Computerpyhsik

Übungsblatt 4

Konrad Beck und Max Reicherd

Inhaltsverzeichnis

1 Einleitung **2**

2 Nummerische Lösung der DGL mit dem Runge-Kutta-Verfahren **2**

3 Untersuchung der Amplitude, Phasenverschiebung und Resonanzfrequenz **4**

4 Zusammenfassung **6**

1 Einleitung

In dieser Übung werden Eigenschaften einer angeregten elastischen Feder Oszillation mit Dämpfung untersucht. Aus der nicht linearen Rückstellkraft, dem Dämpfungsterm, der treibenden Kraft kann mittels des Kräftegleichgewichts eine gewöhnlichen Differentialgleichung (DGL) zweiter Ordnung aufgestellt werden.

$$\ddot{x} + \delta\dot{x} + \alpha x + \beta x^3 = \gamma \cos \omega t \text{ mit } \textit{Auslenkung } \mathrm{x(t)}$$

Durch das nummerische lösen der DGL mit dem Runge-Kutta-Verfahren wird die Pendelbewegung graphisch dargestellt. Mittels der nummerisch bestimmten Lösung wird die Amplitude und Phasenverschiebung Φ für $\beta \in [0, 0.05]$ als Funktion von $\omega \in [\omega_{min}, \omega_{max}]$ mit $\omega_{min} = \frac{1}{2}$ und $\omega_{max} = 2$ bestimmt. Entlang der Aufgabenstellung folgt $\alpha = \gamma = 1$ und $\delta = 0.1$.
Weiterhin wird aus den Maxima von A per β die Resonanzfrequenz $\omega_r(\beta)$ bestimmt.

2 Nummerische Lösung der DGL mit dem Runge-Kutta-Verfahren

Die vorgegebene DGL 2 Ordnung wird durch analytische Umformung in ein System von zwei DGL's erster Ordnung umgeformt:

$$\begin{aligned} \gamma \cos \omega t &= \ddot{x} + \delta\dot{x} + \alpha x + \beta x^3 \\ \Leftrightarrow \ddot{x} &= -\delta\dot{x} - \alpha x - \beta x^3 + \gamma \cos \omega t \end{aligned}$$

Mit der Substitution: $y_1 = x(t)$ und $y_2 = \dot{x}(t)$ lässt sich ein DGL System erster Ordnung aufstellen:

$$\vec{y} = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \text{ und } \vec{y} = \begin{pmatrix} \dot{y_1} \\ \dot{y_2} \end{pmatrix} = \begin{pmatrix} \dot{x}(t) \\ \ddot{x}(t) \end{pmatrix} = \begin{pmatrix} y_2 \\ -\delta y_2 - \alpha y_1 - \beta y_1^3 + \gamma \cos \omega t \end{pmatrix}$$

Somit wird eine vektorwertige DGL betrachtet:

$$\dot{\vec{y}} = \vec{f}\,(t, \vec{y})$$

Anschließend wird das DGL System mittels dem klassischen Runge-Kutta-Verfahren nummerisch gelöst. Hier wurde ein vierstufiges Runge-Kutta-Verfahren implementiert. Dieses hat die Fehlerordnung $O(h^5)$. Dabei ist dieses im allgemeinen entlang des CP-Skriptes wie folgt definiert:

$$\begin{aligned} k_1 &= hf\,(t_i, x_i) \\ k_2 &= hf\left(t_i + \frac{h}{2}, x_i + \frac{k_1}{2}\right) \\ k_3 &= hf\left(t_i + \frac{h}{2}, x_i + \frac{k_2}{2}\right) \\ k_4 &= hf\,(t_i + h, x_i + k_3) \end{aligned}$$

mit der Iterationsvorschrift:

$$x_{i+1} = x_i + \left(\frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)\right)$$

Dieses Verfahren wurde wie folgt in Python umgesetzt

```python
def rungeKutta(f, f2, t0, x0, v0, h, n):
    result = [[t0, x0, v0]]

    for i in range(n):
        a1 = f(result[i][0], result[i][1], result[i][2])
        a2 = 2 * f(result[i][0] + h / 2, result[i][1] + h / 2 * result[i][2],
        result[i][2] + a1 * h / 2)
        a3 = 2 * f(result[i][0] + h / 2, result[i][1] + h / 2 * result[i][2],
        result[i][2] + a2 * h / 2)
        a4 = f(result[i][0] + h, result[i][1] + h * result[i][2],
        result[i][2] + a3 * h)
        a = (a1 + a2 + a3 + a4) / 6 #Berechnung der Beschleunigung

        v1 = f2(result[i][0], result[i][1], a1)
        v2 = 2 * f2(result[i][0] + h / 2 * v1, result[i][1] + h / 2 * result[i][2],
        a2/2)
        v3 = 2 * f2(result[i][0] + h / 2 * v2, result[i][1] + h / 2 * result[i][2],
        a3/2)
        v4 = f2(result[i][0] + h, result[i][1] + h * v3,
        a4)
        v = (v1 + v2 + v3 + v4) / 6#Berechnung der Geschwindigkeit
        result.append([result[i][0] + h, result[i][1] + h * v, result[i][2] + h * a])
        #Hinzuf gen der i+1 Werte

    return np.array(result)
```

Der Funktion wird die Funktion zum berechnen von a: f und die zum berechnen von v: $f2$ Übergeben. Außerdem werden die Startwerte, die Schrittweite und Schrittzahl übergeben. Der Rückgabewert ist ein Array mit den Zeiten, Geschwindigkeiten und Orten. Mit der Annäherung ergibt sich die Graphik:

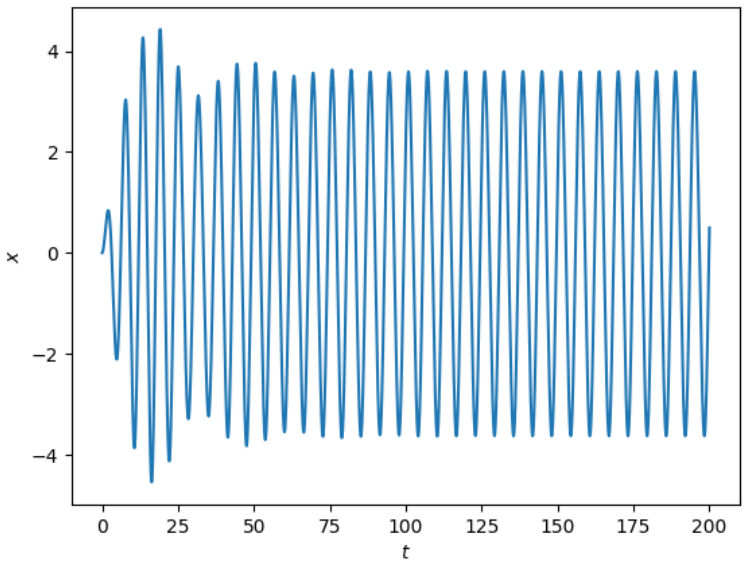

Abbildung 1: Graphische Darstellung der angenäherten Bewegungsgleichung ($\beta = 0.03$)

Diskussion

Aus der Graphik lässt sich interpretieren, dass die Amplitude für große n gegen einen konstanten Wert konvergiert. Weiterhin sind leicht chaotische Schwankungen im bis 200 Zeiteinheiten zu sehen. Diese kommen durch das Einpendeln und dem nicht linearen Dämpfungsterm zustande. Da im Vergleich zu einem linearen Dämpfungsterm (hooksches Gesetz) die Amplitude monoton fällt, entspricht dieses Verhalten den Erwartungen.

3 Untersuchung der Amplitude, Phasenverschiebung und Resonanzfrequenz

In der Graphik 1 ist zu sehen, dass die Amplitude für große n gegen einen festen Wert konvergiert. Unter dieser Annahme wird mit der Funktion **getAmplitude** der hinterste Abschnitt der Funktion betrachtet und von diesem das Maximum bestimmt.

Mit der Funktion **getAmplitude** wird ebenfalls die Phasenverschiebung bestimmt, wie viele Array(Zeit)punkte eine Schwingung ausmachen. Innerhalb dieser Intervalle kann die Position der maximalen Auslenkung zwischen der erregten Schwingung und dem elastischen Pendel ermittelt werden. Anschließend wurde der Betrag, des maximalen Werts bestimmt, dessen Ort und die Differenz (Phasenverschiebung) beider Werte bestimmt.

```
def getAmplitude(f, f2, t0, x0, v0, h, n,frequenz): #Berechnet Die Amplitude und die Phasenverschiebung am Ende des Einschwingvorgangs.
    g = rungeKutta(f, f2, t0, x0, v0, h, n)
    T = 2 * np.pi / frequenz # Periodendauer
    schritte = int(T / h) #Periodendauer in vielfachen von h
    g = g[n-schritte:n] # Aus dem Array wird genau eine Periode geschnitten
    g = g.transpose()
    c = np.cos(frequenz*g[0])# Erregerschwingung
    maxAmplitude = np.max(g[1])
    phi1 = np.where(c == np.max(c))
    phi2 = np.where(g[1] == maxAmplitude)
    phasenverschiebung= np.abs(g[0][phi1]-g[0][phi2])# Die Phasenverschiebung ergibt sich aus der Differenz der Zeiten bei welchen die Macxima erreicht sind
    return [maxAmplitude,phasenverschiebung]
```

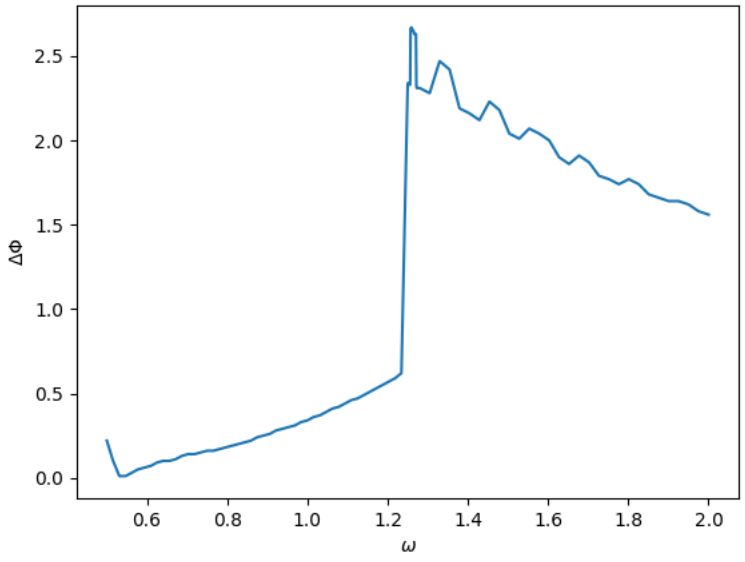

Abbildung 2: Beispielhafte Phasenverschiebung für $\beta = 0.03$

Weiterhin wurde die Resonanzkurve ermittelt. Dazu wurde die Amplitude gegen die Frequenz graphisch dargestellt. Durch die Bestimmung des Maximums kann die Resonanzfrequenz ermittelt werden:

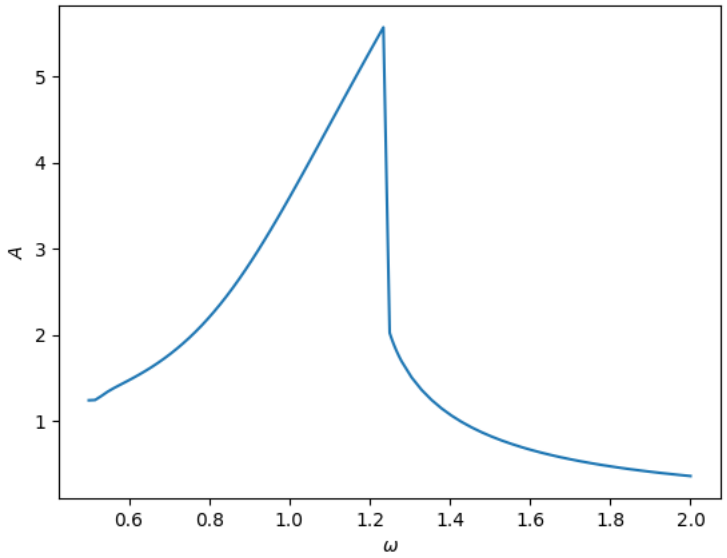

Abbildung 3: Beispielhafte Resonanzkurve für $\beta = 0.03$

Im letzten Schritt wurde die Resonanzfrequenz $\omega_r(\beta) = \omega_{max}$ für 10 gleichmäßig verteilte $\beta \in [0, 0.05]$ bestimmt:

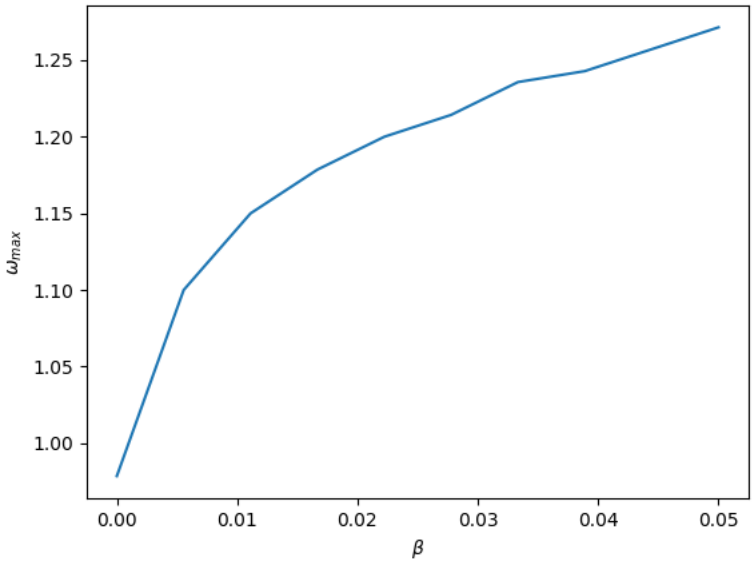

Abbildung 4: Resonanzfrequenzen für 10 gleichmäßig verteilte $\beta \in [0, 0.05]$

Der plot wurde mit folgendem code erzeugt

```
def plotResFreq(): # Aufgabenteil 2
    res = []

    x = np.linspace(0,0.05,10)
    for b in x:
        a=getResonanz(b,0.02,10000,True)
        m = np.max(a[1])
```

```
        i= np.where(a[1] == m)
        print(i)
        res.append(a[0][i[0]])
    plt.xlabel(r"$\beta$")
    plt.ylabel(r"$\omega_{max}$")
    plt.plot(x,res)
    plt.savefig("wMaxGegenBeta")
    plt.show()
```

Die Funktion berechnet für verschiedene β die Resonanzfrequenz

Diskussion
Amplitude: Entlang der physikalischen Argumentation, dass sich für lange Schwingzeiten eine konstante Amplitude einstellt entspricht die Graphik und die Amplitude den Erwartungen.
Phasenverschiebung: Die Abbildung der Phasenverschiebungen $\Delta\Phi$ enrspricht den Erwartungen. Jedoch ist im Abschnitt nach der Resonanzfrequenz der Graph nicht gerade/linear. Diese Anomalie kann durch Ungenauigkeiten der Schwingungsintervale zustande kommen.
Resonanzfrequenz: Die Graphik entspricht dem erwarteten Verlauf einer Resonanzkurve.

4 Zusammenfassung

In dieser Übung wurde eine Differentialgleichung zur Bewegung eine elastischen Feder gegeben. Diese Differentialgleichung zweiter Ordnung wurde mittles eines vierstufigen Runge-Kutta-Verfahren numerisch gelöst.Die Schwingung dieser Feder wurde graphisch dargestellt und anschließend aus dieser Darstellung Ansätze für die Bestimmung der Amplitude entwickelt.
Die Amplitude wurde unter der Annahme einer konvergierenden Amplitude für $n\,\mathrm{inf}$ als das Maxmimum des hintersten Funktionsabschnitter bestimmt. Die Phasenverschiebungen $\Delta\Phi$ wurde durch den Vergleich der Orte der maximalen Amplituden ermittelt. Das Ergebnis entsprach den Erwartungen. Es ergaben sich allerdings Anomalien die durch die Ungenauigkeiten der Periodenintervalle zustande gekommen sind. Weiterhin wurde die Resonanzfrequenz $\omega_r(\beta)$ für 10 gleichmäßig verteilte $\beta \in [0, 0.05]$ bestimmt.