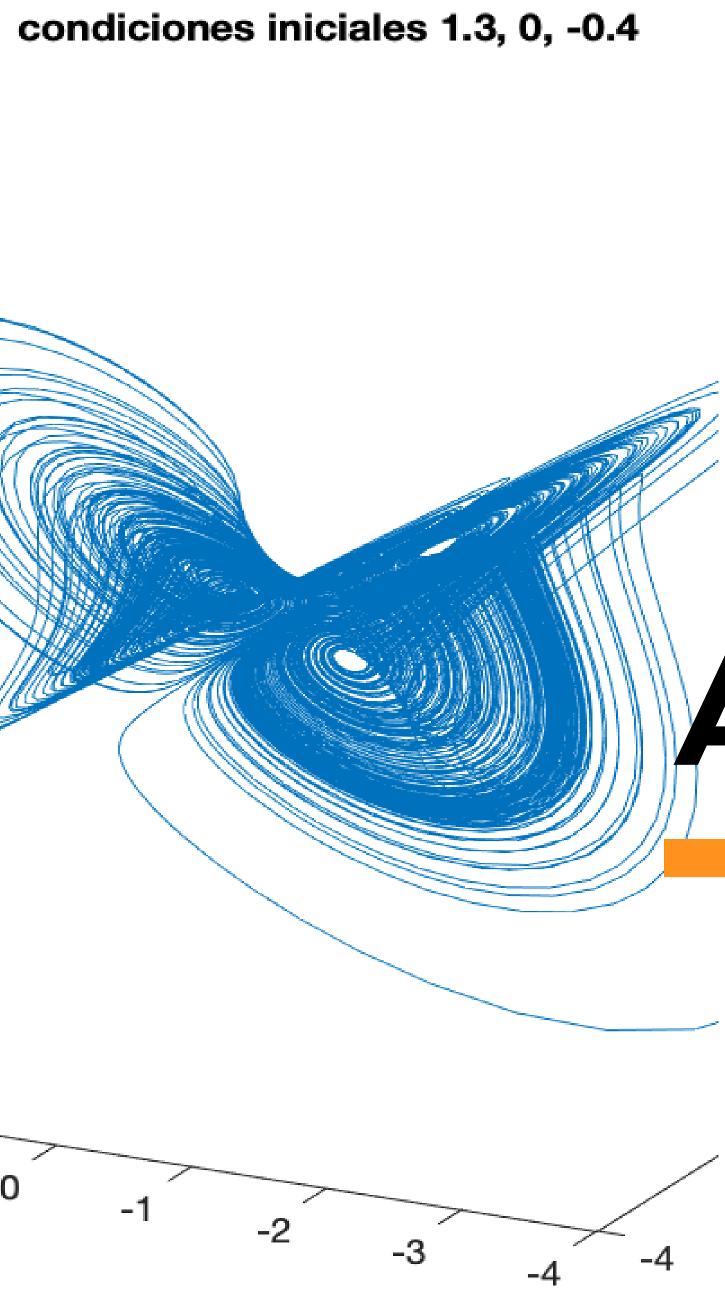

Four Wing Atracttor

Javier Viseras
Miguel Angel Vallejo
Bernardo Ordás

Introducción

El Four-Wing Attractor es un atractor diseñado para estudiar propiedades caóticas ante unas condiciones diferentes. En este proyecto nos centraremos en la estabilidad de sus puntos críticos. Esencialmente, este atractor es un sistema autónomo de ecuaciones diferenciales y nuestro objetivo es realizar un estudio de la estabilidad.

Problema estudiado, donde A, B y C son constantes:

$$(PC) \equiv \begin{cases} x'(t) = Ax + yz \\ y'(t) = Bx + Cy - xz \\ z'(t) = -z - xy \end{cases}$$

Para dicho estudio hemos usado el software MATLAB, que aparte de permitirnos modificar fácilmente las condiciones iniciales, para poder observar así el comportamiento caótico, nos permite realizar una representación gráfica de las soluciones bastante buena.

A continuación se detalla cómo se ha procedido al estudio, los resultados y conclusiones obtenidas.

Procedimiento:

Estudio de los puntos críticos:

```
syms x y z

% Valores de las constantes
A = 0.2;
B = 0.01;
C = -0.4;

% Definir las ecuaciones del sistema
eq1 = A*x + y*z == 0;
eq2 = B*x + C*y - x*z == 0;
eq3 = -z - x*y == 0;
equations = [A*x + y*z, B*x + C*y - x*z, -z - x*y];
puntos_critic = solve([eq1, eq2, eq3], [x, y, z]);


for i = 1:5
    crit_x = puntos_critic.x(i)
    crit_y = puntos_critic.y(i)
    crit_z = puntos_critic.z(i)
    % Definir las variables simbólicas en la función de solución
    solution = [crit_x, crit_y, crit_z];
    % Llamar a la función EDOS_NO_LIN
    EDOS_NO_LIN(equations, [x, y, z], solution);
end
```

En primer lugar se definen las constantes A, B y C del problema. Posteriormente se calculan los puntos críticos o soluciones estacionarias igualando x', y' y z' a 0, ya que dichos puntos críticos son aquellos puntos del plano de fases que verifican el sistema cuándo las derivadas son nulas. La solución del sistema es dicho punto para cualquier valor de t.

Una vez calculados, se llama a la función EDOS_NO_LIN() que devuelve la clasificación de dichos puntos críticos.

A continuación se explica el funcionamiento de EDOS_NO_LIN():

Código de la función:

```
function [] = EDOS_NO_LIN(equations, variables, solution)
    % Definir las variables simbólicas
    syms x y z t ;

    % Obtener el número de ecuaciones y variables
    n = length(equations);
    m = length(variables);

    % Inicializar la matriz Jacobiana
    J = sym(zeros(n, m));

    % Calcular las derivadas parciales para formar la matriz Jacobiana
    for i = 1:n
        for j = 1:m
            J(i, j) = diff(equations(i), variables(j));
        end
    end

    % Sustituir las variables por las soluciones en la matriz Jacobiana
    substituted_Jacobian_matrix = subs(J, [x, y, z], solution);
    display(substituted_Jacobian_matrix);
    % Calcular los valores propios de la matriz Jacobiana
eigenvalues = eig(substituted_Jacobian_matrix);
display(eigenvalues)

% Convertir los valores propios a números
numeric_eigenvalues = double(eigenvalues);

% Analizar la estabilidad según los valores propios
if isreal(numeric_eigenvalues(1))
    if numeric_eigenvalues(1) < 0 && numeric_eigenvalues(2) > 0
        disp('Punto de silla -> Inestable');
    elseif numeric_eigenvalues(1) < 0 && numeric_eigenvalues(2) < 0
        disp('Sumidero -> Asintóticamente Estable');
    elseif numeric_eigenvalues(1) > 0 && numeric_eigenvalues(2) > 0
        disp('Repulsor -> Inestable');
    end
else
    if real(numeric_eigenvalues(1)) < 0
        disp('Foco Atractor (Espiral) -> Asintóticamente Estable')
    elseif real(numeric_eigenvalues(1)) > 0
        disp('Foco Repulsor (Espiral) -> Inestable')
    elseif real(numeric_eigenvalues(1)) == 0
        disp('¡¡¡CRITERIO NO DECIDE!!! -> Usar Lyapunov ')
    end
end
```

Esta función obtiene la matriz Jacobiana a partir del sistema (ya que el sistema no es lineal)y sustituye los valores de los puntos críticos y obtiene sus autovalores para cada punto crítico.

Luego, en función del signo de dichos autovalores, si son reales, o del signo de su parte real si son complejos se decide mediante una serie de condicionales el tipo de punto crítico.

Se debe tener en cuenta que este criterio no tiene la capacidad de decidir para autovalores complejos con parte real nula, en este caso habrá que usar otros métodos como Lyapunov o ver si el sistema es conservativo. Se tiene tambien una función dotada de algoritmos para calcular la estabilidad de dichos puntos dado la *edo*, los puntos críticos y el funcional de lyapunov a aplicar.

Vista gráfica de las soluciones

Introducción: Los diagramas representan la solución del sistema de ecuaciones diferenciales a medida que evoluciona en el tiempo. Cada punto en la trayectoria representa el estado del sistema en un momento específico.

Programa de matlab para graficar

```
% Condiciones Inciciales
c_iniciales = [1.3;0.01; -0.4];

for i = 1.3:0.01:1.33
    for j = 0:0.01:0.03
        for k = -0.4:0.01:-0.35
            % Definimos el sistema de ecuaciones diferenciales
            edo = @(t, X) [A*X(1) + X(2)*X(3); B*X(1) + C*X(2) - X(1)*X(3); -X(1)*X(2) - X(3)];
            % Establecemos las condiciones iniciales
            c_iniciales = [i; j; k];
            % Definimos el tiempo de ejecucion
            rang_tiempo = [0, 10000];
            % ODE45
            [t, X] = ode45(edo, rang_tiempo, c_iniciales)
            %Graficamos la solucion de ODE45
            figure;
            hold on
            plot3(X(:,1), X(:,2), X(:,3))
            title(['condiciones iniciales ' num2str(c_iniciales(1)) ', ' num2str(c_iniciales(2)) ', ' num2str(c_iniciales(3))]);
            view([-63.28 29.91]);
            hold off
        end
    end
end
```

Breve explicacion del codigo:
El código muestra la implementación de graficar varios diagramas para diferentes condiciones iniciales. Para modificarlas, se ha implementado una serie de bucles anidados que nos permitirán analizar el comportamiento para una variación en las tres componentes de las *C.Iniciales*. Para elegir una manera idónea de representación donde se muestre la peculiaridad de los sistemas dinámicos no lineales caóticos se ha elegido un video que barre las diferentes gráficas explicadas con anterioridad.

Se puede ver el video con todas las gráficas desde el siguiente link.
https://youtu.be/pKm_Zaz-8nc

Conclusiones

Tras analizar las soluciones estacionarias para varios valores de A,B,C y clasificarlas según correspondía y tras pintar las soluciones del problema para ciertas constantes seleccionadas, se llega a la conclusión de que cambiando A,B,C, las soluciones varían ligeramente pero más o menos su estructura sigue siendo la misma (la estructura típica de un sistema de 4 alas o *four-wing*). Tenemos que las soluciones estacionarias son puntos de silla para cualquier valor de las constantes.