Ejercicio de computación 1
Modelos dinámicos discretos

1. **Crear diagrama de telaraña para la ecuación logística**

$$x_{n+1} = \lambda x_n(1 - x_n), \quad n \geq 0, \quad x_0 \text{ dado} \tag{1}$$

Input: $x_0 \in [0, 1], N =$ número de iteraciones computadas, $\lambda \in [0, 4]$.

Diagrama: construir con el comando **line** las rectas que unen los puntos

$$(x_i, x_{i+1}) \longrightarrow (x_{i+1}, x_{i+1}) \longrightarrow (x_{i+1}, x_{i+2}), \qquad \text{para cada } i = 1, \ldots, N$$

Recuérdese que en Matlab/Octave la primera componente de un vector siempre es 1 (no 0).

Nota: es preferible el comando **line** que el **plot** porque no necesita usar el comando **hold on** para que se junten todas las líneas consecutivamente. Si se usa el **plot** hay que poner **hold on** y darle un color para que no pinte cada segmento de un color diferente.

2. **Crear diagrama de bifurcación:** El *diagrama de bifurcación o diagrama de órbitas* ayuda a detectar la existencia de puntos de bifurcación de una familia paramétrica de esquemas iterativos:

$$x_{n+1} = f_\lambda(x_n), \quad n \geq 0, \qquad x_0 \text{ dado}, \qquad \lambda \in [\lambda_{min}, \lambda_{max}]. \tag{2}$$

Input: $x_0, \lambda_{min}, \lambda_{max}, N_l =$número de valores de $\lambda \in [\lambda_{min}, \lambda_{max}]$ que se toman, $N =$número de iteraciones que dará el esquema iterativo, $N_f =$número que decide cuántas de las últimas iteraciones se dibujan. Por ejemplo, si $N = 100$ y $N_f = 10$, se dibujan las iteraciones $x_{91}, x_{92}, \ldots, x_{100}$.

Algoritmo: hacer un bucle que recorra todos los N_l valores de λ, y dentro de este un segundo bucle que realice todas las N iteraciones de (2), arrancando siempre del mismo x_0. Se tiene que construir un vector p de dimensión N_f que almacene las N_f últimas iteraciones para un λ dado, y un vector l de la misma dimensión N_f, cuyos elementos son todos iguales $l = (\lambda, \ldots, \lambda)$ para poder hacer la representación gráfica del vector l frente al p.

Usar el **hold on** para que se dibujen juntas todas las gráficas de todos los valores de λ y ponerlas con estilo punteado ('.' en Matlab u Octave).

Experimentos:

(a) Ecuación logística: $f(x) = \lambda x(1 - x), \lambda \in [0, 4]$, $x_0 = 0.5, N_l = 100, N = 1000, N_f = 50$. Téngase en cuenta que puede tardar más de un minuto en la ejecución. Si se tiene paciencia, probar también con $Nl = 1000, N = 5000$.

(b) Aplicación tienda $f(x) = \lambda x$, si $0 \leq x \leq 0.5$ y $f(x) = \lambda(1 - x)$, si $0.5 \leq x \leq 1$, con $\lambda \in [0, 2]$, $x_0 = 0.5, N_l = 100, N = 1000, N_f = 50$.

3. **Conjuntos de Julia:** estos conjuntos surgen del estudio de los esquemas iterativos en el campo complejo:

$$z_{n+1} = z_n^2 + c, \quad n \geq 0, \quad z_0 \in \mathbb{C} \text{ dado} \tag{3}$$

Si $c = c_1 + ic_2$, $c_1, c_2 \in \mathbb{R}$, $z_n = x_n + iy_n$, realmente (3) es equivalente al sistema no lineal en el plano real

$$x_{n+1} = x_n^2 - y_n^2 + c_1, \quad y_{n+1} = 2x_ny_n + c_2, \quad n \geq 0, \qquad (x_0, y_0) \in \mathbb{R}^2 \text{ dado} \tag{4}$$

Fijado un valor c, se llama **conjunto de Julia** J_c al conjunto de valores iniciales $z_0 \in \mathbb{C}$, tal que la órbita correspondiente está acotada (entendiendo que no tiende a infinito, puede incluir el caso de tener ciclos atractores),

$$J_c = \{z_0 \in \mathbb{C} : \{z_0, z_1, \dots\} \subset B, \text{ con } B \text{ acotado}\}$$

Se puede probar que si hay un valor z_n tal que $|z_n| > |c| + 1$, la órbita no está acotada, por tanto, el z_0 correspondiente no está en J_c.

Para graficar los conjuntos de Julia tenemos que construir el siguiente programa Matlab/Octave, que dibuja los puntos (x, y) de un rectángulo en el plano que pertenecen a un conjunto de Julia determinado.

Input: c_1, c_2, rectángulo $[a, b] \times [c, d]$, NP =número de puntos que tomamos equiespaciados en $[a, b]$ y en $[c, d]$, N =número de iteraciones de (3) que se computan, K =cota del cuadrado del módulo de las iteraciones.

Algoritmo: crear la malla de $NP \times NP$ puntos en el rectángulo $[a, b] \times [c, d]$ y construir dos bucles anidados que recorran cada uno de los puntos del plano (a_i, b_j) de dicha malla rectangular. Para cada uno de estos puntos arrancar un nuevo bucle que calcule la iteración dada en (4) con valor inicial $x_0 = a_i$, $y_0 = b_j$. En cada vuelta del bucle, calcular el módulo al cuadrado de la iteración obtenida $(x_n^2 + y_n^2)$. Si este valor es mayor que un valor moderado K (por ejemplo, $K = 4$), se para la iteración y se prueba con otro punto de la malla. Si el cuadrado del módulo es menor que K, se sigue iterando hasta N. Si se ha llegado a la iteración N, (x_N, y_N) sin que el módulo haya pasado de K, el valor inicial (x_0, y_0) está en el conjunto de Julia. Guardar estos puntos que están en el conjunto de Julia en dos vectores X e Y que almacenarán los valores de las abscisas y las ordenadas de dichos puntos. Finalmente, dibujar todos estos puntos con estilo punteado.

Experimentos: con $[a, b] \times [c, d] = [-1.5, 1.5] \times [-1.5, 1.5]$, $NP = 1000$, $N = 20$, $K = 4$, dibujar los conjuntos de Julia

$$J_{-1},\ J_0,\ J_{0.3},\ J_{-0.125+0.65i},\ J_{0.11+0.66i},\ J_{-0.7+0.1i},\ J_{0.3+0.5i}$$