

## Projekt: Mechanisches System mit Zwangsbedingung

### 1 Einführung

Um ein mechanisches System zu beschreiben, gibt es im wesentlichen zwei Möglichkeiten: die Lagrange-Formulierung oder die Formulierung von Hamilton. Hier unten werden wir diese zwei Beschreibungen oberflächlich betrachten. Mehr Details kann man, z.B. auf Wikipedia, <http://www.damtp.cam.ac.uk/user/tong/dynamics.html> oder im Buch Klassische Mechanik, F. Kuypers finden.

#### 1.1 Lagrange Formalismus (1788)

Seien  $q_1, q_2, \dots, q_n$  die Ortskoordinaten eines Systems und  $\dot{q}_1, \dots, \dot{q}_n$  die Geschwindigkeiten. Wir definieren die *Funktion von Lagrange*

$$L(q, \dot{q}) = T(q, \dot{q}) - U(q),$$

wobei  $q = (q_1, \dots, q_n)^T$ ,  $\dot{q} = (\dot{q}_1, \dots, \dot{q}_n)^T$ ,  $T$  die kinetische Energie und  $U$  die potentielle Energie unseres Systems sind. Die Bewegung des Systems wird dann beschrieben durch die *Euler-Lagrange-Gleichung*

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{q}_k} \right) - \frac{\partial L}{\partial q_k} = 0 \quad \text{für } k = 1, \dots, n. \quad (1)$$

**Beispiel: Das mathematische Pendel aus der Vorlesung.** Wir betrachten ein Pendel der Länge  $\ell$  und Masse  $m$ . Wir nehmen als Position  $q_1 = \theta$  (siehe Abbildung 1).

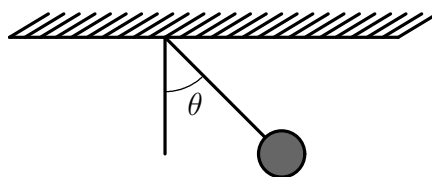


Abbildung 1: Das mathematische Pendel.

Die Energien sind dann  $T = \frac{m\ell^2}{2} \dot{\theta}^2$  und  $U = -\ell mg \cos(\theta)$ . Aus (1) folgt dann die Gleichung

$$\ell \ddot{\theta} = -g \sin(\theta),$$

vergleichen Sie mit der Vorlesung.

## 1.2 Hamilton Formalismus (1834)

Wir führen neue Variablen ein: die *Impulse*  $p_k = \frac{\partial L}{\partial \dot{q}_k}(q, \dot{q})$  für  $k = 1, \dots, n$ . Wir fassen die *Hamiltonsche Funktion*  $H = p^T \dot{q} - L(q, \dot{q})$  auf als Funktion von  $p$  und  $q$ .

(a) Zeigen Sie, dass die Euler-Lagrange Gleichungen (1) äquivalent zu den Hamiltonschen Gleichungen

$$\begin{aligned}\dot{p}_k &= -\frac{\partial H}{\partial q_k}(p, q) \\ \dot{q}_k &= \frac{\partial H}{\partial p_k}(p, q) \quad \text{für } k = 1, \dots, n\end{aligned}\tag{2}$$

sind.

## 2 Zwangsbedingung

Es ist jetzt Zeit um die Bewegung des Systems einzuschränken. Dafür nehmen wir an, dass wir  $m$  *Zwangsbedingungen* (ZB) haben:  $g_1(q) = 0, \dots, g_m(q) = 0$ .

### 2.1 Lagrange Formalismus

Die Idee von Lagrange war die folgende Funktion zu betrachten

$$L(q, \dot{q}, \lambda) = T(q, \dot{q}) - U(q) - \lambda_1 g_1(q) - \dots - \lambda_m g_m(q).\tag{3}$$

Die  $\lambda_j$  sind die *Lagrange-Multiplikatoren*. Eine wichtige Bemerkung ist, dass da  $L$  unabhängig von  $\dot{\lambda}_j$  ist, sind die Ableitungen nach  $\lambda_j$  in (1) gegeben als

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\lambda}_j}\right) - \frac{\partial L}{\partial \lambda_j} = 0 \quad \text{für } j = 1, \dots, m.$$

Also genau was wir brauchen: die ZB  $0 = g_j(q)$ ! Bemerkung: Die Lagrange-Multiplikatoren sind Unbekannte, die man bestimmen muss so, dass die ZB erfüllt sind.

**Beispiel: Das Pendel in kartesischen Koordinaten**  $(x, y)$ . Wir haben natürlich die ZB  $x^2 + y^2 - \ell^2 = 0$  (wir bewegen uns auf einem Kreis vom Radius  $\ell$ ) und die Funktion von Lagrange ist

$$L(x, y, \lambda) = \frac{m}{2}(\dot{x}^2 + \dot{y}^2) - mgy - \lambda(x^2 + y^2 - \ell^2).$$

Wir bekommen dann die Euler-Lagrange-Gleichung

$$\begin{aligned}m\ddot{x} &= -2x\lambda \\ m\ddot{y} &= -2y\lambda \\ 0 &= x^2 + y^2 - \ell^2.\end{aligned}$$

(Bemerkung: hier entspricht  $\lambda$  die Seilspannung.)

## 2.2 Hamilton Formalismus

Wie früher verwenden wir die Impulse  $p = \frac{\partial L}{\partial \dot{q}}$ .

(b) Wie lautet  $p$  für das Beispiel vom Pendel in kartesischen Koordinaten?

Nun betrachten wir  $H(p, q) + g_1(q)\lambda_1 + \dots + g_m(q)\lambda_m$  anstatt (3). Wir bekommen die sogenannte *Hamiltonsche Gleichung für ein System mit Zwangsbedingungen*

$$\begin{aligned}\dot{q}_k &= \frac{\partial H}{\partial p_k}(p, q) \\ \dot{p}_k &= -\frac{\partial H}{\partial q_k}(p, q) - G(q)^T \lambda \quad \text{für } k = 1, \dots, n \\ 0 &= g_j(q) \quad \text{für } j = 1, \dots, m,\end{aligned}\tag{4}$$

mit  $G(q) = \frac{\partial g}{\partial q}(q)$ ,  $g(q) = (g_1(q), \dots, g_m(q))^T$  und  $\lambda = (\lambda_1, \dots, \lambda_m)^T$ .

(c) Wie lauten diesen Gleichungen für das Pendel in kartesischen Koordinaten?

## 3 Numerisches Verfahren

Das Ziel ist jetzt die Gleichung (4) numerisch zu betrachten. Dafür passen wir unsere symplektisches Euler-Verfahren an. Man bekommt:

$$\begin{aligned}\hat{p}_{n+1} &= p_n - h(H_q^T(\hat{p}_{n+1}, q_n) + G(q_n)^T \lambda_{n+1}) \\ q_{n+1} &= q_n + hH_p^T(\hat{p}_{n+1}, q_n) \\ 0 &= g(q_{n+1}).\end{aligned}\tag{5}$$

Aus technischen Gründe verlangen wir noch

$$\begin{aligned}p_{n+1} &= \hat{p}_{n+1} - hG(q_{n+1})^T \mu_{n+1} \\ 0 &= G(q_{n+1})H_p^T(p_{n+1}, q_{n+1}).\end{aligned}\tag{6}$$

Die  $p$ -Variablen sind dann implizit gegeben und die  $q$ -Variablen explizit. Man muss noch die Unbekannten  $\lambda_{n+1}$  und  $\mu_{n+1}$  bestimmen so, dass die letzten Gleichungen in (5) und (6) erfüllt sind.

Wir betrachten nun ein sphärisches Pendel. Wir setzen  $q = (q_1, q_2, q_3)^T$  die kartesischen Koordinaten eines Punktes mit Masse  $m = 1$  verbunden mit dem Ursprung  $(0, 0, 0)$  durch einen masselosen Stab von Länge  $\ell = 1$ . Die Energien sind  $T = \frac{m}{2}(\dot{q}_1^2 + \dot{q}_2^2 + \dot{q}_3^2)$  und  $U = mgq_3$ .

(d) Leiten Sie die Hamiltonsche Gleichung (4) für dieses Problem her.

(e) Schreiben Sie ein Matlab Code `Pendel.m` um diese Gleichung zu lösen. Um die numerische Lösung zu überprüfen, plotten Sie dazu noch die gesamte Energie des Systems  $H(p, q)$ .

Die Struktur des Matlab-Programms könnte z.B. so aussehen:

```
Tend=10;h=0.01;N=floor(Tend/h);
% Anfangswerten
q=zeros(3,1);p=zeros(3,1);
q(1)=sin(1.3);q(2)=0;q(3)=cos(1.3);
```

```

p(1)=3*cos(1.3);p(2)=6.5;p(3)=-3*sin(1.3);
ham0= ... ; % Anfang gesamte Energie
...
spher0=norm(q); % um zu sehen ob man auf der Sphäre bleibt
...
for j=1:N
    % Bestimmung von lambda_{n+1} so, dass
    % g(q_{n+1})=0 (bei Hand, explizit!)
    ...
    % Berechnung von phut_{n+1} und q_{n+1}
    ...
    % Bestimmung von mu_{n+1} so, dass
    % G(q_{n+1})H_p(p_{n+1},q_{n+1})=0 (bei Hand, explizit!)
    ...
    % Berechnung von p_{n+1}
    ...
    % Berechnung Energie+Sphäre
    % Einspeichern für die Plots
    ...
end
% Plot Energie+Sphäre
...
% Schöne Visualisierung
fig=figure;set(fig,'doublebuffer','on')
xlim([-1 1]);ylim([-1 1]);zlim([-1 1])

p=plot3(q1plot(1),q2plot(1),q3plot(1),'ro');
for j=2:N
    set(p,'xdata',q1plot(1:j),'ydata',q2plot(1:j),'zdata',q3plot(1:j));
    drawnow;
end

```

## 4 Ein schwieriges Beispiel

Zum Schluss betrachten wir die Bewegung eines elektrischen Teilchens in einem elektrischen Feld. In kartesischen Koordinaten  $q = (q_1, q_2, q_3)^T$  bekommen wir die Hamiltonsche Funktion

$$H(p, q) = \frac{1}{2}((p_1 + q_2)^2 + (p_2 - q_1)^2 + p_3^2) - q_3$$

unter der ZB (man bleibt auf einer Sphäre)

$$g(q) = q_1^2 + q_2^2 + q_3^2 - 1 = 0.$$

Um dieses Problem numerisch zu integrieren, benutzen wir das sogenannte *(1,1)-Gauss-Lobatto SPARK*-Verfahren von L. Jay (2007). Ein Schritt dieses (impliziten) Verfahrens

lautet

$$\begin{aligned}
 Q_1 &= q_0 + \frac{h}{2} H_p^T(P_1, Q_1) = \frac{1}{2}(q_1 + q_0) \\
 P_1 &= p_0 - \frac{h}{2} H_q^T(P_1, Q_1) - \frac{h}{2} g_q^T(q_0) \Lambda_0 \\
 q_1 &= q_0 + h H_p^T(P_1, Q_1) \\
 0 &= g(q_1) \\
 p_1 &= p_0 - h H_q^T(P_1, Q_1) - \frac{h}{2} g_q^T(q_0) \Lambda_0 - \frac{h}{2} g_q^T(q_1) \Lambda_1 \\
 0 &= g_q(q_1) H_p^T(p_1, q_1).
 \end{aligned} \tag{7}$$

Wie früher bestimmen wir die Unbekannten  $\Lambda_0$  und  $\Lambda_1$  so, dass die vierte und die letzte Gleichung in (7) erfüllt sind. Dafür werden wir ein Newton Verfahren benutzen.

(f) Schreiben Sie ein Matlab Code `ElekTeilchen.m` um diese Gleichung zu lösen. Die Struktur des Matlab-Programms könnte z.B. so aussehen:

```

Tend=30;h=0.12;N=floor(Tend/h);
% Anfangswerten q0,p0 + Anfangsenergie ham0
q0=zeros(3,1);q0(1)=0.2;q0(2)=0.2;q0(3)=sqrt(0.92);
p0=zeros(3,1);p0(1)=1;p0(2)=-1;p0(3)=0;
ham0=...
% Erste Approximation mit explizites Euler
qit=zeros(3,1);pit=zeros(3,1);Qit=zeros(3,1);Pit=zeros(3,1);
qit=q0+h.*[p0(1)+q0(2);p0(2)-q0(1);p0(3)];
pit=p0;
Qit=q0+.5*h.*[p0(1)+q0(2);p0(2)-q0(1);p0(3)];
Pit=p0;
lambda0it=1.439;lambda1it=lambda0it;

F=zeros(14,1);Fprime=zeros(14,14);
y0=[Qit;Pit;qit;lambda0it;pit;lambda1it];

for j=1:N
    err=1;
    while ((abs(err) > 10^(-14)))
        %%%%%%%%%%%
        %% (1,1)-GL SPARK %%
        %%%%%%%%%%%
        F(1:3)=.5.*(qit+q0)-Qit;
        F(4)=p0(1)-.5*h*(Qit(1)-Pit(2))- ...
            h*q0(1)*lambda0it-Pit(1);
        F(5)=p0(2)-.5*h*(Qit(2)+Pit(1))- ...
            h*q0(2)*lambda0it-Pit(2);
        F(6)=p0(3)+.5*h-h*q0(3)*lambda0it-Pit(3);
        F(7)=q0(1)+h*(Pit(1)+Qit(2))-qit(1);
        F(8)=q0(2)+h*(Pit(2)-Qit(1))-qit(2);
        F(9)=q0(3)+h*Pit(3)-qit(3);
    end
end

```

```

F(10)=qit(1)^2+qit(2)^2+qit(3)^2-1;
F(11)=p0(1)-h*(Qit(1)-Pit(2))-h*q0(1)*lambda0it ...
    -h*qit(1)*lambda1it-pit(1);
F(12)=p0(2)-h*(Qit(2)+Pit(1))-h*q0(2)*lambda0it ...
    -h*qit(2)*lambda1it-pit(2);
F(13)=p0(3)+h-h*q0(3)*lambda0it ...
    -h*qit(3)*lambda1it-pit(3);
F(14)=2*(qit(1)*(pit(1)+qit(2))+ ...
    qit(2)*(pit(2)-qit(1))+ ...
    qit(3)*pit(3));

Fprime(1,:)=[-1 0 0 0 0 0 1/2 0 0 0 0 0 0 0];
Fprime(2,:)=[0 -1 0 0 0 0 0 1/2 0 0 0 0 0 0];
Fprime(3,:)=[0 0 -1 0 0 0 0 0 1/2 0 0 0 0 0];
Fprime(4,:)=[-h/2 0 0 -1 h/2 0 0 0 0 -h*q0(1) ...
    0 0 0 0];
Fprime(5,:)=[0 -h/2 0 -h/2 -1 0 0 0 0 -h*q0(2) ...
    0 0 0 0];
Fprime(6,:)=[0 0 0 0 0 -1 0 0 0 -h*q0(3) ...
    0 0 0 0];
Fprime(7,:)=[0 h 0 h 0 0 -1 0 0 0 0 0 0 0];
Fprime(8,:)=[-h 0 0 0 h 0 0 -1 0 0 0 0 0 0];
Fprime(9,:)=[0 0 0 0 0 h 0 0 -1 0 0 0 0 0];
Fprime(10,:)=[0 0 0 0 0 0 2*qit(1) 2*qit(2) ...
    2*qit(3) 0 0 0 0 0];
Fprime(11,:)=[-h 0 0 0 h 0 -h*lambda1it 0 0 ...
    -h*q0(1) -1 0 0 -h*qit(1)];
Fprime(12,:)=[0 -h 0 -h 0 0 0 -h*lambda1it 0 ...
    -h*q0(2) 0 -1 0 -h*qit(2)];
Fprime(13,:)=[0 0 0 0 0 0 0 0 -h*lambda1it ...
    -h*q0(3) 0 0 -1 -h*qit(3)];
Fprime(14,:)=[0 0 0 0 0 0 ...
    2*(pit(1)+qit(2))-2*qit(2) ...
    2*qit(1)+2*(pit(2)-qit(1)) ...
    2*pit(3) 0 ...
    2*qit(1) 2*qit(2) 2*qit(3) 0];
Fprime=sparse(Fprime);
% Newton Verfahren + Berechnung des Fehlers
...
end
% Berechnung der Energie+Einspeichern für die Plots
% Update die Daten
...
end
% Plot der Energie+Visualisierung
...;fig=figure;set(fig,'doublebuffer','on')

```

```
xlim([-1 1]);ylim([-1 1]);zlim([-1 1])

p=plot3(q1plot(1),q2plot(1),q3plot(1),'ro');
for j=2:N
    set(p,'xdata',q1plot(1:j),'ydata',q2plot(1:j),'zdata',q3plot(1:j));
    drawnow;
end
```

Am 26. oder 27. Januar 2009 erwarten wir eine kurze Präsentation (auf dem Rechner) und einen kleinen Bericht von Ihrem Projekt. *Besprechung Termine: 03.12.08, 17.12.08 und 19.01.09 um 17 : 30.*