

Standardabbildung

Die Standardabbildung

$$\begin{aligned}\theta_{n+1} &= \theta_n + p_n \\ p_{n+1} &= p_n + K \sin \theta_{n+1}\end{aligned}\tag{1}$$

beschreibt die Dynamik des gekickten Rotors jeweils direkt nach dem $(n + 1)$ -ten Kick. Sie zeigt das grundlegende dynamische Verhalten von Hamilton'schen Systemen: Übergang von regulärer ($K = 0$) zu überwiegend chaotischer ($K \gg 1$) Dynamik sowie die typische Koexistenz regulärer und chaotischer Phasenraumbereiche.

Aufgabe 1.1:

(16 Punkte)

Betrachten Sie die Standardabbildung (1) für periodische Randbedingungen $\theta \in [0, 2\pi[$ und $p \in [-\pi, \pi[$. Schreiben Sie ein Programm, das für einen mittels der linken Maustaste vorgegebenen Startpunkt (θ_0, p_0) die Punkte $\{(\theta_i, p_i)\}_{0 \leq i \leq n}$ für $n = 1000$ Iterationen grafisch darstellt.

Wählen Sie die Symbole so, dass sich automatisch unterschiedliche Farben für verschiedene Orbits ergeben.

Betrachten Sie den Phasenraum für verschiedene Werte des Parameters K : $K = 0$; 0,2; 0,9; 2,1; 2,5; 6,0; 6,5. Schauen Sie sich mittels der Zoomfunktion `matplotlibs` den Grenzbereich regulärer und chaotischer Dynamik auf immer feineren Skalen an. Beschreiben Sie Ihre Beobachtungen in einem Kommentar am Ende des Programms.

Geben Sie das Programm für $K = 2,6$ ab.

Vorgaben und Hinweise:

- 1 Verwenden Sie eine Funktion die n Iterationen der Standardabbildung auf dem Torus berechnet und die Koordinaten als Arrays zurückgibt.

- ❶ Nutzen Sie die Modulo-Operation `%` zur Realisierung der periodischen Randbedingungen. Probieren Sie folgende Befehle aus:

```
1 print(1.5 % 2)
2 print(10.5 % 2)
3 print(-0.5 % 2)
4 import numpy as np
5 x = np.linspace(-3*np.pi, 3*np.pi, 5)
6 print(x % (2*np.pi))
```

Somit erhält man mittels

$$(p + np.pi) \% (2*np.pi) - np.pi$$

die Modulo-Operation für $p \in [-\pi, \pi[$.

- ❷ Für die Geschwindigkeit der grafischen Darstellung ist es ungünstig, wenn es zu viele (≥ 1000) Plotbefehle `plot()` gibt. Daher sollten Sie alle 1000 Punkte berechnen, in einem Array speichern und mit einem einzigen Plotbefehl diese Punkte hinzuzeichnen.
- ❸ Einmal dargestellte Orbits sollen sichtbar bleiben.
- ❹ Um sicherzustellen, dass nur bei Klick mit linker Maustaste innerhalb des Koordinatensystems und bei deaktivierter Zoomfunktion Orbits gestartet werden, nutzen Sie folgende Abfrage:

```
# Test, ob Klick mit linker Maustaste und im Plotbereich ax
# erfolgt sowie ob Zoomfunktion des Plotfensters deaktiviert ist:
mode = event.canvas.toolbar.mode
if event.button == 1 and event.inaxes == ax and mode == '':
    # starte neuen Orbit
```

Details zur Abgabe:

- Siehe Einführung I.
- Checkliste zur Abgabe: [checkliste.pdf](#) auf der Webseite.
- TAB-checker und Zeilenlängenüberprüfung: [filecheck.py](#) auf der Webseite.