Relación 5

1. Consideremos como figura iniciadora de un fractal un cuadrado y en cada paso dividamos el cuadrado en n^2 cuadrados iguales. De esos n^2 quitamos k cuadrados y repetimos el proceso *at infinitum*. Calcular la dimensión fractal del objeto así obtenido. Haced lo mismo suponiendo que en vez de un cuadrado tengo un cubo y lo divido en 3^3 cubos menores. Supongamos que quitamos el cubo central y los 6 cubos vecinos proximos. Iterar at infinitum y calcular la dimensión fractal del objeto resultante.

2. Sea el sistema de reacción difusión siguiente:

$$\frac{\partial u}{\partial t} = f(u) + D\frac{\partial^2 u}{\partial x^2}$$

con $f(u) = A(u - u1)(u_2 - u)(u - u_3)$. Encontrar la solución solitónica $U(z \equiv x - ct)$ verificando $U(-\infty) = u_3$ y $U(\infty) = u_1$, con velocidad de ondas

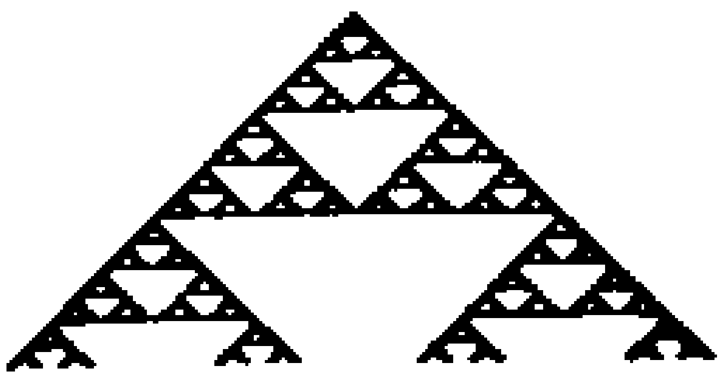
$$c = \left(\frac{AD}{2}\right)^{1/2} (u_1 - 2u_2 + u_3).$$

3. Generar la JS en el ordenador, y estudiar sus propiedades, en particular, convencerse de su auto–semejanza. Hay varias formar de generarlo, veamos algunas, sencillas en principio:

   a) Consideremos un **autómata celular uni-dimensional** que consta de una línea de sitios, cada uno con una variable s_1 capaz de k estados distintos, $0, \ldots, k-1$. Los valores de s_i se actualizan en tiempos discretos de acuerdo con una regla determinista que depende de $2r$ sitios alrededor de i, esto es,

$$a_i^{(t+1)} = \varphi\left[a_{i-r}^{(t)}, a_{i-r+1}^{(t)}, ..., a_{i+r}^{(t)}\right]$$

   la misma $\forall i$. Este modelo presenta un comportamiento extraordinariamente complejo, incluso en el caso más sencillo (y familiar) en el que $k = 2$ (dos estados posibles, 0 ó 1, en cada sitio), y $r = 1$ (sólo los vecinos adyacentes a i influyen en su actualización). Considerando este caso, y partiendo de una línea con un único estado 1 en el centro, encontrad la regla φ que produce la JS:



donde el tiempo progresa desde el vértice superior hacia abajo y se ha representado 0 (blanco) y 1 (negro).[1]

b) El **juego del caos**, que usa números aleatorios. El algoritmo consiste en
   - elegir tres puntos cualesquiera en un papel y etiquetarlos 1, 2 y 3, respectivamente.
   - Elegimos cualquier otro punto como inicial.
   - Sorteamos entre los enteros 1 a 3 con igual probabilidad.
   - Si hemos obtenido $\zeta \in [1, 2, 3]$, hacemos una señal en el papel que sea el punto medio del segmento entre el punto inicial y el punto ζ.
   - Tomamos este punto medio como inicial, y repetimos el proceso (desde el sorteo) muchas veces.

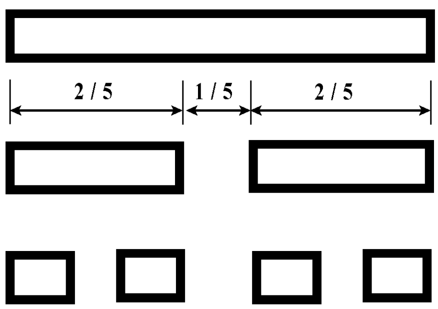
c) Programad este algoritmo en el ordenador y mostrad que el resultado es la JS.[2]

d) Sea el **mapa** (en un espacio) **bi-dimensional** que, partiendo de $m+1$ constantes dadas, $\{a, X_i, Y_i; i = 1, \ldots, m\}$, transforma el punto (x_n, y_n) en el (x_{n+1}, y_{n+1}) de acuerdo con la regla

$$x_{n+1} = ax_n + X_i$$
$$y_{n+1} = ay_n + Y_i,$$

donde el punto $P_i \equiv (X_i, Y_i)$ se elige al azar en cada transformación homogéneamente de entre los m dados, partiendo de un punto (x_1, y_1) arbitrario. El mapa produce patrones variados. Mostrad que, excluyendo los primeros pocos puntos, la JS se sigue para $a = \frac{1}{2}$ y $m = 3$ (si los tres puntos no son co-lineales).[3]

4. Calcular la dimensión del siguiente conjunto de Cantor:



[1]**Solución**: la regla está dada en S. Wolfram, *Rev. Mod. Phys.* **55**, 601-644 (1983), donde está catalogada como 126.

[2]**Solución**: Un programa BASIC detallado está listado en Lam p.229. Descartando los primeros pocos puntos, se obtiene una forma definida, la de la JS, a pesar de que la intuición sugiere que los puntos aleatorios podrían tender a ir cubriendo el papel homogéneamente.

[3]**Solución**: Hay listado un programa muy sencillo en Lam p. 231, donde también se muestra la figura resultante para la elección $P_i = (0{,}3, 0{,}9)\,, (0{,}2, -0{,}9)\,, (-0{,}4, 0{,}6)\,.$

También se muestra allí cómo se obtienen otras figuras, por ejemplo, la **alfombra de Sierpinski** para $a = 1/3$ y $m = 8$, con P_i los vértices y puntos medios de los lados de un cuadrado arbitrario.