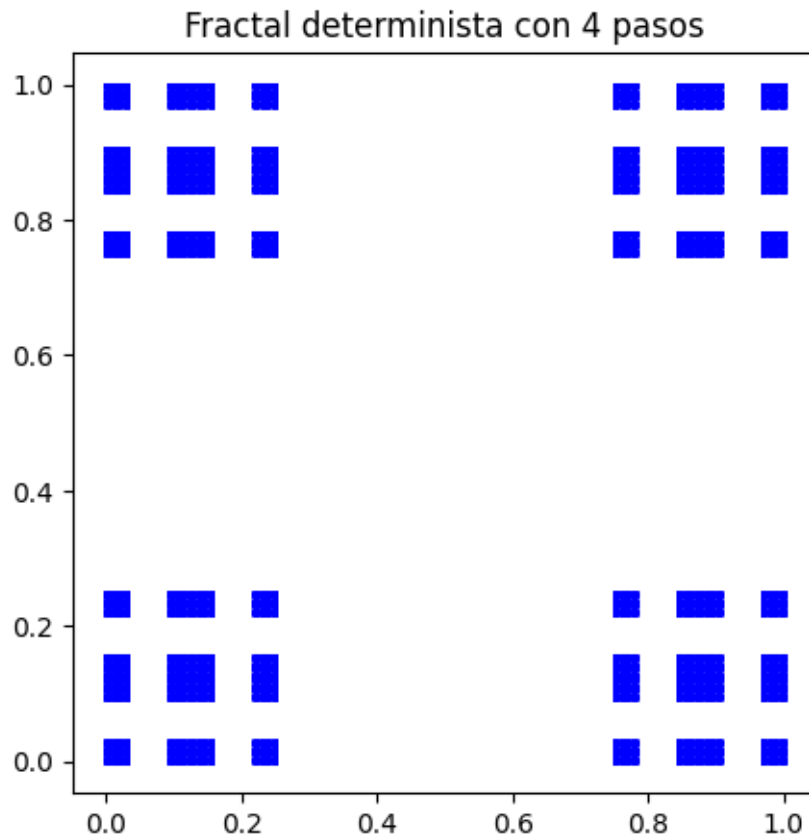
# El SFI es la lista de estas 16 funciones
sfi = [f1, f2, f3, f4, f5, f6, f7, f8, f9, f10, f11, f12, f13, f14, f15, f16]

pasos = 4

semilla = (0,0)

ejecucion(sfi, pasos, semilla)

```



2C:

[11]: *# Definimos las transformaciones según la tabla dada*

```
def f1(p):
    x, y = p
    return (1/4 * (x), 1/4 * (y + 3))

def f2(p):
    x, y = p
    return (1/4 * (x + 1), 1/4 * (y + 1))

def f3(p):
    x, y = p
    return (1/4 * (x + 1), 1/4 * (y + 2))

def f4(p):
    x, y = p
    return (1/4 * (x + 2), 1/4 * (y + 2))

def f5(p):
    x, y = p
```

```

    return (1/4 * (x + 2), 1/4 * (y + 1))

def f6(p):
    x, y = p
    return (1/4 * (x + 3), 1/4 * (y + 3))

def f7(p):
    x, y = p
    return (1/4 * (x + 3), 1/4 * (y))

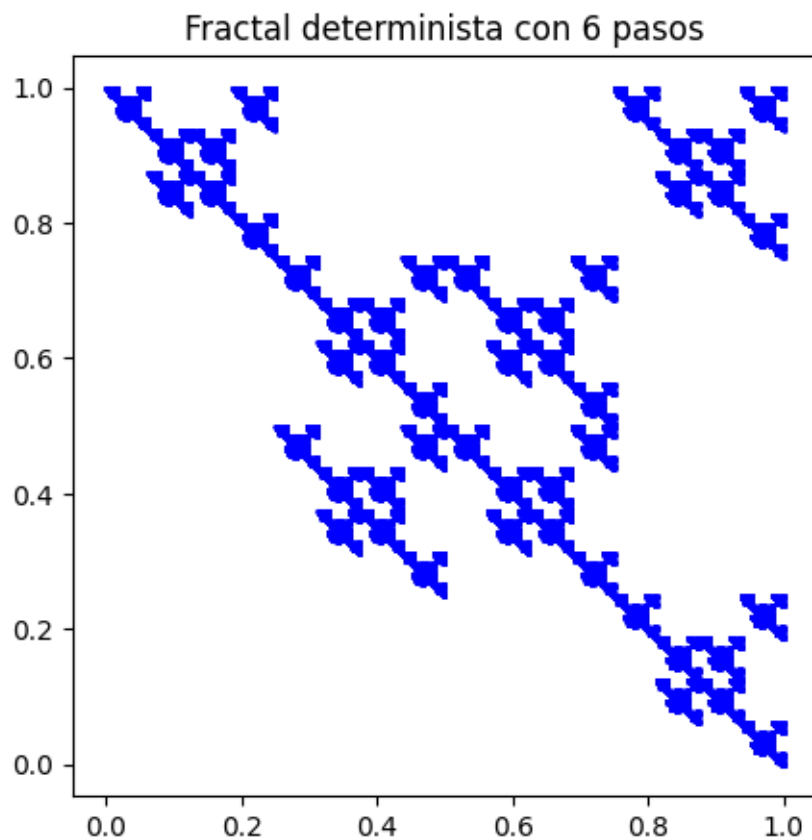
# El SFI es la lista de estas 7 funciones
sfi = [f1, f2, f3, f4, f5, f6, f7]

pasos = 6

semilla = (0,0)

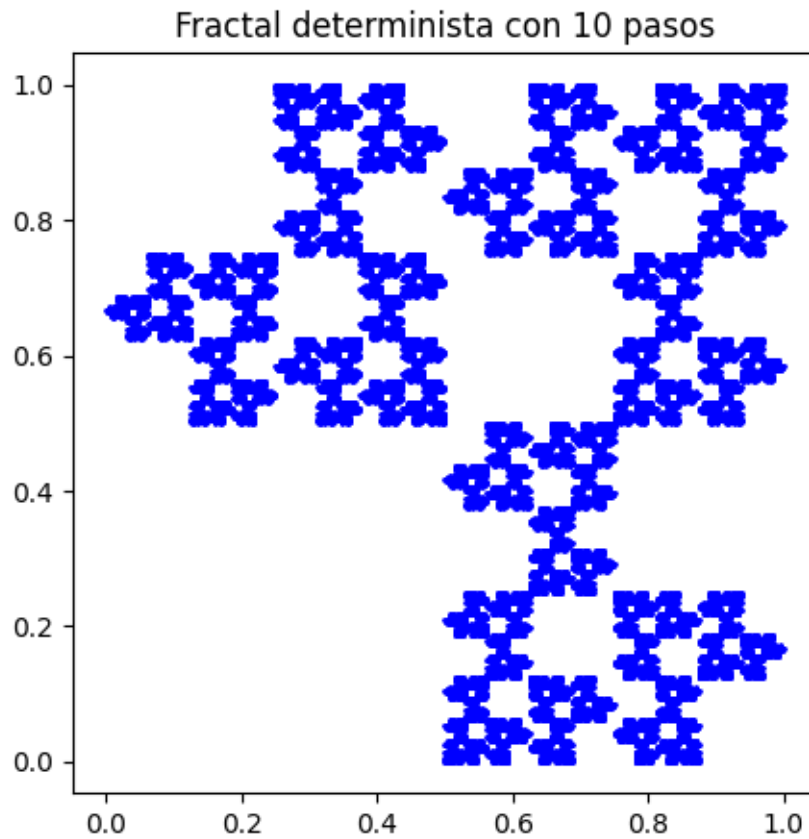
ejecucion(sfi, pasos, semilla)

```



0.0.2 Ejercicio 2: Verificar el ejercicio 3 dibujando la solución con el algoritmo aleatorio de tus prácticas entregadas.

```
[31]: def f1(p):  
      x, y = p  
      return (0.5 * x + 0.5 , 0.5 * y + 0.5)  
  
      def f2(p):  
          x, y = p  
          return (0.5 * x + 0, (-0.5) * y + 1)  
  
      def f3(p):  
          x, y = p  
          return ((-0.5) * x + 1, (-0.5) * y + 0.5) ##Marron  
  
      # El SFI es la lista de estas 3 funciones  
      sfi = [f1, f2, f3]  
  
      pasos = 10  
  
      semilla = (0,0)  
  
      ejecucion(sfi, pasos, semilla)
```



0.0.3 Ejercicio 3: Modificar el algoritmo de desplazamiento del punto medio que entregaste, para que dado un $t > 0$, en cada etapa, el punto de cada segmento de extremos A y B en el que se produce el desplazamiento sea $(t/3)A + ((1-t)/3)B$.

[]: