Lagrange formalism

Description of mechanics more generally, with consideration of constraints. According to Newton, forced forces, such as the paths of a pendulum, would first have to be factored in and then eliminated again in the end.

If, on the other hand, a system with n freedom lines is generally described by n generalized coordinates and their velocities $q_1, \dot{q}_1 ... q_n, \dot{q}_n$, so you can save many coordinates (according to Newton about 3 dimensions). For a pendulum, for example, the generalized coordinate is simply the deflection $\varphi, \dot{\varphi}$, because the radius would be fixed and you only need one. So the coordinates do not have to be Cartesian, but you have to be able to define the kinetic and potential energy with them.

Because if one presses depending on $q, \dot{q}$ Now the kinetic T and the potential U The Lagrangian function defined as:

Lagrangian function

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

This is of great importance for us because using this directly gives the differential equations of the equations of motion for all n freedom lines. Derivation and proof of this **Euler-Lagrange equation**is listed below.

Euler-Lagrange equation

$\forall\ i = 1, ... N:$

$$\frac{d}{dt} \frac{\partial L(q, \dot{q})}{\partial \dot{q}_i} - \frac{\partial L(q, \dot{q})}{\partial q_i} = 0$$

$\frac{\partial L}{\partial \dot{q}}$ is also generalized impulse p called. Thus:

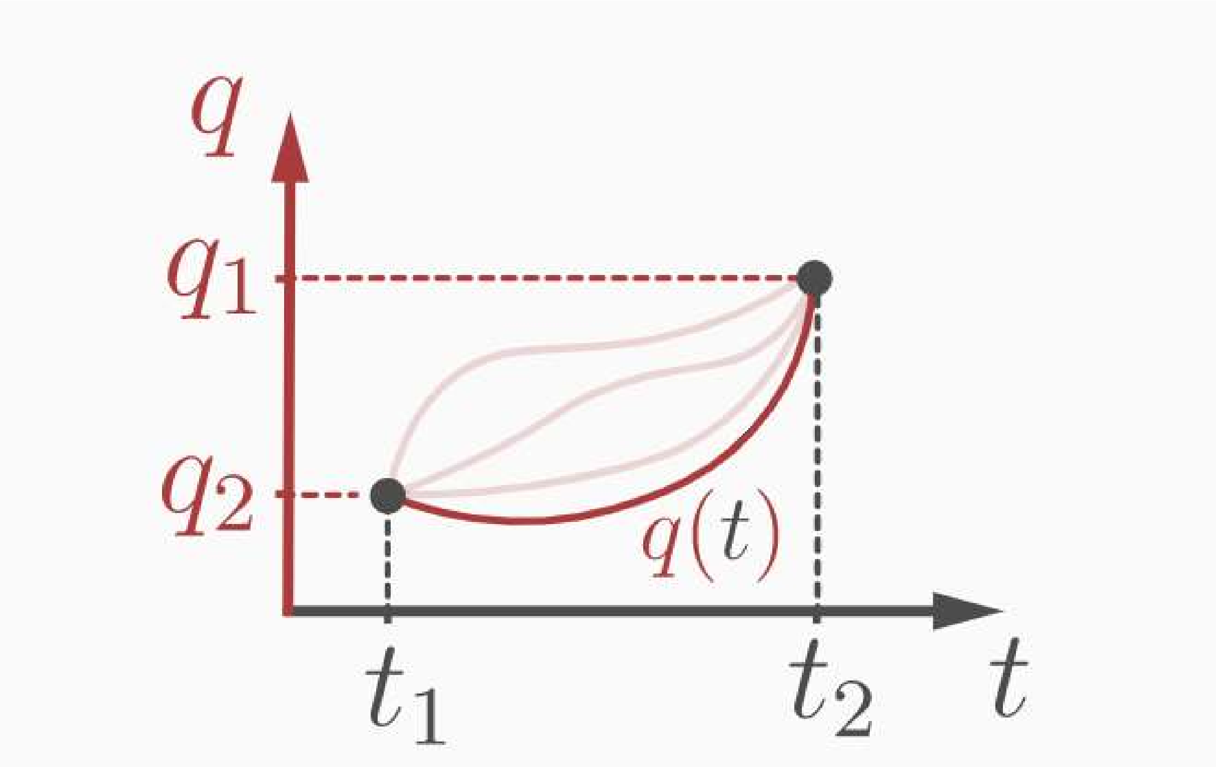
$$\frac{dp_i}{dt} = \frac{\partial L}{\partial q_i}$$

Derivation by Hamiltonian principle

Nature follows "extrema", and so we can argue here using the variation principle. Let's take the term effect as functional S to every conceivable trajectory q one

$$S[q(t)] = \int_a^b L(q, \dot{q})\ dt$$

so we can formulate that the system between two points $(t = a, q = q_a), (t = b, q = q_b)$ assumes the trajectory along which the effect S is stationary - i.e. the effect assumes an extrema. The derivative of the effect according to the strength of the variation is therefore zero, i.e. an infinitisimal change of the trajectory q to a variation q' does not change the effect.



Let's formalize the variation of a trajectory q' we can formulate the above requirement for the effect mathematically. Let's define our varied trajectory with

$$\begin{aligned} q'(t,\alpha) &= q(t) + \alpha \cdot \eta(t) && (1) \\ \Rightarrow \dot{q}'(t,\alpha) &= q(t) + \alpha \cdot \dot{\eta}(t) && (2) \end{aligned}$$

with the aid of any function $\eta(t)$whereby $\eta(a) = \eta(b) = 0$ so that the start and end point remain unchanged, so the strength of the variation depends on α from time to time to meet the extrema condition, we demand for $\alpha = 0$

$$\frac{\partial S[q'(t,\alpha)]}{\partial \alpha} \stackrel{!}{=} 0.$$

This directly requires the above differential equations: $\frac{d}{dt}\frac{\partial L(q,\dot{q})}{\partial \dot{q}_i} - \frac{\partial L(q,\dot{q})}{\partial q_i} = 0$ as we show below.

Proof

$L(q',\dot{q}') = L(q+\alpha\eta, \dot{q}+\alpha\dot{\eta})$ we get approximately from the Taylor development around $\alpha = 0$ around:

$$\begin{aligned} L(q+\alpha\eta, \dot{q}+\alpha\dot{\eta}) &= L(q+0\cdot\eta, \dot{q}+0\cdot\dot{\eta}) + \alpha \cdot \frac{\partial L(q+\alpha\eta, \dot{q}+\alpha\dot{\eta})}{\partial \alpha} + O(\alpha^2) && (3) \\ &= L(q,\dot{q}) + \alpha \cdot \frac{\partial L(q+\alpha\eta, \dot{q}+\alpha\dot{\eta})}{\partial \alpha} + O(\alpha^2) && (4) \end{aligned}$$

To be the total differential of $\frac{\partial L(q+\alpha\eta,\dot{q}+\alpha\dot{\eta})}{\partial \alpha}$ to calculate we use that for $(x,y) \mapsto f(x,y)$

$$df = \frac{\partial f}{\partial x}dx + \frac{\partial f}{\partial y}dy$$

Applies. We receive:

$$\frac{\partial L(q+\alpha\eta, \dot{q}+\alpha\dot{\eta})}{\partial\alpha} = \frac{\partial L}{\partial(q+\alpha\eta)} + \frac{\partial L}{\partial(\dot{q}+\alpha\dot{\eta})}$$

We solve this with the chain rule:

$$\begin{aligned}
\frac{\partial L}{\partial(q+\alpha\eta)} + \frac{\partial L}{\partial(\dot{q}+\alpha\dot{\eta})} &= \frac{\partial L}{\partial q}\cdot\frac{(\partial q+\alpha\eta)}{\partial\alpha} + \frac{\partial L}{\partial\dot{q}}\cdot\frac{(\partial\dot{q}+\alpha\dot{\eta})}{\partial\alpha} && (5)\\
&= \frac{\partial L}{\partial q}\cdot\eta + \frac{\partial L}{\partial\dot{q}}\cdot\dot{\eta} && (6)
\end{aligned}$$

we use the derivative in the summands of the Taylor development:

$$L(q,\dot{q}) + \alpha\cdot(\frac{\partial L}{\partial q}\cdot\eta + \frac{\partial L}{\partial\dot{q}}\cdot\dot{\eta}) + O(\alpha^2)$$

Um nun mit dem Integral die Wirkung berechnen zu können:

$$S[q(t)] = \int_a^b L(q,\dot{q}) + \alpha\cdot\frac{\partial L}{\partial q}\cdot\eta + \alpha\cdot\frac{\partial L}{\partial\dot{q}}\cdot\dot{\eta}\;dt + O(\alpha^2)$$

Die Ableitung der Wirkung nach α soll für $\alpha = 0$ wie oben gefordert gegen null gehen. Wir stellen die Gleichung auf:

$$\begin{aligned}
\lim_{\alpha\to 0}\frac{\partial\, S[q(t)]}{\partial\alpha} &= \frac{\partial}{\partial\alpha}\int_a^b L(q,\dot{q}) + \alpha\cdot\frac{\partial L}{\partial q}\cdot\eta + \alpha\cdot\frac{\partial L}{\partial\dot{q}}\cdot\dot{\eta}\;dt && (7)\\
&= \int_a^b \frac{\partial L(q,\dot{q})}{\partial\alpha} + \frac{\partial\alpha}{\partial\alpha}\cdot\frac{\partial L}{\partial q}\cdot\eta + \frac{\partial\alpha}{\partial\alpha}\cdot\frac{\partial L}{\partial\dot{q}}\cdot\dot{\eta}\;dt && (8)\\
&= \int_a^b \frac{\partial L}{\partial q}\cdot\eta + \frac{\partial L}{\partial\dot{q}}\cdot\dot{\eta}\;dt && (9)
\end{aligned}$$

Wir gehen im Weitere mit Partieller Integration vor. Beginnend mit dem zweiten Summanden:

$$\int_a^b \frac{\partial L}{\partial\dot{q}}\cdot\dot{\eta}\;dt$$

Zur Erinnerung für Partielle Integration:

$$\int_a^b u(x)v'(x)\,dx = \Big[u(x)v(x)\Big]_a^b - \int_a^b u'(x)v(x)\,dx \qquad (10)$$

wir können für unsere Anwendung einsetzen $u(x) = \frac{\partial L}{\partial\dot{q}}$ und $v'(x) = \cdot\dot{\eta}$, dann ist offensichtlich $v(x) = \eta$

so ergibt sich:

$$\begin{aligned}
\lim_{\alpha\to 0}\frac{\partial\, S[q(t)]}{\partial\alpha} &= \int_a^b \frac{\partial L}{\partial q}\cdot\eta + \frac{\partial L}{\partial\dot{q}}\cdot\dot{\eta}\;dt && (11)\\
&= \int_a^b \frac{\partial L}{\partial q}\cdot\eta\,dt + \Big[\frac{\partial L}{\partial\dot{q}}\cdot\eta\Big]_a^b - \int_a^b \frac{\partial L}{\partial\dot{q}}\cdot\eta\,dt && (12)
\end{aligned}$$

$\Big[\frac{\partial L}{\partial\dot{q}}\cdot\eta\Big]_a^b$ fällt weg, da $\eta(a) = \eta(b) = 0$ und wir können η im verbleibenden Therm ausklammern und, da $\lim_{\alpha\to 0}\frac{\partial\, S[q(t)]}{\partial\alpha}$ null ergibt die Gleichung gleich null setzen.

$$\lim_{\alpha \to 0} \frac{\partial\, S[q(t)]}{\partial \alpha} = \eta \cdot \int_a^b \frac{\partial L}{\partial q} \; - \frac{\partial L}{\partial \dot{q}}\, dt \tag{13}$$

$$= \eta \cdot \int_a^b \frac{\partial L}{\partial q} \; - \frac{\partial L}{\partial \dot{q}}\, dt = 0 \tag{14}$$

Somit folgt wie oben angekündigt die Euler Lagrange Gleichung aus dem Hamilton'schen Prinzip in dem wir gezeigt haben, dass die Wirkung der Trajektorie, dann ein Extremum annimmt, wenn für L von der Trajektorie gilt:

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} = 0$$

Hamilton Formalismus

Oben haben wir den Lagrangian kennengelernt. Die selbige Information ist handlicher und Symmetrische darstellbar als Hamiltonian, welche vom Lagrangian nur eine Legendre Transformation entfernt ist. Dafür würden wir gerne anstelle der Geschwindigkeit $\dot{q}$ den Impuls p verwenden, den wir oben bereits als $\frac{\partial L}{\partial \dot{q}}$ kennengelernt haben.

Die Legendre Transformation zu einer Funktion $f : x \mapsto y$ welche auf eine neue Funktion $f^* : \frac{df}{dx} \mapsto y$ abbildet ist gegeben durch:

$$f^*\Big(\frac{df}{dx}\Big) = \frac{df}{dx}x - f$$

Geometrisch bedeutet dies, dass wir zu einer gegebenen Funktion den y-Achsenabschnitt der Tangente an jedem Punkt von f berechnen bzw. die negation davon. So lässt sich bildlich vorstellen, dass dabei keine Information über die Funktion verloren geht, da man rückwärts die Funktion auch wieder herleiten kann.

Wenden wir diese Legendre Transformation auf $L(q, \dot{q})$ an, so wollen wir $\dot{q}$ ersetzen durch $\frac{\partial L}{\partial \dot{q}}$, um so den Impuls zu verwenden anstatt der Geschwindigkeit, da dieser leichter generalisiert werden kann. Dies

$$f^*\Big(q, \frac{\partial L}{\partial \dot{q}}\Big) = \frac{\partial L}{\partial \dot{q}}\dot{q} - L \tag{15}$$

$$= f^*(q,p) = p\dot{q} - L \tag{16}$$

$$= f^*(q,p) = \sum_i p_i \dot{q}_i - L \tag{17}$$

Dies nennen wir den Hamiltonian H:

$$H(q,p) = \sum_i p_i \dot{q}_i - L$$

daraus folgen mit einigen Umformungen die beiden Hamiltongleichungen

$$\dot{p}_i = -\frac{\partial H}{\partial q_i}, \dot{q}_i = \frac{\partial H}{\partial p_i}$$

bemerkenswert ist die Symmetrie, und dass für die Newtonsche Mechanik $p = m \cdot \dot{q}$ sowie $F = -\frac{\partial V(q)}{\partial q}$ aus dem Hamiltonian für einen Massepunkt die Newtonschen Gesetze folgen:

$$H = \frac{p^2}{2m} + V(x) \tag{18}$$

$$\Rightarrow \dot{p}_i = \frac{\partial H}{\partial q_i} = \frac{\partial V(q)}{\partial q} = F \tag{19}$$

$$\Rightarrow \dot{q}_i = \frac{\partial H}{\partial p_i} = \frac{\partial}{\partial p_i}\frac{p^2}{2m} = \frac{p}{m} \Leftrightarrow \ddot{q} = \frac{\dot{p}}{m} \Leftrightarrow \ddot{q}m = F \tag{20}$$

Nachweis

Um im Folgenden die Umformung vorzuführen, stellen wir zunächst ein paar Gleichungen auf die trivialerweise gelten müssen allgemein für ein delta δ einer Funktion H oder L gegenüber einem delta ihrer Eingabe Parameter:

$$\delta H = \frac{\partial H}{\partial q_i}\delta q_i + \frac{\partial H}{\partial p_i}\delta p_i \tag{21}$$

$$\delta L = \frac{\partial L}{\partial q_i}\delta q_i + \frac{\partial L}{\partial p_i}\delta p_i \tag{22}$$

Und speziell für H gleich dem Hamiltonian:

$$\delta H = \delta(p_i \dot{q}_i) - \delta L$$

und zuletzt, mithilfe der Produktregel:

$$\delta(p_i \dot{q}_i) = \dot{q}_i \delta p_i + p_i \delta \dot{q}_i$$

führen wir diese Gleichungen nun zusammen, können wir Formen:

$$0 = \delta H - \delta H \tag{23}$$

$$= \Big(\frac{\partial H}{\partial q_i}\delta q_i + \frac{\partial H}{\partial p_i}\delta p_i\Big) - \Big(\delta(p_i \dot{q}_i) - \delta L\Big) \tag{24}$$

$$= \Big(\frac{\partial H}{\partial q_i}\delta q_i + \frac{\partial H}{\partial p_i}\delta p_i\Big) - \Big(\delta(p_i \dot{q}_i) - (\frac{\partial L}{\partial q_i}\delta q_i + \frac{\partial L}{\partial p_i}\delta p_i)\Big) \tag{25}$$

$$= \Big(\frac{\partial H}{\partial q_i}\delta q_i + \frac{\partial H}{\partial p_i}\delta p_i\Big) - \Big(\dot{q}_i \delta p_i + p_i \delta \dot{q}_i - (\frac{\partial L}{\partial q_i}\delta q_i + \frac{\partial L}{\partial p_i}\delta p_i)\Big) \tag{26}$$

da wir definiert haben, dass $p_i = \frac{\partial L}{\partial \dot{q}}$, können wir dabei nutzen, dass

$$\frac{\partial L}{\partial q_i} = \frac{d}{dt}\frac{\partial L}{\partial \dot{q}_i} = \frac{d}{dt}p_i = \dot{p}_i \tag{27}$$

$$\frac{\partial L}{\partial \dot{q}_i} = p_i \tag{28}$$

gilt. Und damit vereinfachen:

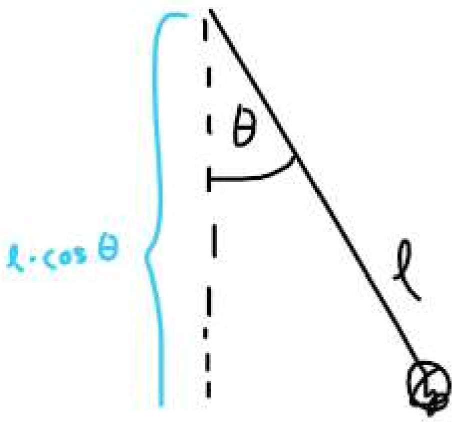
$$
\begin{align}
0 &= \Big(\frac{\partial H}{\partial q_i}\delta q_i + \frac{\partial H}{\partial p_i}\delta p_i\Big) - \Big(\dot{q}_i\delta p_i + p_i\delta\dot{q}_i - \dot{p}_i\delta q_i - p_i\delta p_i\Big) \tag{29}\\
&= \Big(\frac{\partial H}{\partial q_i}\delta q_i + \frac{\partial H}{\partial p_i}\delta p_i\Big) - \Big(\dot{q}_i\delta p_i - \dot{p}_i\delta q_i\Big) \tag{30}\\
&= \Big(\frac{\partial H}{\partial q_i} + \dot{p}_i\Big)\delta q_i + \Big(\frac{\partial H}{\partial p_i} - \dot{q}_i\Big)\delta p_i \tag{31}
\end{align}
$$

Die Lösung dieser Gleichung erfordert im Allgemeinen:

$$
\begin{align}
\Rightarrow 0 = \frac{\partial H}{\partial q_i} + \dot{p}_i \;\Rightarrow\; \dot{p}_i = -\frac{\partial H}{\partial q_i} \tag{32}\\
\Rightarrow 0 = \frac{\partial H}{\partial p_i} - \dot{q}_i \;\Rightarrow\; \dot{q}_i = \frac{\partial H}{\partial p_i} \tag{33}
\end{align}
$$

Wir erhalten unsere Hamiltongleichungen also durch geschicktes Umformen aus dem Lagrangian und Hamiltonian.

Beispiel am Starren Pendel



Wir wählen den Auslenkungswinkel $\theta(t)$ als generalisierte Koordinate für unser Pendel mit der Kantenlänge ℓ. Dann ergibt sich als Hamiltonian

$$H(\theta, p_\theta) = p_\theta\dot{\theta} - L(\theta, \dot{\theta}),$$

welcher den generalisierten Impuls p_θ zugehörig zu θ verwendet mit dem Lagrangian

$$L(\theta, \dot{\theta}) = T(\theta, \dot{\theta}(p_\theta)) - U(\theta);\;\; T(\theta, \dot{\theta}) = \frac{1}{2}m(\dot{\theta}\cdot\ell)^2;\;\; U(\theta) = mg\ell\cdot cos\,\theta$$

aus dem wir per Definition p_θ erhalten, und sehen, dass er ein Drehimpuls ist:
$Masse\ x\ Frequenz\ x\ Länge^2 = Länge\ x\ Masse\ x\ Geschwindigkeit$

$$
\begin{align}
p_\theta = \frac{\partial L}{\partial\dot{p}} = \frac{\partial T}{\partial\dot{\theta}} = \frac{1}{2}\frac{\partial(m\dot{\theta}^2\ell^2)}{\partial\dot{\theta}} = m\dot{\theta}\ell^2 \tag{34}\\
T(\theta, \dot{\theta}) = \frac{m^2\theta^2\ell^4}{2m\ell^2} \tag{35}\\
\Leftrightarrow T(\theta, p_\theta) = \frac{p_\theta^2}{2m\ell^2} \tag{36}
\end{align}
$$

direkt

$$H(\theta, p_\theta) = p_\theta \cdot \dot{\theta} - \frac{p_\theta^2}{2m\ell^2} - mg\ell \cdot cos\,\theta \tag{37}$$
$$= \frac{p_\theta^2}{2m\ell^2} - mg\ell \cdot cos\,\theta. \tag{38}$$

Wir setzten den Hamiltonian $H(\theta, p)$ nun in die Hamiltongleichungen ein:

$$\dot{p}_\theta = - \frac{\partial H}{\partial \theta} \Rightarrow \dot{p}_\theta = - \frac{\partial}{\partial \theta}\left(\frac{p_\theta^2}{2m\ell^2} - mg\ell \cdot cos\,\theta\right) \tag{39}$$
$$\dot{\theta} = \frac{\partial H}{\partial p} \Rightarrow \dot{\theta} = \frac{\partial}{\partial p_\theta}\left(\frac{p_\theta^2}{2m\ell^2} - mg\ell \cdot cos\,\theta\right) \tag{40}$$

und erhalten

$$\dot{p}_\theta = -mg\ell \cdot sin\,\theta \tag{41}$$
$$\dot{\theta} = \frac{p_\theta}{m\ell^2} \tag{42}$$
$$\Rightarrow \ddot{\theta} = -\frac{mg\ell \cdot sin\,\theta}{m\ell^2} = -g\ell \cdot sin\,\theta \tag{43}$$

Differenzialgleichungen. Mit diesen können wir nun zu einem Startzustand numerische Verfahren starten oder, da die Differenzialgleichung recht Simpel ist sie für einen gegebenen Startzustand lösen.

Die allgemeine Lösung wäre, wobei wir zur schmaleren Notation $g = \ell = m = \sqrt{g\ell} = \omega = 1$ setzen:

$$\theta(t) = a \cdot \cos(\omega t) + b \cdot sin(\omega t)$$

$$\dot{\theta}(t) = -a \cdot sin(t) + b \cdot cos(t)$$

für t = 0:

$$\theta(0) = a \cdot cos(0) + p_\theta(0) \cdot sin(0) = a = \theta_0$$

$$\dot{\theta}(0) = -a \cdot sin(0) + b \cdot cos(0) = b = p_\theta$$

$$\Rightarrow \theta(t) = \theta_0 \cdot \cos(\omega t) + p_\theta \cdot sin(\omega t)$$

Dimensionless representation

Note that now, under the conditions just determined, $g = \ell = m = \sqrt{g\ell} = \omega = 1$ Applies:

$$\dot{\theta} = p_\theta$$

$$\dot{p}_\theta = -sin\,\theta$$

$$T(\theta, p_\theta) + U(\theta) = \frac{p_\theta^2}{2} - cos\,\theta$$

In the following, we apply these differential equations to a pendulum object.

Physical laws of a pendulum

First we import 3 classes that help us to implement the laws of physics. Starting with "Arith", which helps us to set up arithmetic thermal baths, which are preserved with changing variables.

In [ ]:
```
from ArithmeticFunction import Arith

# Damit können wir einen Therm c definieren, der von a und b abhängt und auf deren Änder

a = Arith(2)
b = Arith(1)

c = a * b
print(c)
b.value += 1
c.refresh()
print(c)

# die Arith Objekte (Atome des Therms und Verkettungen dieser) können wir dabei mit alle
```

```
2.000
4.000
```

This is very useful for defining differential equations. Which we formalize with the class "diffEq":

In [ ]:
```
from DifferentialEquation import diffEq

a = Arith(2)
b = Arith(0)
print(f't0: a={a}, b={b}')
# wir können nun zB deffinieren, dass d/dt a = b sei:
diff1 = diffEq(a, b)

# und d/dt b = - a sei:
diff2 = diffEq(b, Arith(-1) * a)


# dann können wir d/dt a, d/dt b berechnen lassen, das Ergebnis wird in myDiffEq gespeic
diff1.calculate()
diff2.calculate()

# und zu einem beliebigeren späteren Zeitpunkt die Differential für einen Zeitraum dt an
# Wichtig ist, dass wir mit .calculate und .apply die Berechnung und Anwendung Zeitlich
# so beeinflussen sich die Differenzialgleichungen nicht ungewohlt und nur zu gewünschte
dt = 0.5
diff1.apply(dt)
diff2.apply(dt)
print(f't1: a={a}, b={b}')

# über ein paar weitere Zeitschritte können wir eine zu erwartende Bewegung beobachten.
# diese ist unabhängig von einem Physikalischen Kontext, kann aber ggf einen bestimmten
# sie ist jedenfalls wie man auch leicht an den Zahlen erkennt kein harmonischer Oszilla

for i in range(2, 10):
    diff1.calculate()
    diff2.calculate()
    diff1.apply(dt)
    diff2.apply(dt)
    print(f't{i}: a={a}, b={b}')
```

```
t0: a=2.000, b=0.000
t1: a=2.000, b=-1.000
t2: a=1.500, b=-2.000
t3: a=0.500, b=-2.750
t4: a=-0.875, b=-3.000
t5: a=-2.375, b=-2.562
t6: a=-3.656, b=-1.375
t7: a=-4.344, b=0.453
t8: a=-4.117, b=2.625
t9: a=-2.805, b=4.684
```

In order to define a specific real physical context, we use the "Body" class and create a sub-class "Pendulum".

Every body object, has properties "props" and specifically for "pendulum", we define these as angle "theta", and momentum "p", as well as energy "energy".
We directly implement the two differential equations set up above and store them in each pendulum, easily accessible for further action.

In [ ]:
```python
from PhysicalBody import Body

class Pendulum(Body):

    params = ["theta", "p", "energy"]
    units = ["radians", "mass * length^2 * Omega", "mass * g * length"]
    timeUnit = "1/Omega"

    def __init__(self, name, theta0, p0):
        #memorize initial for id
        self.th0 = theta0
        self.p0 = p0

        theta = Arith(theta0, self.params[0])
        p = Arith(p0, self.params[1])

        # Energy of Pendulum
        energy = p * p / Arith(2) + (theta.cos() * Arith(-1))
        energy.name = self.params[2]

        Body.__init__(self, name, [theta, p])

        self.addDifferential(diffEq(target=p,term=theta.sin() * Arith(-1)))
        self.addDifferential(diffEq(target=theta,term=p))

    def __str__(self):
        return f'th0={self.th0},p0={self.p0}'
```

Numerical methods

In [ ]:
```python
import numpy as np
from Simulation import Simulation
```

Euler method

to the given θ_t, p_t Results θ_{t+dt}, p_{t+dt} according to the regulation.

$$\theta_{t+dt} = \theta_t + \dot{\theta}(p_t) \cdot dt$$

$$p_{t+dt} = p_t + \dot{p}(\theta_t) \cdot dt$$

With this and a θ_0, p_0 can we then numerically calculate for n Time steps all deflection angles and pulses up to $\theta_{t+n\cdot dt}, p_{t+n\cdot dt}$ calculate.

In [ ]:
```
class EulerSimulation(Simulation):
    def __init__(self, bodys:list[Body], endtime, n):
        Simulation.__init__(self, bodys, "Euler", endtime/ n, n)

    def simulate(self):
        for i in range(self.n):
            for b1 in self.bodys:
                for p in b1.props:
                    b1.props[p].refresh()
                    self.plotBodys[b1.name][p][i] = b1.props[p].value
                for d1 in b1.differentials:
                    d1.calculate()
            for b2 in self.bodys:
                for d2 in b2.differentials:
                    d2.apply(self.dt)
```

Violation procedure

similar to the Euler method, but we shift an axis by $dt/2$:

To the given extent $\theta_t, p_{t+dt/2}$ Results $\theta_{t+dt}, p_{t+dt+dt/2}$ according to the regulation.

$$\theta_{t+dt} = \theta_t + \dot{\theta}(p_{t+dt/2}) \cdot dt$$

$$p_{t+dt+dt/2} = p_{t+dt/2} + \dot{p}(\theta_t) \cdot dt$$

To initially $p_{t_0+dt/2}$, we calculate at the beginning of the procedure:

$$p_{0+dt/2} = p_0 + \dot{p}(\theta_0) \cdot dt$$

So we can now start with a θ_0, p_0 can we then numerically calculate for n Time steps all deflection angles and pulses up to $\theta_{t+n\cdot dt}, p_{t+n\cdot dt}$, whereby we $p_{t+n\cdot dt+dt/2}$ $p_{t+n\cdot dt}$ name

In [ ]:
```
class VerletSimulation(Simulation):

    def __init__(self, bodys:list[Body], endtime, n):
        Simulation.__init__(self, bodys, "Verlet", endtime/ n, n)

    def simulate(self):
        for b in self.bodys:
            # #for all bodys calculating first differential
            b.differentials[0].calculate()
        #special  Verlet step:
        for b in self.bodys:
            #offset all first differentials for half step after calculations have been m
            b.differentials[0].apply(self.dt/2)
        #continue like euler
        for i in range(self.n):
            for b in self.bodys:
                for p in b.props:
                    b.props[p].refresh()
```

```
                    self.plotBodys[b.name][p][i] = b.props[p].value

            for b in self.bodys:
                for d in b.differentials:
                    d.calculate()
                    d.apply(self.dt)
```

Plotter Class

In [ ]:

```
import matplotlib
from matplotlib import pyplot as plt

class Plotter:
    def __init__(self, simulations: list[Simulation]):
        self.sims = simulations

    def plotProp(self, idx=0):
        fig , ax = plt.subplots()
        for s in self.sims:
            for b in s.bodys:
                ax.plot(s.time, s.plotBodys[b.name][b.params[idx]], label=f'{b} - {s.typ
        ax.set_xlabel(f'time [{s.bodys[0].timeUnit}]')
        ax.set_ylabel(f'{s.bodys[0].params[idx]} [{s.bodys[0].units[idx]}]')
        ax.legend()
        plt.show()

    def plot2DPhase(self):
        fig , ax = plt.subplots()
        for s in self.sims:
            for b in s.bodys:
                ax.plot(s.plotBodys[b.name][b.params[0]], s.plotBodys[b.name][b.params[1
        ax.set_xlabel(f'{s.bodys[0].params[0]} [{s.bodys[0].units[0]}]')
        ax.set_ylabel(f'{s.bodys[0].params[1]} [{s.bodys[0].units[1]}]')
        ax.legend()
        plt.show()
```

Euler - Violation - Comparison

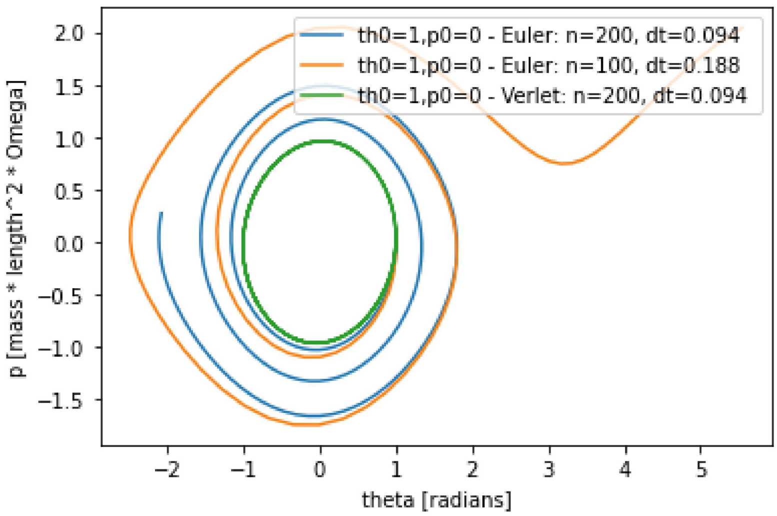
In [ ]:

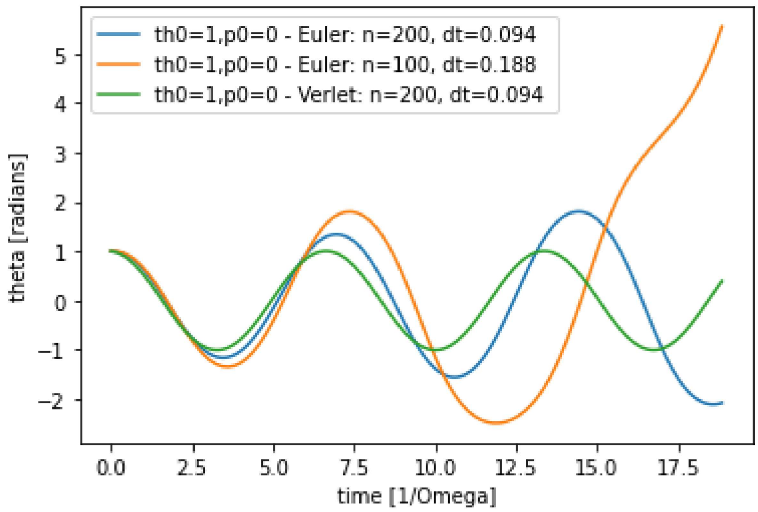
```
myEuler1 = EulerSimulation([Pendulum("Pendel", 1, 0)], 6*np.pi, 200)
myEuler1.simulate()

myEuler2 = EulerSimulation([Pendulum("Pendel", 1, 0)], 6*np.pi, 100)
myEuler2.simulate()

myVerlet = VerletSimulation([Pendulum("Pendel", 1, 0)], 6*np.pi, 200)
myVerlet.simulate()


myPlotter = Plotter([myEuler1, myEuler2, myVerlet])
myPlotter.plot2DPhase()
myPlotter.plotProp()
```


th0=1,p0=0 - Euler: n=200, dt=0.094
th0=1,p0=0 - Euler: n=100, dt=0.188
th0=1,p0=0 - Verlet: n=200, dt=0.094
p [mass * length^2 * Omega]
theta [radians]


th0=1,p0=0 - Euler: n=200, dt=0.094
th0=1,p0=0 - Euler: n=100, dt=0.188
th0=1,p0=0 - Verlet: n=200, dt=0.094
theta [radians]
time [1/Omega]