MatLab Programmentwurf

Timo Johannsen, Benjamin Peiter, Omer Butt

27. April 2025

Inhaltsverzeichnis

1 Visualisierung der Systemdynamik durch Phasenportraits

Zuerst werden 3 verschiedene K-Werte mit $K_1 =]0, 0.6]$, $K_2 = [0.9, 1.1]$ und $K_3 = [1.4, 2.0]$ gewählt und die Länge und Anzahl der Trajektorien definiert.

```
K_values = [rand()*0.6, 0.9 + rand()*0.2, 1.4 + rand()*0.6];
    % Parameter fuer die drei K-Werte
N = 1000; % Laenge der Trajektorien
M = 50; % Anzahl der Trajektorien
```

Daraufhin werden für die 3 K-Werte die Phasenportraits erstellt. In einer Schleife werden durch die K-Werte iteriert und für jeden K-Wert die Trajektorien gezeichnet.

```
figure;
for idx = 1:3
    K = K_values(idx);
    subplot(1,3,idx);
    hold on;
```

Jede Trajektorie bekommt eine zufälligen Startpostition (I_0, θ_0) aus dem Bereich $[0, 2\pi]$ x $[0, 2\pi]$. Zusätzlich wird für die Trajektorie zwei Vektoren der Länge 1000 erstellt, in der die Werte für I und θ gespeichert werden.

```
    for m = 1:M
        I = rand()*2*pi;
        theta = rand()*2*pi;
        I_traj = zeros(1,N);
        theta_traj = zeros(1,N);
```

Die Trajektorie wird dann rekursiv berechnet. Dabei werden die Formeln:

$$I_{n+1} = (I_n + K \cdot \sin(\theta_n)) \bmod 2\pi$$
$$\theta_{n+1} = (\theta_n + I_n) \bmod 2\pi$$

verwendet, wobei I und θ in jedem Schritt aktualisiert werden. Die Trajektorie wird dann in den Vektoren gespeichert und zum Schluss wird jeder Punkt gezeichnet.

```
        for n = 1:N
            I = mod(I + K*sin(theta), 2*pi);
            theta = mod(theta + I, 2*pi);
            I_traj(n) = I;
            theta_traj(n) = theta;
        end
        plot(theta_traj, I_traj, '.', 'MarkerSize', 1);
    end
```

Diese Werte werden dann in einem Diagramm gezeichnet. Dabei wird die x-Achse mit θ und die y-Achse mit I beschriftet.

Dabei charakterisieren die Phasenportraits die Dynamik des Systems, dass für ein wachsendes K das chaotische Verhalten zunimmt.

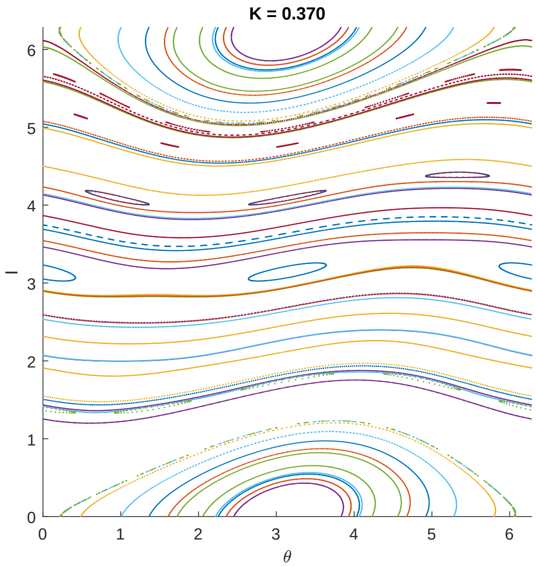

Abbildung 1: Phasenportrait für K_1

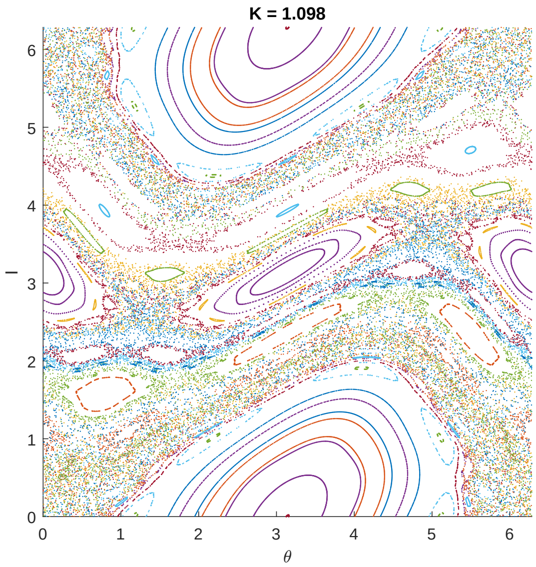

Abbildung 2: Phasenportrait für K_2

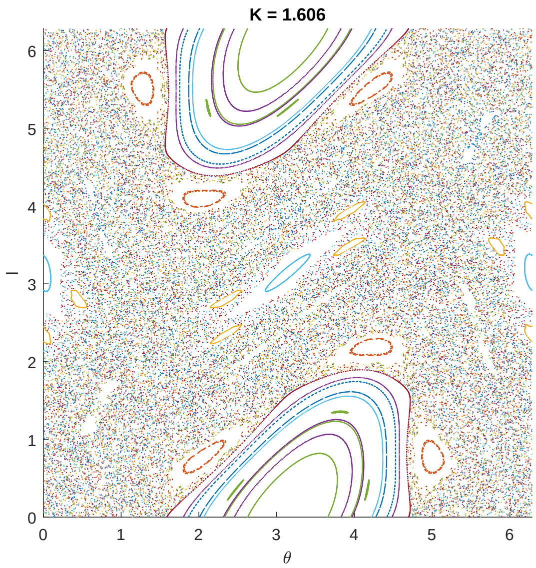

Abbildung 3: Phasenportrait für K_3

2 Quantifizierung des Chaos durch Ljapunov-Exponenten

```
K_vals = linspace(0, 4, 200);
N = 1000;
lambda1 = zeros(size(K_vals));
lambda2 = zeros(size(K_vals));

for idx = 1:length(K_vals)
    K = K_vals(idx);
    I = rand()*2*pi;
    theta = rand()*2*pi;
    Q = eye(2);
    sum_log_diag = zeros(1,2);

    for n = 1:N
        I = mod(I + K*sin(theta), 2*pi);
        theta = mod(theta + I, 2*pi);
        % Ableitungsmatrix DF
        DF = [1, K*cos(theta); 1, 1 + K*cos(theta)];
        A = DF * Q;
        [Q, R] = qr(A);
        sum_log_diag = sum_log_diag + log(abs(diag(R))');
    end

    lambda1(idx) = sum_log_diag(1)/N;
    lambda2(idx) = sum_log_diag(2)/N;
end
```