Harmonischer Oszillator

November 9, 2022

0.1 Der harmonische Oszillator

Notebook erstellt am 08.11.2022 von C. Rockstuhl, überarbeitet von Y. Augenstein

Dieses Skript ist leider nicht in jupyterlite ausführbar, da einige benötigte Pakete dort nicht unterstützt werden.

Der harmonische Oszillator wird mit einer Differentialgleichung zweiter Ordnung mathematisch ausgedrückt, die konkrete Systeme im Kontext vieler verschiedener Untersdisziplinen der Physik ausreichend exakt beschreibt. Der harmonische Oszillator als Gleichung zur Beschreibung der Dynamik in einem System taucht immer dann auf, wenn wir die Bewgung in näherer Umgebung eines Gleichgewichtspunktes beschreiben.

Der harmonische Oszillator geht von einer Kraftwirkung aus, die linear proportional zur Auslenkung eines Teilchens, also zur Abweichung des Teilchens aus dessen Gleichgewichtslage, führt. Die Kraft ist negativ zur Auslenkung. Die Kraft wirkt daher der Auslenkung entgegen und möchte, dass das Teilchen wieder in seine Gleichgewichtslage zurückkehrt.

Konkrete physikalische Systeme, die durch diese Differentialgleichung beschrieben werden, sind:

- Eine Masse m, die an einer Feder aufgehängt und charakterisiert durch die Federkonstante k ist.
- Das mathematische Pendel, bei dem eine Masse an einem Pendel der Länge l hängt, welches unter dem Einfluss einer Gravitationskraft leicht ausgelenkt wird aus seiner Ruhelage.
- Bei der Berechnung der Schwingungsmoden eines zweiatomigen Moleküls.
- Bei der Beschreibung der Ladung auf einem Kondensator in einem Schwingkreis, in dem für einen kurzen Moment ein Strom floss.

Die Dynamik aller diese Systeme, und noch viele mehr, lassen sich durch die gleiche Differentialgleichung beschreiben. Wir wollen im Folgenden die Dynamik eines solchen harmonischen Oszillators mit ansteigender Komplexität beschreiben. Wir beginnen mit einem gedämpften freien harmonischen Oszillator und diskutieren anschließend einen getriebenen harmonischen Oszillator, der angeregt wird mit einer zeitharmonischen Kraft. Wir betrachten hier durchgehend eine Dämpfung, die proportional ist zur Geschwindigkeit.

0.1.1 Setup

Wir importieren hier die Module welche wir im Folgenden benötigen werden und definieren eine Hilfsfunktion, mit welcher wir die Differentialgleichungen lösen und interaktiv anzeigen lassen. Der

Inhalt dieser Zelle ist nicht nötig zum Verständnis der besprochenen Probleme, Sie können ihn also ignorieren.

[1]:
```
%matplotlib ipympl
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import solve_ivp  # Solve Initial Value Problen, siehe␣
↪Notebook "DGL und Wurf"
from ipywidgets import interactive  # Interaktive Plots mit Schiebereglern


def solve_and_plot(ode_problem, x0, t, driven=False):
    """Diese Funktion erstellt zunächst ein leeres Schaubild.
    Sie gibt dann die Funktion solve_and_draw zurück, mit welcher dann
    die DGL gelöst wird und lediglich neue Linien in das bestehende
    Schaubild hineingezeichnet werden.

    Dies hat den Vorteil, dass das Schaubild nicht nach jeder Önderung
    der Parameter komplett neu erstellt werden muss, was dazu führen würde,
    dass das interaktive Schaubild, welches wir zeichnen wollen, flackern␣
↪würde.
    """

    plt.close()
    fig, ax = plt.subplots(1, 1)
    (line0,) = ax.plot(t, np.zeros_like(t), c="tab:blue", label="Auslenkung")
    if ode_problem.driven:
        (line1,) = ax.plot(t, np.zeros_like(t), c="tab:orange",␣
↪label="Anregung")

    ax.grid()
    ax.set_xlabel("Zeit")
    ax.set_ylabel("Auslenkung")
    fig.legend(framealpha=1)
    fig.tight_layout()

    def solve_and_draw(**kwargs):
        # Zunächst lösen wir die DGL
        sol = solve_ivp(ode_problem, (t[0], t[-1]), x0, t_eval=t,␣
↪args=kwargs.values())

        # Und zeichnen dann die relevaten Linien
        line0.set_ydata(sol.y[0])
        if ode_problem.driven:
            omegad = kwargs["omegad_omega0"] * kwargs["omega0"]
            line1.set_ydata(1 / 4 * kwargs["F_m"] * np.sin(omegad * t))
```

```
        # Und als Letztes skalieren wir die Achsen neu, da diese an
        # die neuen Werte von y angepasst werden müssen
        ax.relim()
        ax.autoscale_view()

    return solve_and_draw
```

0.1.2 Der freie harmonische Oszillator

Der freie harmonische Oszillator wird durch die folgende Differentialgleichung beschrieben:

$$\ddot{x} + 2\gamma\dot{x} + \omega_0^2 x = 0 \quad .$$

Hier ist ω_0 die Eigen- oder Resonanzfrequenz. Diese ergibt sich aus der ganz konkreten physikalischen Situation, die wir betrachten. Der Dämpfungskoeffizient γ in dem Dämpfungsterm ist definiert als $\gamma = \frac{r}{2m}$. Der Faktor 2 im Dämpfungsterm ist etwas willkürlich gewählt hier, aber in Übereinstimmung mit der Notatation in der Vorlesung. Analytische Ausdrücke vereinfachen sich durch diese 2.

Zur eindeutigen Lösung der Bewegung des harmonischen Oszillators benötigen wir zwei Anfangsbedingungen. Hierfür betrachtet man dann den Ort bzw. die Geschwindigkeit zu einer Anfangszeit $t_{\mathrm{Anfang}} = t_0$. Aus Gründen der Praktikabilität und da unser System sowieso zeitinvariant ist, wählen wir üblicherweise $t_0 = 0$.

In Analogie zur Lösung der Bewegungsgleichung für den Wurf, formulieren wir diese eine Differentialgleichung zweiter Ordnung zunächst in zwei gekoppelte Differentialgleichungen erster Ordnung um. Diese integrieren wir dann mit einer in einer passenden Python-Bibliothek implementierten Funktion, so dass wir die Position und die Geschwindigkeit des Teilchens nach einer endlichen Zeit berechnen können.

Für die beiden Variablen, die durch Differentialgleichugnen erster Ordnung beschrieben werden, wählen wir den Ort x und die Geschwindigkeit $\dot{x}$ und fassen diese in einem Vektor $\mathbf{X}(t)$ zusammen. Dieser ist definiert als:

$$\mathbf{X}(t) = \begin{pmatrix} X_1(t) \\ X_2(t) \end{pmatrix} = \begin{pmatrix} x(t) \\ \dot{x}(t) \end{pmatrix} \quad .$$

Die von uns zu lösende Bewegungsgleichung lautet dann:

$$\dot{\mathbf{X}}(t) = \begin{pmatrix} \dot{x} \\ -2\gamma\dot{x} - \omega_0^2 x \end{pmatrix} \quad .$$

In dem folgenden Pythonskript variieren wir die Dämpfung und die Eigenfrequenz, um Details der Bewegung zu studieren.

Beachten Sie, dass Sie zwei verschiedene Arten (oder beliebige Kombinationen) von Anfangsbedingungen berücksichtigen können. So können Sie dem harmonischen Oszillator eine Auslenkung geben und würden beobachten, wie er langsam wieder seiner Gleichgewichtslage entgegen strebt. Sie können aber auch die Auslenkung zum Zeitpunkt $t = t_0 = 0$ auf 0 setzen und dem hamonischen Oszillator eine endliche Geschwindigkeit geben. Das entspricht der Situation einer stoßförmigen Anregung.

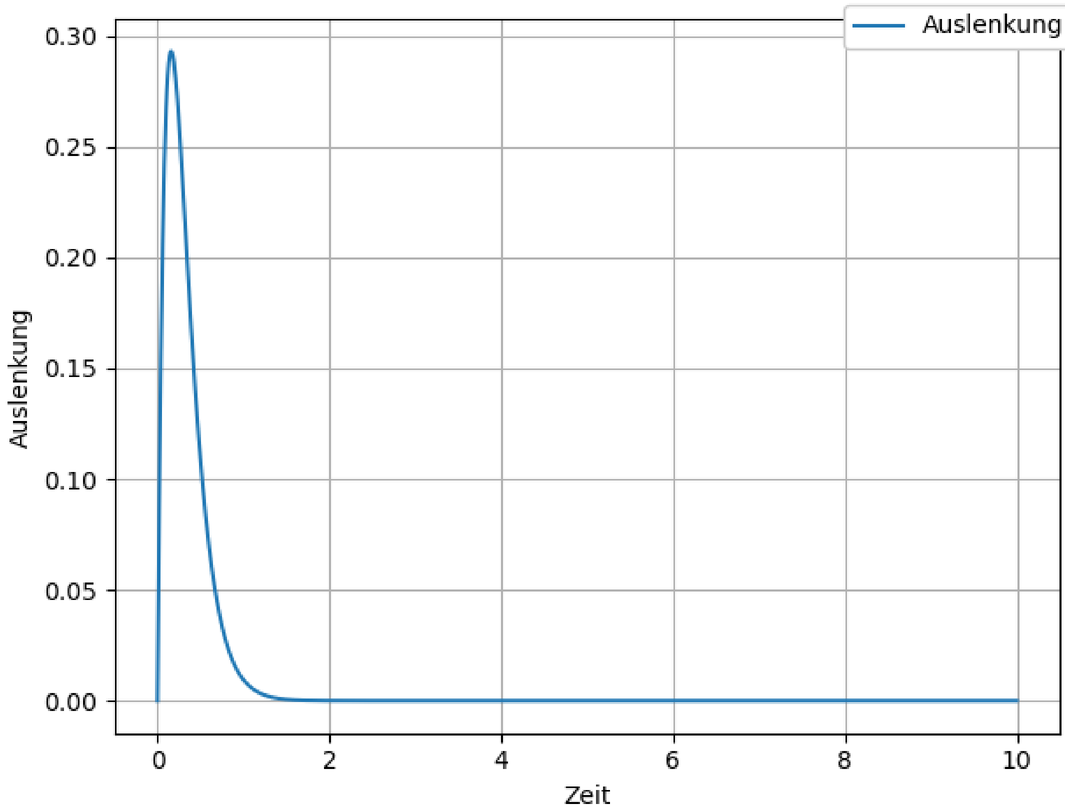
[2]:
```python
def damped_harmonic_oscillator(t, X, gamma, omega0):
    """Definition der DGL
    Als Input bekommen wir die Variablen und geben dann ihre Zeitableitung␣
 ↪zurück.
    """
    x, dx = X
    ddx = -2 * gamma * dx - omega0**2 * x
    return [dx, ddx]


# Hier speichern wir als Funktionsattribut, ob es sich um einen getriebenen␣
 ↪Oszillator handelt,
# denn beim getriebenen Oszillator plotten wir noch die Anregungsfrequenz.
damped_harmonic_oscillator.driven = False


# Anfangsbedingungen: [Auslenkung, Geschwindigkeit]
x0 = [0, 5]  # Hier beginnen wir mit einer endlichen Geschwindigkeit
# x0 = [1, 0]  # Hier beginnen wir mit einer endlichen Auslenkung

# Das Zeitintervall, welches wir hier betrachten
t = np.linspace(0, 10, 1000)

# Die Funktion `interactive` nimmt eine Funktion (welche interaktiv␣
 ↪dargestellt werden soll)
# und erstellt Schieberegler für die veränderlichen Variablen. In unserem␣
 ↪Fall hier sind das
# der Dämpfungskoeffizient gamma sowie die Resonanzfrequenz omega0.
# Die Regler werden dann erstellt wie (min, max, step).
# Jedes Mal, wenn ein Schieberegler verändert wird, wird die Funktion neu␣
 ↪aufgerufen
# für den entsprechenden Wert.
interactive(
    solve_and_plot(damped_harmonic_oscillator, x0, t),
    gamma=(0, 4 * np.pi, 0.01),  # Dämpfungskoeffizient
    omega0=(0, 4 * np.pi, 0.01),  # Resonanzfrequenz
)
```



Spielen Sie mit dem oben stehenden Code etwas herum und versuchen Sie bitte, die in der Vorlesung gezeigten Beispiele zu verifizieren! Beobachten Sie das dynamische Verhalten im Kriechfall ($\omega_0 = \gamma$) und untersuchen Sie, wie sich das System mit sehr großer Dämpfung und sehr niedriger Dämpfung verhält. Beobachten Sie bitte auch, wie sich die Oszillationsperiode ändert, wenn Sie die Eigen- oder Resonanzfrequenz ändern.

0.1.3 Der getriebene harmonische Oszillator mit einer periodischen Anregung

Der getriebene harmonische Oszillator mit einer zeitharmonischen Anregung wird durch die folgende Differentialgleichung beschrieben:

$$\ddot{x} + 2\gamma\dot{x} + \omega_0^2 x - F_m \sin(\omega_d t) = 0 \quad .$$

Die Gleichung ist identisch zum freien Oszillator, nur dass jetzt noch eine periodische oszillierende Kraft auf den Oszillator wirkt, die charakterisiert ist durch die Amplitude F_m und die Oszillationsfrequenz ω_d.

Wir fassen hier wieder den Ort x und die Geschwindigkeit $\dot{x}$ in einem Vektor $\mathbf{X}(t)$ zusammen. Die von uns zu lösende Bewegungsgleichung lautet dann:

$$\dot{\mathbf{X}}(t) = \begin{pmatrix} \dot{x} \\ -2\gamma\dot{x} - \omega_0^2 x + F_m \sin(\omega_d t) \end{pmatrix} \quad .$$

In dem folgenden Pythonskript variieren wir die Dämpfung, die Eigenfrequenz, die Anregungsfrequenz der Kraft und die Amplitude der Kraft, um Details der Bewegung zu studieren.

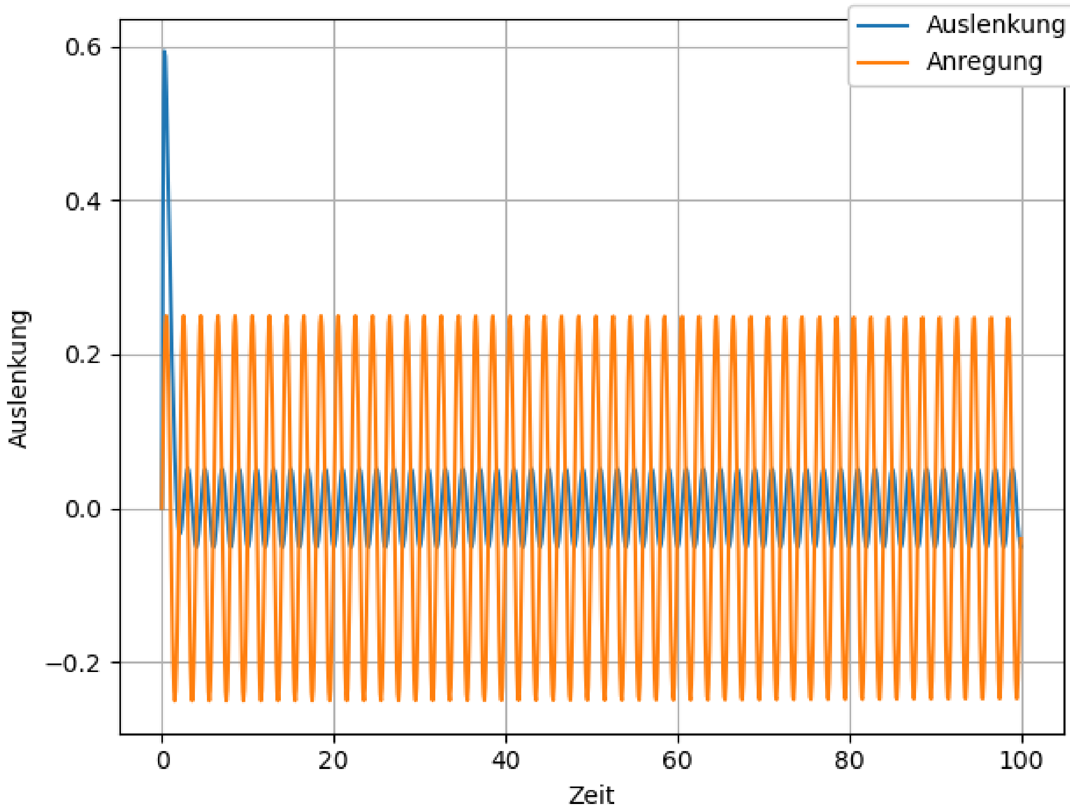
[3]:
```
def driven_harmonic_oscillator(t, X, gamma, omega0, omegad_omega0, F_m):
    x, dx = X
    omegad = omegad_omega0 * omega0  # omegad als Bruchteil von omega0
    ddx = -2 * gamma * dx - omega0**2 * x + F_m * np.sin(omegad * t)
    return [dx, ddx]


driven_harmonic_oscillator.driven = True

# Anfangsbedingungen: [Auslenkung, Geschwindigkeit]
x0 = [0, 5]  # Hier beginnen wir mit einer endlichen Geschwindigkeit
# x0 = [1, 0]  # Hier beginnen wir mit einer endlichen Auslenkung

t = np.linspace(0, 100, 1000)

interactive(
    solve_and_plot(driven_harmonic_oscillator, x0, t),
    gamma=(0, 2 * np.pi, 0.01),  # Dämpfungskoeffizient
    omega0=(0, 2 * np.pi, 0.01),  # Resonanzfrequenz
    omegad_omega0=(0, 2, 0.001),  # Anregungsfrequenz als Bruchteil der␣
 ↪Resonanzfrequenz
    F_m=(0.5, 1.5, 0.01),  # Amplitude der Anregung
)
```



Spielen Sie auch hier bitte wieder mit den Werten der Parameter etwas rum. Beobachten Sie das Einschwingverhalten des Systems, dass es also erst nach einer endlichen Zeit eine zeitharmonische Oszillation ausführt. Können Sie sich erklären, warum dem so ist? Beobachten Sie, dass auch bei einer zeitharmonischen Anregung und einer endlichen Dämpfung die Amplitude der Oszillation nicht bis ins Unermessliche steigt, sondern endlich bleibt. Sie wird sehr groß, je geringer die Dämpfung ist, bleibt aber immer endlich.

Beobachten Sie weiterhin, wie die Amplitude sich verändert, wenn Sie die Anregungsfrequenz relativ zur Eigenfrequenz des harmonischen Oszillators ändern. Achten Sie auch auf die Phase relativ zur Anregung. Unterhalb bzw. oberhalb der Resonanzfrequenz befindet sich der harmonische Oszillator einmal in- und einmal außer-Phase. Alle diese Details werden Ihnen durch die Gleichungen in der Vorlesung beschrieben, aber hier würden Sie noch einmal sehen kann, welche Auswirkung dies auf die Bewegung als solches hat.