Numerische Simulation von Pendelsystemen

Bericht zu den Übungsaufgaben

Autor: Seyam Mir Afghan

16. Januar 2026

Inhaltsverzeichnis

1 Fadenpendel

Gegeben ist die Bewegungsgleichung eines Fadenpendels:

$$\dot{\varphi} + \frac{g}{l} \sin \varphi = 0 \tag{1}$$

Für die folgenden Aufgaben gilt $\frac{g}{l} = 0,1\,\mathsf{s}^{-2}$.

1.1 Aufgabe 1: Analytische und numerische Näherungslösung

1. Bewegungsgleichung und Kleinwinkelnäherung

Die vollständige Gleichung lautet $\ddot{\varphi} + \frac{g}{l} \sin(\varphi) = 0$. Für kleine Auslenkungen gilt die Näherung $\sin(\varphi) \approx \varphi$ (Kleinwinkelnäherung). Die zu lösende homogene lineare Differentialgleichung 2. Ordnung lautet:

$$\ddot{\varphi} + \frac{g}{l}\varphi = 0 \tag{2}$$

2. Lösung der linearen Differentialgleichung

Lösen erfolgt klassisch per Exponentialansatz: $\varphi(t) = e^{\lambda t}$. Da $\lambda^2 + \frac{g}{l} = 0$, folgt die charakteristische Gleichung:

$$\lambda^2 = -\frac{g}{l}$$

Daraus ergeben sich die rein imaginären Eigenwerte:

$$\lambda_{1,2} = \pm i\sqrt{\frac{g}{l}} = \pm i\omega_0$$

Damit ergibt sich die allgemeine Lösung:

$$\varphi(t) = c_1 \cos(\omega_0 t) + c_2 \sin(\omega_0 t)$$

3. Anfangsbedingungen

Das Pendel wird aus der Ruhe mit der Anfangsauslenkung φ_0 aus der Ruhe losgelassen:

$$\varphi(0) = \varphi_0, \quad \dot{\varphi}(0) = 0$$

Die analytische Lösung für das Auslenken und Loslassen lautet:

$$\varphi(t) = \varphi_0 \cos\left(\sqrt{\frac{g}{l}}t\right)$$

4. Periodendauer analytisch bestimmen

Vergleicht man dies mit der allgemeinen Schwingungsgleichung $\varphi(t) = \varphi_0 \cos(\omega t)$, so identifizieren wir die Eigenkreisfrequenz:

$$\omega = \sqrt{\frac{g}{l}} = \sqrt{0,1\,\mathsf{s}^{-2}} \approx 0,3162\,\mathsf{s}^{-1}$$

Daraus lässt sich nun die Periodendauer T (die Zeit für eine vollständige Hin- und Herschwingung) berechnen:

$$T = \frac{2\pi}{\omega} \approx 19,8692\,\mathsf{s}$$

Numerische Näherungslösung

Für die numerische Lösung in MATLAB wird die Differentialgleichung zweiter Ordnung in ein System zweier Differentialgleichungen erster Ordnung überführt. Dabei ergibt sich für das linearisierte Pendel:

$$\begin{pmatrix}\dot{\varphi}\\ \ddot{\varphi}\end{pmatrix} = \begin{pmatrix}\omega\\ -\frac{g}{l}\varphi\end{pmatrix}$$

Dieses System wird mit den Anfangsbedingungen $\varphi_0 = 1^\circ$ und $\dot{\varphi}_0 = 0$ numerisch integriert.

```
clear; clc; close all;

g = 9.81; % Erdbeschleunigung in m/s²
s = 0.1; % in s^(-2)
l = g / s; % Pendellänge ( 98,1m )
omega = sqrt(g / l); % Eigenkreisfrequenz
T_analytisch = 2 * pi / omega; % analytisch Periodendauer
phi_0 = deg2rad(1); % Startwinkel
tspan = [0 100*T_analytisch]; % 0 bis 100 Perioden
y0 = [phi_0; 0]; % Startzustand als Vektor (Start aus Ruhe)

[t, y] = ode45(@(t,y) pendulum_ode(t,y,g,l), tspan, y0);
% Ausgabe: t Vektor mit Zeitpunkten
%          y Matrix: y(:,1) = phi(t)
%                    y(:,2) = Ableitung von phi(t)
```

Listing 1: Definition und Lösung mit ode45

Periodendauer mittels Nulldurchgang Zusätzlich wird die Periodendauer über die Nulldurchgänge der Auslenkung bestimmt. Die Zeitpunkte der Nulldurchgänge werden durch Interpolation zwischen zwei aufeinanderfolgenden Stützstellen mit Vorzeichenwechsel bestimmt. Da aufeinanderfolgende Nulldurchgänge einer harmonischen Schwingung einer halben Periodendauer entsprechen, ergibt sich $T = 2 \cdot \Delta t_{\text{Null}}$.

```
phi = y(:,1);
k = find(phi(1:end-1) .* (phi(2:end)) < 0); % Bilde produkte benachbarter Werte und prüfe
    < 0
% find holt die indexliste
% Zeitpunkte der Nullstellen mittels Interpolation
z = t(k) - (phi(k) .* ((t(k+1) - t(k)) ./ (phi(k+1) - phi(k))));
% Aufeinanderfolgende Nulldurchgänge sind halbe Perioden
T_Nulldurchgang= 2*mean(diff(z));
```

Listing 2: Berechnung mittels Nulldurchgängen

Periodendauer mittels findpeaks Die Periodendauer wird über die zeitlichen Abstände aufeinanderfolgender Maxima der Auslenkung bestimmt. Der Mittelwert dieser Abstände ergibt die numerische Periodendauer.

```
% Höhe der Peaks, Position der Peaks
[pks, locs] = findpeaks(phi, t);
% Abstände zwischen den Peaks rechnen + mean
T_findpeaks = mean(diff(locs));

% Plot, Hilfsfunktion, Ergebnisse aufrufen
t_plot = t <= 5*T_analytisch;
figure;
plot(t(t_plot), phi(t_plot)); grid on;
xlabel("Zeit in s");
ylabel("Auslenkung \phi in rad");
phi_analytisch = phi_0*cos(omega*t(t_plot));
```

```
hold on
plot(t(t_plot), phi_analytisch, '--', 'LineWidth', 1.2)
legend('Numerisch','Analytisch')
title('Numerische Loesung der linearisierten Bewegungsgleichung', 'FontSize', 9);

fprintf('Analytisch ermittelte Periodendauer T: %.4f s\n', T_analytisch);
fprintf('Numerisch (Nulldurchgang) ermittelte Periodendauer T: %.4f s\n', T_Nulldurchgang
    );
fprintf('Numerisch (findpeaks) ermittelte Periodendauer T: %.4f s\n', T_findpeaks);

function dydt = pendulum_ode(t,y,g,l)
    phi = y(1); % y(:,1) = phi(t)
    phi_dydt = y(2); % y(:,2) = Ableitung von phi(t)
    dydt = [phi_dydt; -(g/l) * phi];
end
```

Listing 3: Berechnung mittels findpeaks und Plot

1.2 Aufgabe 2: Exakte Numerische Lösung

1. DGL und Referenzperiode

Zur Bestimmung der Periodendauer für beliebige Anfangsauslenkungen wird die nichtlineare Bewegungsgleichung des Pendels numerisch gelöst:

$$\ddot{\varphi} + \frac{g}{l} \sin \varphi = 0$$

Im Gegensatz zur Kleinwinkelnäherung bleibt der Sinus-Term erhalten, wodurch die Periodendauer von der Anfangsauslenkung φ_0 abhängt. Die Differentialgleichung 2. Ordnung wird als System 1. Ordnung geschrieben. Wie bei Aufgabe 1 verwendet man als Anfangsbedingungen (in Ruhe, losgelassenes Pendel):

$$\varphi(0) = \varphi_0, \quad \dot{\varphi}(0) = 0$$

```
clear; clc; close all;
g = 9.81;
s = 0.1;
l = g/s;
omega = sqrt(g / l);
T0 = 2*pi / omega;
targets = [0.01 0.05 0.10 0.50 1.00];
```

Listing 4: Initialisierung Exakte Lösung

2. Zielwinkel für feste Abweichungen

Zusätzlich werden die Anfangsauslenkungen ermittelt, bei denen die Periodendauer um p größer als T_0 ist. Dazu wird die Nullstelle der Funktion:

$$F(\varphi_0) = \frac{T_{\text{ex}}(\varphi_0)}{T_0} - (1 + p) = 0$$

im Winkelintervall $[1°, 170°]$ numerisch bestimmt.

```
phi0_found = NaN(size(targets));

% Schleife über alle geforderten Abweichungen, numel = Anzahl (5)
for j = 1:numel(targets)
    % Wert rauspicken
    p = targets(j);
```

```
    % Funktion definieren, deren Nullstelle null wird.
    F = @(phi0) T_exact(phi0,g,l,T0)/T0 - (1+p);
    % Intervall von a bis b (hier wird gesucht)
    a = deg2rad(1);
    b = deg2rad(170);
    % fzero sucht wo im Intervall F(phi) = 0 gilt
    phi0 = fzero(F, [a b]);
    phi0_found(j) = rad2deg(phi0);
    % Periodendauer bestimmen
    Tex = T_exact(phi0,g,l,T0);
    fprintf("%3.0f%%  = %9.4f Grad  T = %.8f s\n", 100*p, rad2deg(phi0), Tex);
end
```

Listing 5: Suche der Winkel mit fzero

3. Visualisierung und Funktionen

Die Ergebnisse werden grafisch dargestellt, wobei die Abweichung der Periodendauer gegenüber der harmonischen Näherung über der initialen Auslenkung aufgetragen wird.

```
%------------- Für den Plot---------------
phi0_deg = linspace(0.5, 170, 250);     % beliebige Auslenkungen (nicht bis 180)
phi0_rad = deg2rad(phi0_deg);

Tex_list = NaN(size(phi0_rad));
dev_list = NaN(size(phi0_rad));

for i = 1:numel(phi0_rad)
    Tex_list(i) = T_exact(phi0_rad(i), g, l, T0);
    dev_list(i) = (Tex_list(i)/T0 - 1) * 100;  % Abweichung in %
end

% --- Visualisierung 1: Abweichung vs. Auslenkung ---
figure('Name', 'Abweichung der Periodendauer');
plot(phi0_deg, dev_list, 'b-', 'LineWidth', 2); hold on;

% Markiere die gefundenen Punkte
plot(phi0_found, targets * 100, 'ro', 'MarkerFaceColor', 'r', 'MarkerSize', 6);

% Beschriftung
xlabel('Initiale Auslenkung \phi_0/Grad');
ylabel('Abweichung der Periodendauer/%');
title('Relativer Fehler der Periodendauer gegenueber harmonischer Naeherung');
grid on;
xlim([0 180]);
ylim([0 110]); % Begrenzung fuer bessere Sichtbarkeit bis 100%

legend({'Verlauf der Abweichung', 'Gesuchte Punkte (1%, 5% etc.)'}, 'Location', '
    NorthWest');

% aus Anfangsauslenkung phi0 die exakte Periodendauer berechnen
function T = T_exact(phi0,g,l,T0)
    y0 = [phi0; 0];
    exact_func = @(t,y) [y(2); -(g/l)*sin(y(1))];
    opts = odeset('Events',@event_zero_down);
    tmax = 100*T0;
    [t, y, te, ye, ie] = ode45(exact_func, [0 tmax], y0, opts);
    tq = te(1);
    % Zeitpunkt des ersten Events (x -> -) = Viertelperiode daher mal 4
    T  = 4*tq;
end

function [value,isterminal,direction] = event_zero_down(~,y)
```

```
    value = y(1); % Event trifft auf wenn value = 0 also phi = 0
    isterminal = 1; % Integration stoppen
    direction = -1; % wenn value von + auf - geht
end
```

Listing 6: Plotting und Hilfsfunktionen

1.3 Aufgabe 3: Sekundenpendel

Sekundenpendel Analytisch

DGL: $\ddot{\varphi}+\frac{g}{l}\varphi = 0$. Lösung: $\varphi(t) = \varphi_0 \cos(\omega t)$. Kreisfrequenz: $\omega = \sqrt{\frac{g}{l}}$. Periodendauer einer Schwingung: $T = 2\pi\sqrt{\frac{l}{g}}$. Periodendauer eines Sekundenpendels: $T = 2\,\text{s}$. Einsetzen: $2 = 2\pi\sqrt{\frac{l}{g}} \implies 1 = \pi\sqrt{\frac{l}{g}}$. Umstellen auf Pendellänge:

$$l = \frac{g}{\pi^2}$$

```
format long
g = 9.81;
L_analytisch = g/pi^2;
fprintf('Pendellänge Analytisch: %.15f m\n', L_analytisch)
```

Listing 7: Analytische Länge

Sekundenpendel numerisch

Zur Bestimmung der Pendellänge wird mit `fzero` die Länge bestimmt, für welche die Periodendauer des Pendels genau 2 s beträgt.

```
format short
phi0 = 0.01;    % Startwinkel (rad)
Periodendauer = 2; %Periodendauer (sec)

% Länge des Sekundenpendels bestimmen
L_numerisch = fzero(@(L) Perioden_Fehlerfunktion(L,phi0, Periodendauer), [0.5 2]);

fprintf('Pendellänge Numerisch: %.15f m\n', L_numerisch);

%Absoluter Fehler und Realtiver Fehler:
Fehler_Absolut = abs(L_numerisch - L_analytisch);
fprintf('Absoluter Fehler: %.5f m\n', Fehler_Absolut);

Fehler_Relativ = Fehler_Absolut/L_analytisch;
Fehler_Relativ_Prozent = Fehler_Relativ*100;
fprintf('Relativer Fehler: %.5f bzw. %.5f Prozent \n', Fehler_Relativ,
    Fehler_Relativ_Prozent);
```

Listing 8: Numerische Berechnung der Länge

Länge des Sekundenpendels in Abhängigkeit von der Startauslenkung

Für den Plot der Länge in Abhängigkeit der Auslenkung wird mit einer Schleife die initiale Auslenkung variiert und anschließend die selbe Routine wie für das Sekundenpendel durchgeführt.

```
phi0_vec = linspace(phi0, pi/2, 30);    % Startauslenkung in rad
L_phi = zeros(size(phi0_vec));

for k = 1:length(phi0_vec)
```

```
    phi0 = phi0_vec(k);
    % Sekundenpendel-Länge für diese Auslenkung
    L_phi(k) = fzero(@(L) Perioden_Fehlerfunktion(L, phi0, Periodendauer), [0.5 2]);
end

figure;
plot(phi0_vec, L_phi, '-o');
xlabel('Initiale Auslenkung \phi_0 [rad]');
ylabel('Pendellänge L [m]');
grid on;
title('Länge des Sekundenpendels in Abhängigkeit der Startauslenkung');
```

Listing 9: Länge in Abhängigkeit der Amplitude

2 Bestimmung der Parameter aus Messdaten

Aufgabenstellung: Von einem Pendel mit der Länge $l = 1\,\text{m}$ sind Messdaten bekannt. Bestimmen Sie die Schwerebeschleunigung g.

2.1 Physikalischer Ansatz

Die Auswertung basiert auf dem Energieerhaltungssatz. Die Gesamtenergie eines Pendels ist konstant und setzt sich aus kinetischer und potenzieller Energie zusammen. Die Formel lautet:

$$E_{\text{ges}} = E_{\text{kin}} + E_{\text{pot}} = \frac{1}{2}ml^2\omega^2 + mgl(1 - \cos\varphi) = \text{const}$$

2.2 Methodik

Zur präzisen Bestimmung von g wurde die Energiegleichung linearisiert, um eine lineare Regression durchzuführen. Dabei wurden folgende Zuweisungen für die Achsen gewählt:

- Y-Achse: ω^2 (proportional zur kinetischen Energie)
- X-Achse: $1 - \cos(\varphi)$ (proportional zur potenziellen Energie bzw. der relativen Höhe)

Über die Steigung m_{reg} der resultierenden Regressionsgeraden lässt sich g direkt berechnen. Der Zusammenhang ist:

$$\omega^2 = -\frac{2g}{l}(1 - \cos\varphi) + C \implies m_{\text{reg}} = -\frac{2g}{l} \implies g = -m_{\text{reg}} \cdot \frac{l}{2}$$

2.3 Ergebnisse und MATLAB-Implementierung

Das Bestimmtheitsmaß R^2 liegt nahezu bei 1,0. Dies belegt eine extrem hohe Konsistenz der Daten mit dem physikalischen Modell. Der berechnete Wert für die Schwerbeschleunigung beträgt $g \approx 1{,}62\,\text{m/s}^2$.

```matlab
% Fadenpendel - Teil 4: Bestimmung von g aus Messdaten
clear; clc; close all;

% --- Messdaten ---
% Spalten: t [s], phi [rad], omega [rad/s]
data = [
    0.0000, 0.0000, 1.0000;
    0.2222, 0.2193, 0.9604;
    0.4444, 0.4213, 0.8466;
    0.6667, 0.5910, 0.6712;
    0.8889, 0.7163, 0.4513;
    1.1111, 0.7895, 0.2040;
    1.3333, 0.8061, -0.0549;
    1.5556, 0.7653, -0.3108;
    1.7778, 0.6694, -0.5485;
    2.0000, 0.5241, -0.7518
];

t = data(:, 1); phi = data(:, 2); omega = data(:, 3);
l = 1.0; % Pendellänge in m

% --- Linearisierung für Regression ---
% Gleichung: omega^2 = - (2g/l) * (1 - cos(phi)) + const
y_reg = omega.^2;
x_reg = 1 - cos(phi);
```

```
% --- Lineare Regression (polyfit) ---
p = polyfit(x_reg, y_reg, 1);
slope = p(1);
intercept = p(2);

% --- Bestimmung von g ---
% slope = -2*g / l  =>  g = -slope * l / 2
g_calculated = -slope * l / 2;

fprintf('Steigung der Regressionsgeraden: %.4f\n', slope);
fprintf('Ermittelte Schwerbeschleunigung g:  %.4f m/s^2\n', g_calculated);

% --- Visualisierung ---
figure('Name', 'Bestimmung von g');
subplot(1, 2, 2);
plot(x_reg, y_reg, 'kx', 'MarkerSize', 8, 'LineWidth', 2); hold on;
x_fit = linspace(min(x_reg), max(x_reg), 100);
y_fit = polyval(p, x_fit);
plot(x_fit, y_fit, 'r-', 'LineWidth', 1.5);
xlabel('1 - cos(\phi)'); ylabel('\omega^2 [(rad/s)^2]');
title(['Lineare Regression (g \approx ' sprintf('%.2f', g_calculated) ' m/s^2)']);
grid on;

% Zusatzcheck R^2
y_resid = y_reg - polyval(p, x_reg);
SofS_resid = sum(y_resid.^2);
SofS_total = sum((y_reg - mean(y_reg)).^2);
R2 = 1 - SofS_resid/SofS_total;
fprintf('Bestimmtheitsmass R^2:           %.4f\n', R2);
```

Listing 10: Lineare Regression zur Bestimmung von g

2.4 Interpretation

Der ermittelte Wert von etwa $1,62$ m/s^2 entspricht der Gravitation auf der Mondoberfläche. Die Analyse bestätigt, dass die Messdaten ein physikalisch korrektes Pendel unter Mondbedingungen beschreiben.

3 Doppelpendel

Das Doppelpendel besteht aus zwei aneinander befestigten Pendeln und ist ein klassisches Beispiel für ein physikalisches System, das chaotisches Verhalten zeigen kann. Die Bewegung wird durch zwei gekoppelte, nichtlineare Differentialgleichungen zweiter Ordnung beschrieben.

3.1 Mathematisches Modell

Um die Winkelbeschleunigungen $\ddot{\theta}_1$ und $\ddot{\theta}_2$ numerisch effizient zu berechnen, wird das Gleichungssystem im Code in die Matrixform $A \cdot x = b$ gebracht:

$$\begin{pmatrix} 1 & k_1 \cos(\delta) \\ k_2 \cos(\delta) & 1 \end{pmatrix} \cdot \begin{pmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \tag{3}$$

Hierbei ist $\delta = \theta_1 - \theta_2$. Die Hilfskonstanten und Vektorkomponenten der rechten Seite ergeben sich aus der physikalischen Herleitung (Lagrange-Formalismus) zu:

$$\begin{aligned} k_1 &= \frac{m_2}{m_1 + m_2} \cdot \frac{l_2}{l_1} \\ k_2 &= \frac{l_1}{l_2} \\ b_1 &= -k_1 \dot{\theta}_2^2 \sin(\delta) - \frac{g}{l_1} \sin(\theta_1) \\ b_2 &= k_2 \dot{\theta}_1^2 \sin(\delta) - \frac{g}{l_2} \sin(\theta_2) \end{aligned}$$

3.2 Untersuchte Fälle und Simulation

Für die Simulation wurden im MATLAB-Skript (`5_doppelpendel.m`) folgende Systemparameter gewählt, wobei insbesondere $l_1 = 2l_2$ gilt:

$$m_1 = m_2 = 1,0\,\text{kg}, \quad l_2 = 1,0\,\text{m}, \quad l_1 = 2,0\,\text{m}$$

Es wurden zwei Szenarien mit unterschiedlichen Anfangsbedingungen $y_0 = [\theta_1, \dot{\theta}_1, \theta_2, \dot{\theta}_2]^T$ simuliert:

Fall 1: Aktive Anregung

Anfangsbedingungen:

$$\theta_1 = 0, \quad \dot{\theta}_1 = 0, \quad \theta_2 = \pi, \quad \dot{\theta}_2 = 1$$

Die Simulation zeigt die Reaktion des Systems bei einer spezifischen Anfangsgeschwindigkeit des zweiten Pendels.

Fall 2: Freies Schwingen

Anfangsbedingungen:

$$\theta_1 = 0, \quad \dot{\theta}_1 = 1, \quad \theta_2 = \pi, \quad \dot{\theta}_2 = 0$$

Das System startet aus einer ausgelenkten Position mit Anfangsgeschwindigkeit am ersten Pendel.

3.3 MATLAB-Implementierung

Das Differentialgleichungssystem wird als Zustandsraummodell erster Ordnung formuliert und mittels `ode45` gelöst. In jedem Zeitschritt wird das lineare Gleichungssystem nach den Beschleunigungen aufgelöst.

```
% Globale Parameter und Initialisierung (Auszug)
p.g = 9.81;
p.m1 = 1.0; p.m2 = 1.0;
p.l2 = 1.0; p.l1 = 2 * p.l2; % Laenge 1 ist doppelt so gross wie Laenge 2

% Fall 1: Theta1=0, Theta2=pi, dTheta1=0, dTheta2=1
y0_case1 = [0; 0; pi; 1];

% Fall 2: Theta1=0, Theta2=pi, dTheta1=1, dTheta2=0
y0_case2 = [0; 1; pi; 0];

function dydt = pendulum_dynamics(~, y, p)
    theta1 = y(1); omega1 = y(2);
    theta2 = y(3); omega2 = y(4);

    delta = theta1 - theta2;
    sin_d = sin(delta); cos_d = cos(delta);

    % Matrix A aufstellen
    k1 = (p.m2 / (p.m1 + p.m2)) * (p.l2 / p.l1);
    k2 = p.l1 / p.l2;
    A = [1, k1 * cos_d; k2 * cos_d, 1];

    % Vektor b aufstellen
    b1 = -k1 * omega2^2 * sin_d - (p.g / p.l1) * sin(theta1);
    b2 = k2 * omega1^2 * sin_d - (p.g / p.l2) * sin(theta2);
    b = [b1; b2];

    % Lösen des LGS nach den Winkelbeschleunigungen
    alpha = A \ b;
    dydt = [omega1; alpha(1); omega2; alpha(2)];
end
```

Listing 11: Berechnung der Systemdynamik aus 5_doppelpendel.m