

FY1020 Mekanikk Numerisk Øving 1

Fredrik Scheie-Christiansen

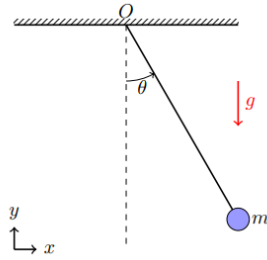
Oskar William Berg

Februar 2026

Innhold

Oppgave 1, Enkel Pendel	4
1.1	4
Kraftregnskap	4
1.2	5
1.3	5
a	5
b	6
Oppgave 2, Pendel fra fri bjelke	9
2.1	9
2.2	9
2.3	10
2.4	11

All kode brukt finnes i denne Github repoen.



Figur 1: Enkel pendel

Oppgave 1, Enkel pendel

1.1

Illustrasjonen i 1 viser en enkel pendel fastmontert til en bjelke i et punkt O . Vi modellerer pendelen som en punktmasse $m = 0.1$ kg som er forbundet med en masseløs stang med lengde $\ell = 0.1$ fra O til punktmassen. Gravitasjonen virker nedover med en akselerasjon $g = 9.81 \text{ ms}^{-2}$. Vi ønsker å studere kreftene som virker på kulen og på bjelken ved vinkelutslag θ . På kulen er det gravitasjonskraften $G = -mg$ som virker nedover i y retning, og snordraget S som virker parallelt med snoren, og bidraget i x og y retning avhenger følgelig av vinkelutslaget θ . Bjelken har selvfølgelig også en gravitasjonskraft G_b , men den er irrelevant for systemet da bjelken er fastmontert. I tillegg virker snordraget nedover parallelt med snora, og vi må ha $S_b = -S$, siden snoren er stram. Vi kan også da merke oss at denne S_b kraften vil gi opphav til en kraft som virker horisontalt på bjelken, men det vil igjen være irrelevant da bjelken er fastmontert. Men vi skal se senere at den vil komme til nytte. Vi kan dermed sette opp et kraftregnskap.

Kraftregnskap

Kule

$$\begin{aligned} y : \quad G &= -mg \\ S_y &= S \cos \theta \\ x : \quad S_x &= S \sin \theta \end{aligned}$$

Bjelke

$$\begin{aligned} y : \quad G_b &= -Mg \\ S_{by} &= -S \cos \theta \\ N &= -G_b - S_{by} \\ x : \quad S_{bx} &= S \sin \theta \\ N_x &= -S_{bx} \end{aligned}$$

Normalkreftene på bjelken er bare de nødvendige motkreftene som oppstår fra at bjelken er fastspent.

1.2

Vi ønsker å finne et uttrykk for vinkelutslaget θ . For å oppnå dette kan vi studere rotasjonen. Merk først at snordraget virker innover mot sentrum av bevegelsen, og bidrar dermed ikke til noe dreiemoment. Så vi ser på gravitasjonskraften G . Denne kan dekomponeres i en retning tangentielt på bevegelsen, og normalt på bevegelsen. Den normale komponenten gir da heller ikke noe dreiemoment og vi står igjen med den tangentielle komponenten som åpenbart blir

$$G_t = G \sin \theta = -mg \sin \theta. \quad (1)$$

Vi kan så bruke Newtons andre lov for rotasjon til å komme fram til en differensiallikning for θ ;

$$\begin{aligned} \sum \tau &= I\alpha \\ G_t \ell &= m\ell^2 \frac{\partial^2 \theta}{\partial t^2} \\ -g \sin \theta &= \ell \ddot{\theta} \\ \ddot{\theta} &= -\omega_0^2 \sin \theta, \end{aligned} \quad (2)$$

hvor ω_0 er den naturlige frekvensen til denne harmoniske oscillatoren, og denne er selvfølgelig gitt ved

$$\omega_0 = \sqrt{\frac{g}{\ell}} \quad (3)$$

1.3

a)

Vi ønsker å løse 2 numerisk. Vi holder det enkelt og bruker eksplisitt Euler i denne omgang;

$$x_{n+1} = x_n + hf(x_n).$$

Men umiddelbart ser vi jo at dette blir litt vrient når dobbelderiverte er involvert. Heldigvis kan selv et barn i barnehagen se at om vi lar $x = \theta$ og $y = \dot{\theta}$, kan vi lage systemet

$$\begin{aligned} \dot{x} &= y \\ \dot{y} &= -\omega_0^2 \sin x, \end{aligned} \quad (4)$$

som enkelt lar seg løse i Python:

```
def euler(theta_0,n,h,l):
    w0 = np.sqrt(G/l)
    theta = np.zeros(n)
    theta[0]=theta_0
    theta_hjelp = np.zeros(n)

    for i in range(1,len(theta)):
        theta[i]=theta[i-1] + h*theta_hjelp[i-1]
        theta_hjelp[i]=theta_hjelp[i-1]-h*(w0**2)*np.sin(theta[i-1])
    t = np.arange(0, n*h, h)

    return theta,t
```

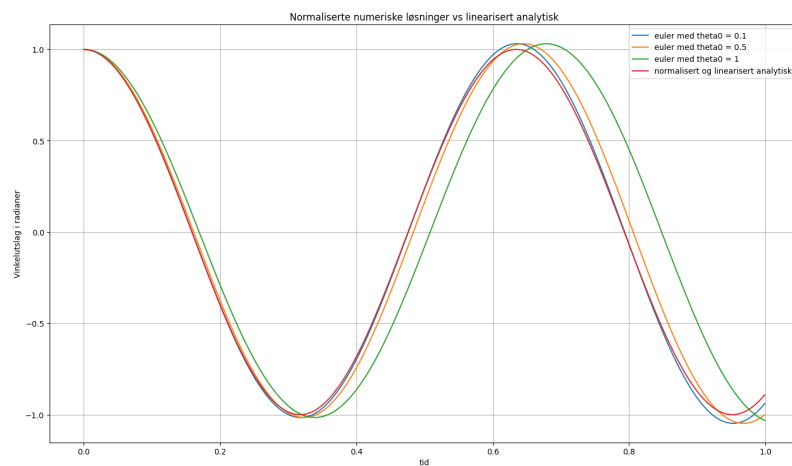
Vi er også interessert i å sammenligne de numeriske løsningene med den analytiske løsningen vi kan skaffe ved en linearisering $\sin \theta \approx \theta$ for $\theta \ll 1$. Denne ser vi lett at blir

$$\theta = A \sin(\omega_0 t) + B \cos(\omega_0 t),$$

som med initialbetingelser $\theta(0) = \theta_0$, og $\dot{\theta}(0) = 0$ gir

$$\theta = \theta_0 \cos(\omega_0 t). \quad (5)$$

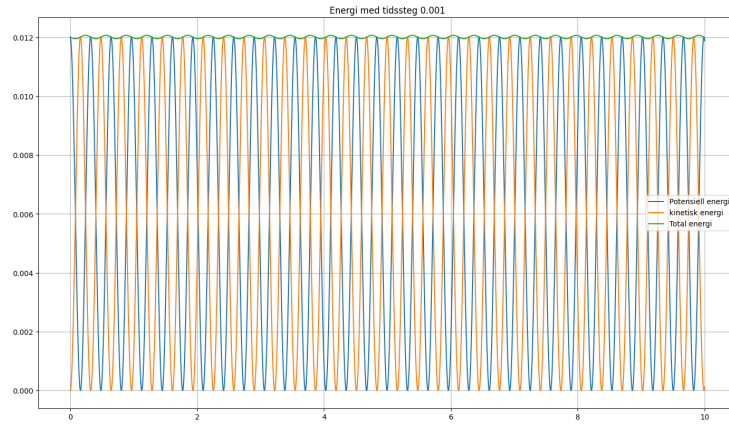
Vi sammenligner den analytiske løsningen med den numeriske for $\theta_0 = \{0.1, 0.5, 1.0\}$, men normalisert med θ_0 . Som vi ser gir mindre verdier for θ_0 en nærmere løsning til den analytiske,



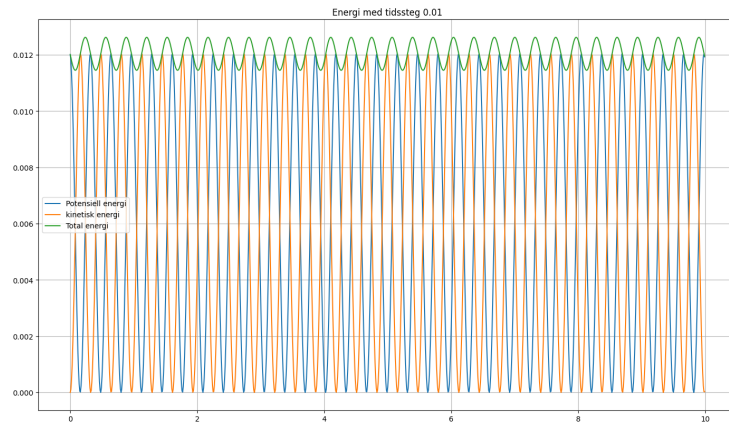
Figur 2: Numeriske løsninger med $h = 0.001$, 10 sekunders intervall

som gir mening når man bruker approksimasjoner nettopp for små vinkler. Ellers er det akkurat som forventet at når man gir større initiell utslag så vil også perioden bli lengre, siden den også vil svinge mer.

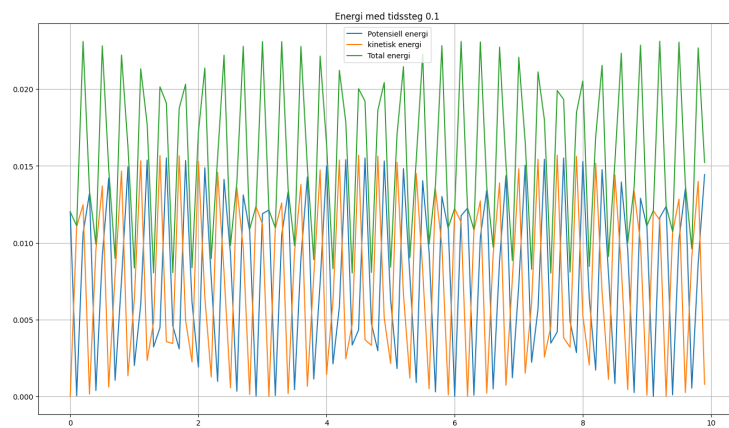
b)



(a) $h = 10^{-3}$



(b) $h = 10^{-2}$



(c) $h = 10^{-1}$

Figur 3: Sammenligning av energiutvikling for ulike tidssteg ved bruk av symplektisk Euler.

Vi undersøker energiutviklingen i systemet numerisk for tidssteg $h = \{.001, .01, .1\}$. Vi legger nullpunkt i bunnen av svingebevegelsen, slik at vi får

$$U = mgl(1 - \cos(\theta))$$

$$K = \frac{1}{2}I\omega^2 = \frac{1}{2}m\ell^2\dot{\theta}^2.$$

For å gjøre dette numerisk vil symplektisk Euler;

$$\omega_{n+1} = \omega_n + h \cdot a(\theta_n)$$

$$\theta_{n+1} = \theta_n + h\omega_{n+1},$$

være mer nøyaktig. I kode ser det slik ut:

```
T = 10
def energi(theta_0, n,h,l,m):
    w0 = np.sqrt(G/l)
    theta = np.zeros(n)
    theta[0]=theta_0
    theta_hjelp = np.zeros(n)

    for i in range(1,len(theta)):
        theta_hjelp[i]=theta_hjelp[i-1]-h*(w0**2)*np.sin(theta[i-1])
        theta[i]=theta[i-1] + h*theta_hjelp[i] #symplektisk euler

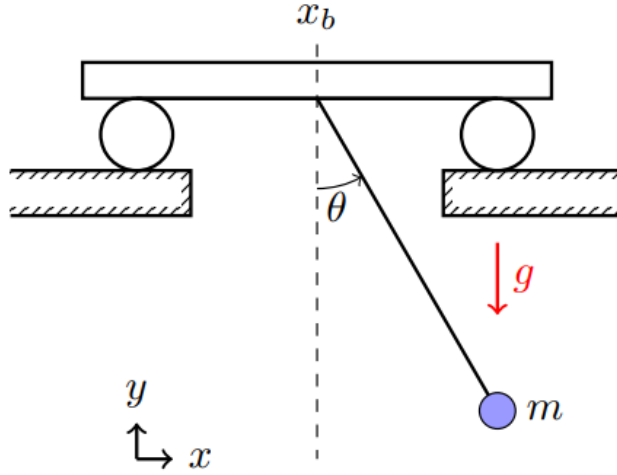
    t = np.arange(0, n*h, h)

    U=m*G*l*(1-np.cos(theta))
    K=.5*m*l**2*theta_hjelp**2
    return U,K,t
```

Det er fordi om vi skulle brukt eksplisitt Euler ville det ført til at feil akkumuleres over tid og den ville eksplodert. Det hadde sett ut som systemet blir tilført energi fra utsiden, noe det ikke gjør. Faseplottet til eksplisitt metode blir en utadgående spiral, men med symplektisk blir det tilnærmet en lukket kurve som er en mer realistisk representasjon av systemet vi studerer. Som vi ser får vi ikke en helt konstant verdi for den totale mekaniske energien, men det oscillerer rundt en fast verdi, og vi kan også se at mindre tidssteg gjør amplituden i denne oscilleringen mindre. Merk også at systemet bryter helt samme når $h = 0.1$. Trunkteringsfeilen blir for stor til at vi får en nøyaktig representasjon, men den unngår forstøt å eksplodere slik den eksplisitte metoden gjør.

Oppgave 2, Pendel fra fri bjelke

2.1



Figur 4: Pendel fra fri bjelke

Vi ser nå på et likt system, bortsett fra at bjelken ikke lengre er fastspennet. Bjelken har masse $M = 1\text{ kg}$ og vi neglesjierer friksjon og treghetsmomentet til sylindrene. Kraftbalansen er for det meste den samme som i forrige oppgave, men det finnes ikke lengre noen normalkraft i x retning på bjelken. Dermed vil den horisontale komponentet som snordraget bidrar med akselerere bjelken horisontalt. Men når bjelken akselererer horisontalt, og pendelen er festet til bjelken, så vil en såkalt imaginær kraft oppstå horisontalt på punktmassen m , som ved Newtons tredje lov må virke i motsatt retning av akselerasjonen til bjelken. Dermed er det ikke lengre kun gravitasjonen som gir opphav til et dreiemoment på pendelen, det er også den tangentielle komponenten til den imaginære kraften F_{im} .

2.2

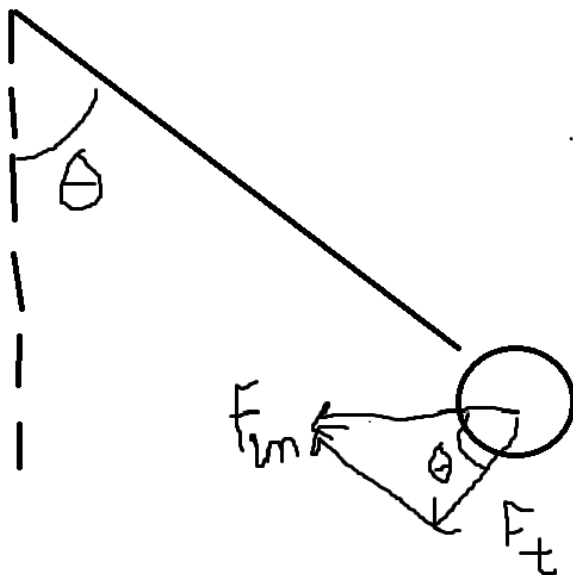
Vi ønsker å komme fram til en differensiallikning som beskriver systemet. Først observerer vi at det er ingen ytre krefter som virker på systemet. Dermed er bevegelsesmengden til massesenteret bevart. Vi ser da at

$$\begin{aligned}x_{CM} &= \frac{Mx_b + mx}{M + m} \\(M + m)x_{CM} &= Mx_b + mx \\(M + m)\dot{x}_{CM} &= M\dot{x}_b + m\dot{x} \\M\ddot{x}_b &= -m\ddot{x},\end{aligned}\tag{6}$$

siden den deriverte av bevegelsesmengden er null. Men siden x er pendelen sin x posisjon, som vi vet kan uttrykkes som $x_b + \ell \sin \theta$, ser vi at

$$\begin{aligned}M\ddot{x}_b &= -m\frac{d^2}{dt^2}(x_b + \ell \sin \theta) \\(M + m)\ddot{x}_b &= -m\ell(\ddot{\theta} \cos \theta - \dot{\theta}^2 \sin \theta) \\ \ddot{x}_b &= \frac{m\ell}{M_{tot}}(\dot{\theta}^2 \sin \theta - \ddot{\theta} \cos \theta).\end{aligned}$$

Dette er bra. Nå må vi bare undersøke θ littegrann. La oss se på krefetene som gir opphav til dreiemoment for pendelen. Som tidligere har vi åpenbart $-mg \sin \theta$. Men hva med fra F_{im} ? Vi



Figur 5: Dekomponering av F_{im}

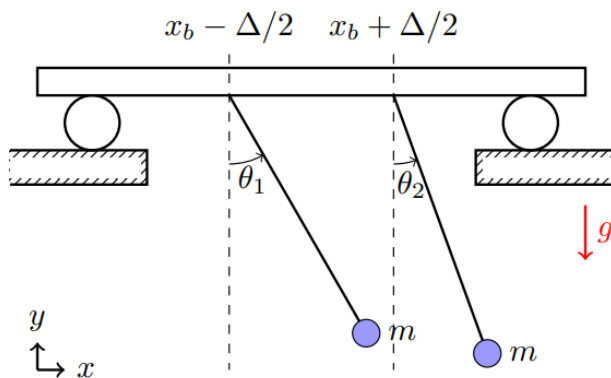
dekomponerer den. Observer i vakre figur 5. Vi kaller den tangentielle komponentet F_t . Da ser vi at vi har $F_t = F_{im} \cos \theta = -m\ddot{x}_b \cos \theta$. Men da kan vi igjen bruke Newtons andre for rotasjon,

$$\begin{aligned}
 I\ddot{\theta} &= \sum \tau \\
 m\ell^2\ddot{\theta} &= (-mg \sin \theta - m\ddot{x}_b \cos \theta)\ell \\
 \ddot{\theta} &= -\frac{g}{\ell} \sin \theta - \frac{m \cos \theta}{M_{tot}}(\dot{\theta}^2 \sin \theta - \ddot{\theta} \cos \theta).
 \end{aligned} \tag{7}$$

Og da har vi et system av nødvendige differensiallikninger for å beskrive systemet.

2.3

Forrige system var ikke komplisert nok. Vi introduserer enda en pendel, slik at det nå er to av dem som henger like langt unna massesenteret til bjelken x_b . Men frykt ikke, for systemet



Figur 6: Enda en pendel

vil oppføre seg nogelunde likt. For det første er det fortsatt ikke noen ytre kraft, som betyr at

bevegelsesmengden til massesenteret er bevart, slik at vi trenger kun å modifisere likning 6 litt;

$$\begin{aligned}
 x_{CM} &= \frac{Mx_b + m(x_1 + x_2)}{M + 2m} \\
 M_{tot}x_{CM} &= Mx_b + m(x_b - \frac{\Delta}{2} + \ell \sin \theta_1) + m(x_b + \frac{\Delta}{2} + \ell \sin \theta_2) \\
 M_{tot}x_{CM} &= M_{tot}x_b + m\ell(\sin \theta_1 + \sin \theta_2) \\
 \Rightarrow M_{tot}\ddot{x}_b &= -m\ell(\ddot{\theta}_1 \cos \theta_1 + \ddot{\theta}_2 \cos \theta_2 - \dot{\theta}_1^2 \sin \theta_1 - \dot{\theta}_2^2 \sin \theta_2) \\
 \ddot{x}_b &= \frac{m\ell}{M_{tot}}(\dot{\theta}_1^2 \sin \theta_1 + \dot{\theta}_2^2 \sin \theta_2 - \ddot{\theta}_1 \cos \theta_1 - \ddot{\theta}_2 \cos \theta_2).
 \end{aligned}$$

Så da trenger vi bare Komme fram til en differensiallikning for θ_1 og θ_2 . Men faktisk har vi allerede det, fordi de vil være like som i likning 7, bare at \ddot{x}_b vil nå være en funksjon av begge pendlene sin vinkel, og dermed vil vi få et system av differensiallikninger:

$$\begin{aligned}
 \ddot{\theta}_1 &= -\frac{g}{\ell} \sin \theta_1 - \frac{m}{M_{tot}} \cos \theta_1 (\dot{\theta}_1^2 \sin \theta_1 + \dot{\theta}_2^2 \sin \theta_2 - \ddot{\theta}_1 \cos \theta_1 - \ddot{\theta}_2 \cos \theta_2) \\
 \ddot{\theta}_2 &= -\frac{g}{\ell} \sin \theta_2 - \frac{m}{M_{tot}} \cos \theta_2 (\dot{\theta}_1^2 \sin \theta_1 + \dot{\theta}_2^2 \sin \theta_2 - \ddot{\theta}_1 \cos \theta_1 - \ddot{\theta}_2 \cos \theta_2).
 \end{aligned}$$

Vi vil jo løse dette numerisk, så man kan heller sette det opp som et lineært likningssystem ved å mikse og trikse litt på likningene:

$$\begin{bmatrix} 1 - \frac{m}{M_{tot}} \cos^2 \theta_1 & -\frac{m}{M_{tot}} \cos \theta_1 \cos \theta_2 \\ -\frac{m}{M_{tot}} \cos \theta_1 \cos \theta_2 & 1 - \frac{m}{M_{tot}} \cos^2 \theta_2 \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} = \begin{bmatrix} -\frac{g}{\ell} \sin \theta_1 - \frac{m \cos \theta_1}{M_{tot}} (\dot{\theta}_1^2 \sin \theta_1 + \dot{\theta}_2^2 \sin \theta_2) \\ -\frac{g}{\ell} \sin \theta_2 - \frac{m \cos \theta_2}{M_{tot}} (\dot{\theta}_1^2 \sin \theta_1 + \dot{\theta}_2^2 \sin \theta_2) \end{bmatrix}. \quad (8)$$

Vakkert! Da er dette superenkelt å løse med scipy, som vi gjør i seksjon 2.4

2.4

Da var det å løse systemet. Med scipy er det barnemat:

```

def deriverte(t, U):
    theta1, omega1, theta2, omega2 = U

    #definerer matrisen A
    a11 = L*m*L*(np.cos(theta1)**2)/M_tot
    a12 = -np.cos(theta1)*np.cos(theta2)*m*L/M_tot
    a21 = -np.cos(theta1)*np.cos(theta2)*m*L/M_tot
    a22 = L*m*L*(np.cos(theta2)**2)/M_tot

    A = np.array([[a11,a12],[a21,a22]])

    fellesfaktor = m*L*(np.sin(theta1)*omega1**2+np.sin(theta2)*omega2**2)/M_tot

    #så var den den andre vektoren ja
    b1 = -G*np.sin(theta1)-np.cos(theta1)*fellesfaktor
    b2 = -G*np.sin(theta2)-np.cos(theta2)*fellesfaktor
    B = np.array([b1,b2])

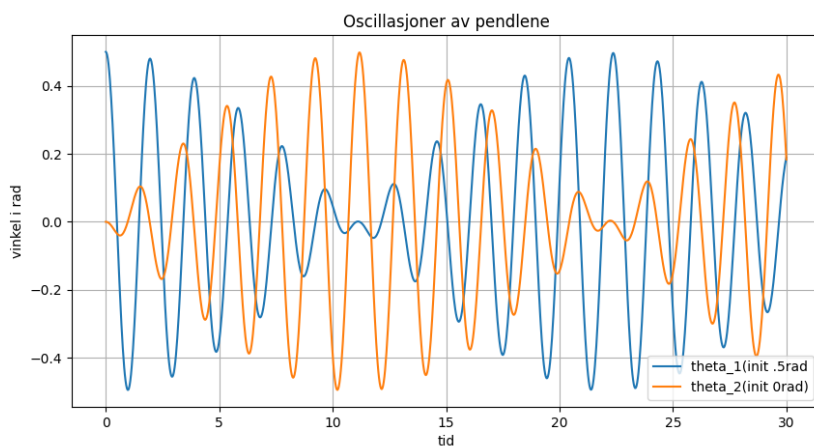
    akselerasjoner = np.linalg.solve(A,B) #løse for akselerasjonene
    return [omega1, akselerasjoner[0],omega2, akselerasjoner[1]]

```

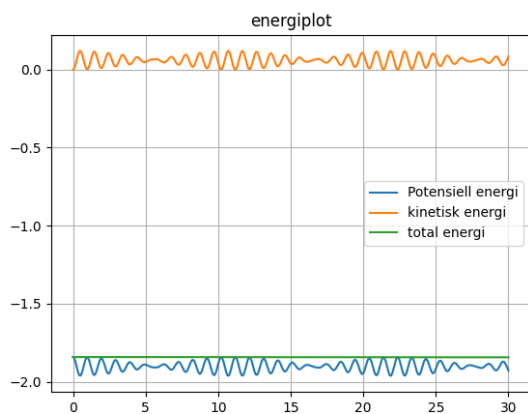
```
#initialbetingelser
U0 = [.5,0,0,0]
t_span = (0,30)
t_eval = np.linspace(0,30,1000)

resultat = solve_ivp(deriverte,t_span, U0,t_eval=t_eval, method = 'RK45')
```

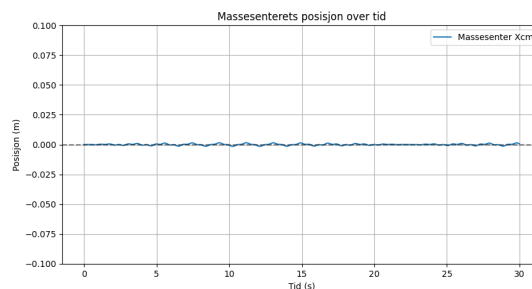
I korte trekk løser koden det koblede systemet ved å transformere to andreordens differensialligninger til et system av fire førsteordens ligninger i tilstandsvektoren \mathbf{U} . Siden vinkelakselerasjonene $\ddot{\theta}_1$ og $\ddot{\theta}_2$ er avhengig gjennom bjelkens bevegelse setter funksjon opp et lineært ligningssystem som i 8, for hvert tidskritt. Akselerasjonene isoleres med `np.linalg.solve`, og deretter integreres det numerisk av `solve_ivp` så vi får simulert bevegelsene til pendelene. For å bekrefte at løsningen



Figur 7: Vinkelutslag på begge pendlene



(a) Energi over tid



(b) Massesenterets posisjon

Figur 8: For å bekrefte at løsningen gir en nøyaktig representasjon av virkeligheten, sjekker vi energiplottet og posisjonen til massesenteret over tid. Siden verdiene følger direkte fra de beregnede posisjonene, kan vi plote disse for å verifisere stabiliteten i modellen.

gir en nøyaktig representasjon av virkeligheten vil vi sjekke energiplottet og posisjonen av massesenteret over tid. Alt vi trenger blir jo regnet ut når vi regner ut posisjonen, så da er det bare å plote verdiene.

Og dette ser jo helt riktig ut. Den totale energien er konstant, og massesenteret er sånn ca. på samme posisjon hele tiden. Litt støy forventer man jo.

Til slutt vil vi beskrive hva som skjer med pendlene over tid. Det vi observerer er en periodisk overføring. Amplituden til θ_1 vil gravis avta mens amplituden til θ_2 vil øke. Dette skjer fordi bjelken beveger seg. Når pendel 1 svinger dytter den bjelken, som dytter pendel 2. Bjelken fungerer som en bro som frakter energi mellom dem. Etter en viss tid vil pendel 2 svinge med maksimal amplitude mens pendel 1 nesten står stille. Så snur det. Hvis vi ser på energiplottet så ser vi at den totale energien er konstant. Derfor må tapet i amplitude hos den ene svares til med en økning hos den andre.

Ved å anta små utslag ($\theta_i \ll 1$) og svak kobling ($m \ll M$) kan systemet beskrives av de lineariserte ligningene $\ddot{\theta}_1 + \omega_0^2 \theta_1 = \frac{m}{M}(\ddot{\theta}_1 + \ddot{\theta}_2)$ og $\ddot{\theta}_2 + \omega_0^2 \theta_2 = \frac{m}{M}(\ddot{\theta}_1 + \ddot{\theta}_2)$. Disse ligningene viser at pendlene fungerer som to koblede oscillatorer som utveksler energi gjennom bjelken. Siden pendlene har nesten like egenfrekvenser, oppstår det et svevningsfenomen der amplituden til den ene pendelen minker mens den andre øker. Dette forklarer hvorfor energien vandrer periodisk frem og tilbake mellom pendlene i simuleringen, samtidig som den totale energien bevares.