FYS1105 - Simon Berg

Oppgave 1:

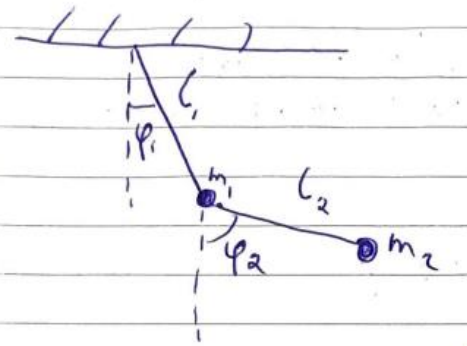
For dette midtveisprosjektet valgt Jeg og Alexander Kristoffersen kanskje det simpleste eksempelet av et kaotisk system, nemlig dobbel pendelen. Selv om det er et relativt simpelt system, ble ikke oppgaven det.

Oppgave 2:

FYS1105 - Alexander Kristoffersen + Simon Berg

Dobbeltpendel analytisk:

Antagelser: $\varphi_1, \varphi_2, \dot\varphi_1, \dot\varphi_2$ små.



polarkoordinater

$$T = T_1 + T_2 = \frac{1}{2}m_1(l_1\dot\varphi_1)^2 + \frac{1}{2}m_2(l_1\vec{\dot\varphi_1} + l_2\vec{\dot\varphi_2})^2$$

$$= \frac{1}{2}m_1(l_1\dot\varphi_1)^2 + \frac{1}{2}m_2(l_1\dot\varphi_1)^2 + \frac{1}{2}m_2(l_2\dot\varphi_2)^2 + \frac{1}{2}m_2(2l_1l_2\dot\varphi_1\dot\varphi_2\cos(\varphi_1-\varphi_2))$$

$$\approx \frac{1}{2}(m_1+m_2)(l_1\dot\varphi_1)^2 + \frac{1}{2}m_2(l_2\dot\varphi_2)^2 \cancel{+0}$$

~~, for små $\dot\varphi_1$ og $\dot\varphi_2$ fordi $\dot\varphi_1 \cdot \dot\varphi_2 \cdot \cos(\varphi_1-\varphi_2)$ blir veldig lite.~~

$$+ m_2 l_1 l_2 \dot\varphi_1 \dot\varphi_2$$ ← egentlig $\boxed{\cos(\varphi_1-\varphi_2)\approx 1}$

siden $\dot\varphi_1$ og $\dot\varphi_2$ er små, vil $\cos(\varphi_1-\varphi_2)$ bli neglisjerbart forskjellig fra 1

$$U = U_1 + U_2 = m_1 g h_1 + m_2 g h_2$$

h_1: l_1 φ_1 $l_1(1-\cos\varphi_1)$

$$= m_1 g l_1(1-\cos\varphi_1) + m_2 g l_1(1-\cos\varphi_1) + m_2 g l_2(1-\cos\varphi_2)$$

$h_2 - h_1$: l_2 φ_2 $l_2(1-\cos\varphi_2)$

$$= (m_1+m_2) g l_1(1-\cos\varphi_1) + m_2 g l_2(1-\cos\varphi_2)$$

Vi får altså:

$$\mathcal{L} = T - U \approx \frac{1}{2}(m_1+m_2)(l_1\dot{\varphi}_1)^2 + \frac{1}{2}m_2(l_2\dot{\varphi}_2)^2 - (m_1+m_2)gl_1(1-\cos\varphi_1) - m_2 g l_2(1-\cos\varphi_2)$$

$$\frac{\partial \mathcal{L}}{\partial \varphi_1} = -(m_1+m_2)gl_1\sin\varphi_1$$

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{\varphi}_1}\right) = (m_1+m_2)l_1^2\ddot{\varphi}_1$$, og siden $\frac{\partial \mathcal{L}}{\partial \varphi_1} = \frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{\varphi}_1}\right)$

$$-(m_1+m_2)gl_1\sin\varphi_1 = (m_1+m_2)l_1^2\ddot{\varphi}_1$$

$$\frac{\partial \mathcal{L}}{\partial \varphi_2} = -m_2 g l_2 \sin\varphi_2$$

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{\varphi}_2}\right) = m_2 l_2^2 \ddot{\varphi}_2$$, så

$$-m_2 g l_2 \sin\varphi_2 = m_2 l_2^2 \ddot{\varphi}_2$$

V3 Vi får altså:

$$\mathcal{L} = T - V = \frac{1}{2}(m_1+m_2)(l_1\dot\varphi_1)^2 + \frac{1}{2}m_2(l_2\dot\varphi_2)^2 + m_2 l_1 l_2 \dot\varphi_1 \dot\varphi_2$$
$$- (m_1+m_2)g l_1(1-\cos\varphi_1) - m_2 g l_2(1-\cos\varphi_2)$$

$$\frac{\partial \mathcal{L}}{\partial \varphi_1} = -(m_1+m_2) g l_1 \sin\varphi_1$$

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot\varphi_1}\right) = (m_1+m_2) l_1^2 \ddot\varphi_1 + m_2 l_1 l_2 \ddot\varphi_2$$

Så: $(m_1+m_2) l_1 \ddot\varphi + m_2 l_1 l_2 \ddot\varphi_2 = -(m_1+m_2) g l_1 \sin\varphi_1$
$\approx -(m_2+m_1) g l_1 \varphi_1$ (små φ_1)

$$\frac{\partial \mathcal{L}}{\partial \varphi_2} = -m_2 g l_2 \sin\varphi_2$$

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot\varphi_2}\right) = m_2 l_2^2 \ddot\varphi_2 + m_2 l_1 l_2 \ddot\varphi_1 \text{, så:}$$

$$m_2 l_2^2 \ddot\varphi_2 + m_2 l_1 l_2 \ddot\varphi_1 = -m_2 g l_2 \sin\varphi_2 \approx -m_2 g l_2 \varphi_2$$

Dette kan skrives som én stor matriseligning!

~~$(m_1+m_2)l_1^2\ddot\varphi + m_2 l_1 l_2$~~

$$\begin{bmatrix} (m_1+m_2)l_1^2 & m_2 l_1 l_2 \\ m_2 l_1 l_2 & m_2 l_2^2 \end{bmatrix} \begin{bmatrix} \ddot\varphi_1 \\ \ddot\varphi_2 \end{bmatrix} = -\begin{bmatrix} +(m_1+m_2) g l_1 & 0 \\ 0 & +m_2 g l_2 \end{bmatrix} \begin{bmatrix} \varphi_1 \\ \varphi_2 \end{bmatrix}$$

dvs. $M\ddot\varphi = -K\varphi$

v4

men, hvis vi antar at $m_1 = m_2 = m$ og

$l_1 = l_2 = l$, og definerer $\omega_0 = \sqrt{\frac{g}{l}}$ får vi

$$M = \begin{bmatrix} 2ml^2 & ml^2 \\ ml^2 & ml^2 \end{bmatrix} = ml^2 \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix}$$

og

$$K = \begin{bmatrix} 2mlg & 0 \\ 0 & mlg \end{bmatrix} = ml^2 \begin{bmatrix} 2\omega_0^2 & 0 \\ 0 & \omega_0^2 \end{bmatrix}.$$

Vi definerer så $\varphi(t) = Re\left\{\begin{bmatrix} a_1 \\ a_2 \end{bmatrix} e^{i\omega t}\right\} = Re\, z(t)$, og får

$-\omega^2 M \vec{a} e^{i\omega t} = -K \vec{a} e^{i\omega t}$, stryker $e^{i\omega t}$ og flytter over!

(1) $[K - \omega^2 M]\vec{a} = 0$. Dette er bare sant så lenge $\det(K - \omega^2 M) = 0$, fordi denne matrisen kan tenkes på som en forstørrelsesfaktor

~~$0 = \det(K - \omega^2 M) = \det \begin{bmatrix} 2ml^2 \cdot 2\omega_0^2 & 0 \\ 0 & ml^2\omega_0^2 - 2ml^2 \end{bmatrix}$~~

$$0 = \det|K - \omega^2 M| = ml^2 \det \begin{bmatrix} 2\omega_0^2 - 2\omega^2 & 0 - \omega^2 \\ 0 - \omega^2 & \omega_0^2 - \omega^2 \end{bmatrix}$$

$$= ml^2 (2\omega_0^2 - 2\omega^2)(\omega_0^2 - \omega^2) - (-\omega^2)(-\omega^2)$$

$\Rightarrow 2(\omega_0^2 - \omega)^2 - \omega^2 = 0$, og abc-formel gir

$\omega_1 = \omega_0(2 + \sqrt{2})$, $\omega_2 = \omega_0^2(2 - \sqrt{2})$

V5

Vi regner så med ω_1 først:

$$(K-\omega^2 M) = ml^2 \begin{bmatrix} 2\omega_0^2 & 0 \\ 0 & \omega_0^2 \end{bmatrix} - \omega_1 ml^2 \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix}$$

$$= ml^2 \begin{bmatrix} 2\omega_0^2 - 2\omega_0^2(2+\sqrt{2}) & -\omega_0(2+\sqrt{2}) \\ -\omega_0(2+\sqrt{2}) & \omega_0^2 - \omega_0^2(2+\sqrt{2}) \end{bmatrix}$$

$$= ml^2 \cdot \omega_0^2 (1+\sqrt{2}) \begin{bmatrix} 2 & \sqrt{2} \\ \sqrt{2} & 1 \end{bmatrix}.$$

regner vi så ut (7) innsatt ω_1 får vi:

$$ml^2 \cdot \omega_0^2 (1+\sqrt{2}) \begin{bmatrix} 2 & \sqrt{2} \\ \sqrt{2} & 1 \end{bmatrix} \cdot \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = 0$$

$$\Rightarrow \begin{bmatrix} 2 & \sqrt{2} \\ \sqrt{2} & 1 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = \vec{0} = \begin{bmatrix} 2a_1 + \sqrt{2} a_2 \\ \sqrt{2} a_1 \ a_2 \end{bmatrix} \Rightarrow a_2 = \sqrt{2} \cdot a_1$$

Vi gjør tilsvarende utregninger for $\omega_2 = \omega_0^2(2-\sqrt{2})$, og får der at $a_1 = -\sqrt{2} a_2$

Det gir:

$$\omega_1 : \varphi(t) = Re\left\{ \begin{bmatrix} a_1 \\ \sqrt{2} a_1 \end{bmatrix} e^{i\omega_1 t} \right\} = Re\left\{ a_1 \begin{bmatrix} 1 \\ \sqrt{2} \end{bmatrix} e^{i\omega_1 t} \right\}$$

$$= a_1 \begin{bmatrix} 1 \\ \sqrt{2} \end{bmatrix} \cos(\omega_1 t \underbrace{- \delta_1})$$

tar hensyn til hvor vi begynner ved $t=0$, altså forskyvningen.

og ω_2: $\varphi(t) = Re\left\{ a_2 \begin{bmatrix} 1 \\ -\sqrt{2} \end{bmatrix} e^{i\omega_2 t} \right\} = a_2 \begin{bmatrix} 1 \\ -\sqrt{2} \end{bmatrix} \cos(\omega_2 t - \delta_2)$

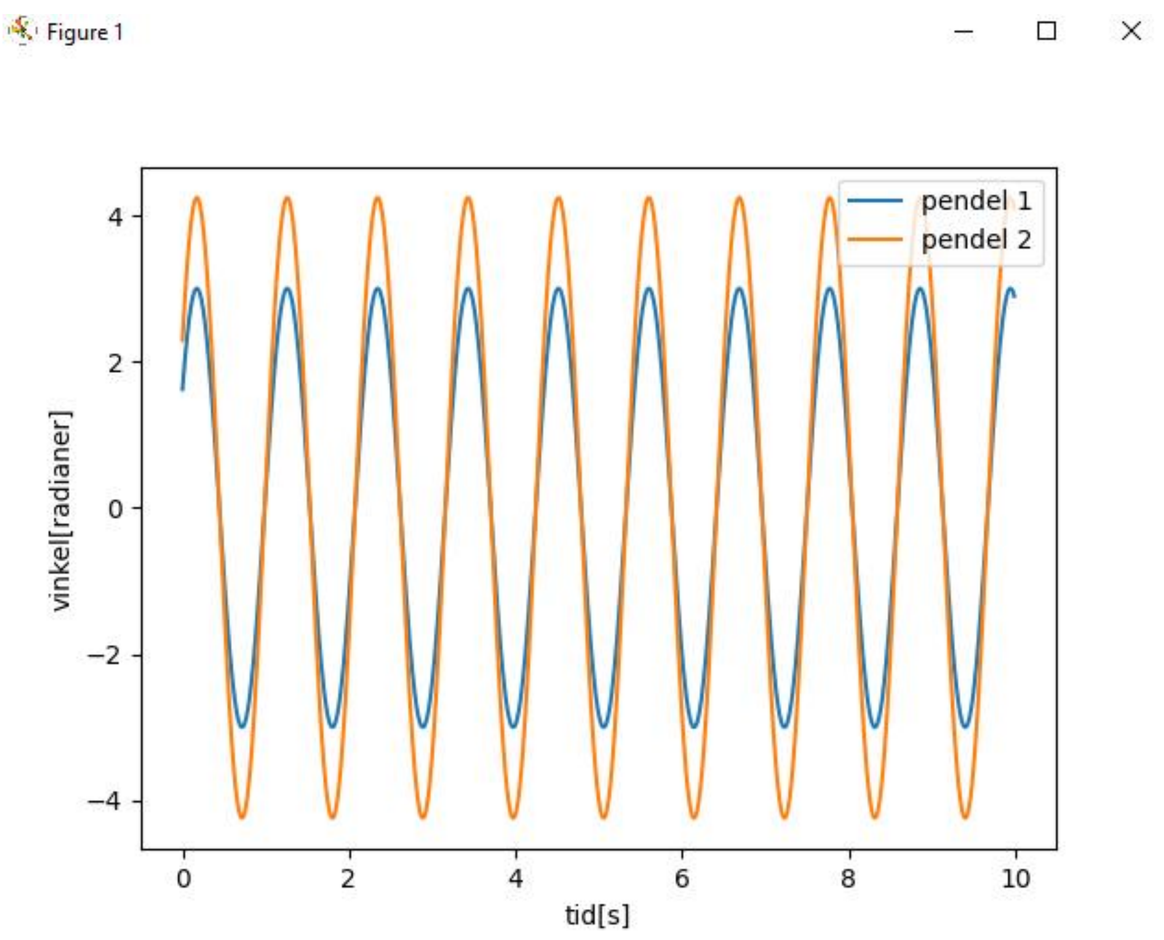
og vi husker på at ω_1 og ω_2 er bestemt av ω_0. a_1, a_2 og δ_1, δ_2 kan bestemmes fritt.

Oppgave 3:

a)

De første 2 plottene er generelle løsninger av dobbel pendelen som vi vet fra før. Der første plot er ved initial vinkel for begge pendelene =pi/2.

Det andre plottet har initial vinkel i motsatt rettning for pendel1 = pi/2 pendel2 = -pi/2



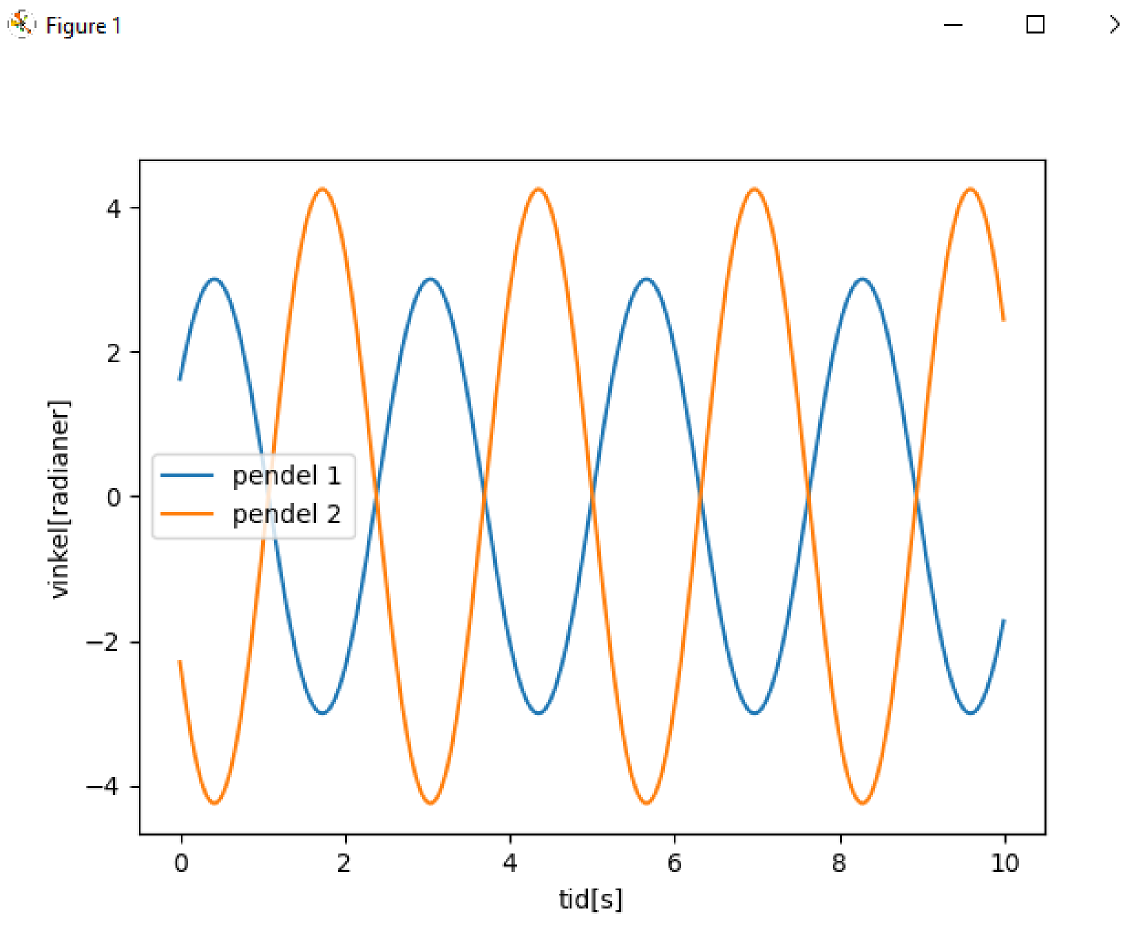


Figure 1
pendel 1
pendel 2
vinkel[radianer]
tid[s]
4
2
0
−2
−4
0
2
4
6
8
10

uke9.py | alexanderKristoffersen_opp3.py | Telemetry Consent

```python
import numpy as np
import matplotlib.pyplot as plt


l = 1
g=9.81


dt = 0.01
tmax = 10
n = int(tmax/dt)

phi1 = np.zeros(n)
phi2 = np.zeros(n)

t = np.array([i*dt for i in range(0,n)])

omega = np.sqrt(g/l) #definert slik
omega1 = omega*np.sqrt(2+np.sqrt(2))#fikk når vi løste
omega2 = omega*np.sqrt(2-np.sqrt(2))


def plot1(a1, delta1):
    phi1_1 = a1*np.cos(omega1*t-delta1)
    phi2_1 = np.sqrt(2)*a1*np.cos(omega1*t-delta1)


    plt.plot(t,phi1_1, label="pendel 1")
    plt.plot(t,phi2_1,label = "pendel 2" )
    plt.ylabel("vinkel[radianer]")
    plt.xlabel("tid[s]")


    plt.legend()
    plt.show()

def plot2(a2, delta2):
```

```python

def plot2(a2,delta2):
    phi1_2 = a2*np.cos(omega2*t-delta2)
    phi2_2 = -np.sqrt(2)*a2*np.cos(omega2*t-delta2)

    plt.plot(t,phi1_2, label="pendel 1")
    plt.ylabel("vinkel[radianer]")
    plt.xlabel("tid[s]")
    plt.plot(t,phi2_2,label = "pendel 2" )


    plt.legend()
    plt.show()

#Tester noen verdier:

plot1(3,1)
plot2(3,1)
```

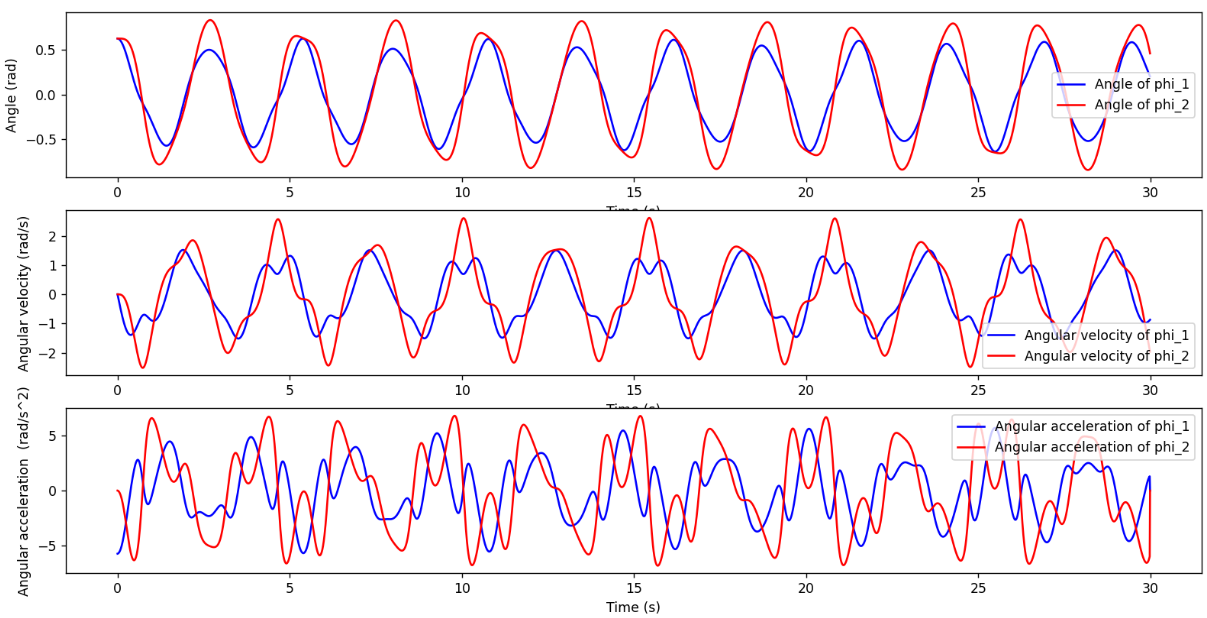
a) Numerisk med tilnærminger:

Bruker vi samme initial betingelser som i den generele løsningen der vi antar initial betingelsene for masse og lengde å være like i de følgende plotttee, men først koden vi har brukt

```
import numpy as np
import numpy.linalg as lin
import matplotlib.pyplot as plt

m1= 1 #Setter massene like...
m2= 1
l1= 1
l2= 1
g= 9.81

phi10= 0.2
phi20= 0.3

dphi10= 0
dphi20= 0

ddphi10= 0
ddphi20= 0

dt= 0.01
tmax= 30
n = int(tmax/dt)

t = np.zeros(n)

phi = np.zeros((n,2,1))
dphi = np.zeros((n,2,1))
ddphi = np.zeros((n,2,1))

phi[0] = np.array([[phi10], [phi20]])
dphi[0] = np.array([[dphi10],[dphi20]])
ddphi[0] = np.array([[ddphi10],[ddphi20]])

M = np.zeros((2,2))
a = np.zeros((2,1))


for i in range(n-1):
```

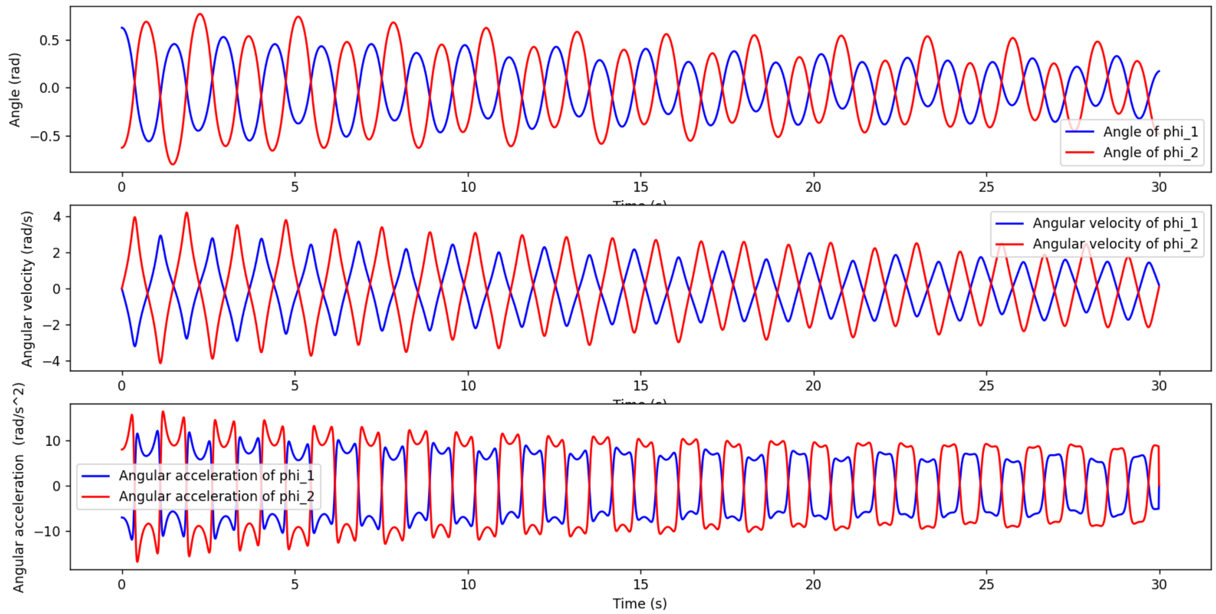
```python
for i in range(n-1):
    phi1i = phi[i][0][0]
    phi2i = phi[i][1][0]
    dphi1i = dphi[i][0][0]
    dphi2i = dphi[i][1][0]
    M[0][0] = (m1+m2)*l1**2
    M[0][1] = m2*l1*l2#*np.cos(phi1i-phi2i)
    M[1][0] = m2*l1*l2#*np.cos(phi1i-phi2i)
    M[1][1] = m2*l1**2

    a[0][0] = m2*l1*l2*dphi2i*np.sin(phi1i-phi2i)*(dphi1i-dphi2i)-m2*l1*l2*dphi1i*dphi2i*np.sin(phi1i-phi2i)-(m1+m2)*g*l1*np.sin(phi1i)
    a[1][0] = m2*l1*l2*dphi1i*np.sin(phi1i-phi2i)*(dphi1i-dphi2i)+m2*l1*l2*dphi1i*dphi2i*np.sin(phi1i-phi2i)-m2*g*l2*np.sin(phi2i)
    ddphi[i] = np.linalg.pinv(M)@a

    dphi[i + 1] = dphi[i] + ddphi[i]*dt
    phi[i + 1] = phi[i] + dphi[i+1]*dt

    t[i + 1] = t[i] + dt

plt.subplot(3,1,1)
plt.plot(t,phi[:,0,0], "b", label="Angle of phi_1")
plt.plot(t,phi[:,1,0], "r", label="Angle of phi_2")
plt.xlabel("Time (s)")
plt.ylabel("Angle (rad)")
plt.legend()
plt.subplot(3,1,2)
plt.plot(t,dphi[:,0,0], "b", label="Angular velocity of phi_1")
plt.plot(t,dphi[:,1,0], "r", label="Angular velocity of phi_2")
plt.xlabel("Time (s)")
plt.ylabel("Angular velocity (rad/s)")
plt.legend()
plt.subplot(3,1,3)
plt.plot(t,ddphi[:,0,0], "b", label="Angular acceleration of phi_1")
plt.plot(t,ddphi[:,1,0], "r", label="Angular acceleration of phi_2")
plt.xlabel("Time (s)")
plt.ylabel("Angular acceleration  (rad/s^2)")

plt.legend()
plt.show()
```

I dette første plotte er phi1=phi2=pi/5



Som vi ser ved å sammenlikne plotet vist tidligere i sub oppgaven ser man at endringen i vinklene er ganske like

I dette plottet er initial vinklene motsatt rettet så phi1 = pi/5 og phi2 = -pi/5



Man ser også her at løsningen for vinklene er ganske like

b) Numerisk løsning uten tillnærminger

Først koden:

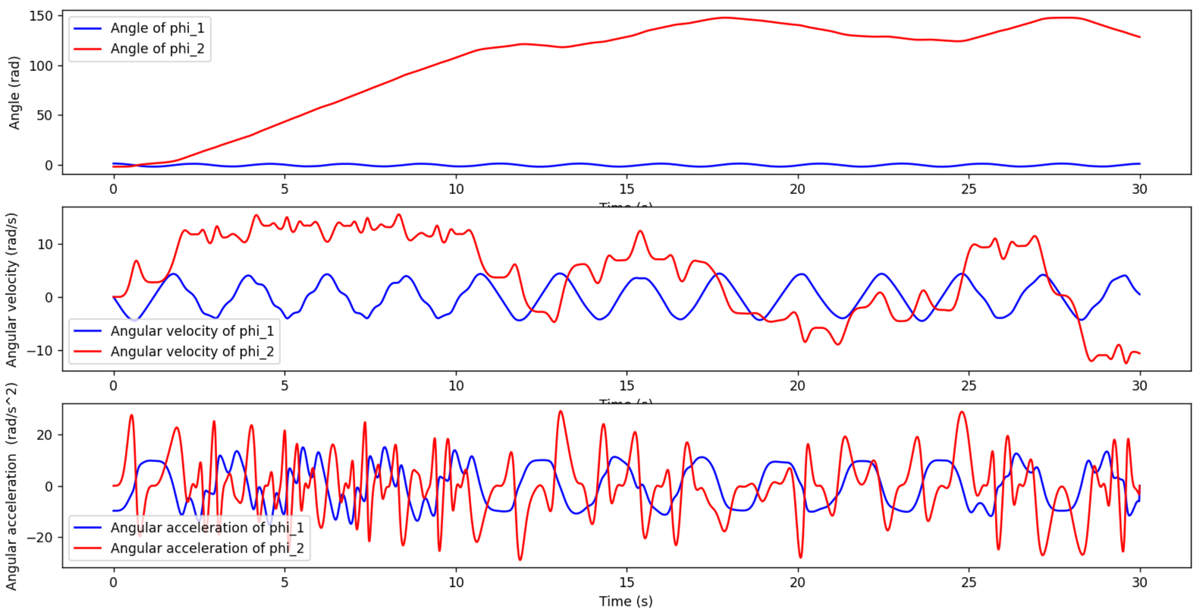
C: > Users > simon > OneDrive > Dokumenter > undervisning > FYS1105 > Prosjekt

```python
import numpy as np
import numpy.linalg as lin
import matplotlib.pyplot as plt

m1= 1
m2= 60
l1= 5
l2= 1
g= 9.81

phi10= np.pi/2
phi20= -np.pi/2

dphi10= 0
dphi20= 0

ddphi10= 0
ddphi20= 0

dt= 0.01
tmax= 40
n = int(tmax/dt)

t = np.zeros(n)

phi = np.zeros((n,2,1))
dphi = np.zeros((n,2,1))
ddphi = np.zeros((n,2,1))

phi[0] = np.array([[phi10], [phi20]])
dphi[0] = np.array([[dphi10],[dphi20]])
ddphi[0] = np.array([[ddphi10],[ddphi20]])

M = np.zeros((2,2))
a = np.zeros((2,1))


```

```python
for i in range(n-1):
    phi1i = phi[i][0][0]
    phi2i = phi[i][1][0]
    dphi1i = dphi[i][0][0]
    dphi2i = dphi[i][1][0]
    M[0][0] = (m1+m2)*l1**2
    M[0][1] = m2*l1*l2*np.cos(phi1i-phi2i)
    M[1][0] = m2*l1*l2*np.cos(phi1i-phi2i)
    M[1][1] = m2*l1**2

    a[0][0] = m2*l1*l2*dphi2i*np.sin(phi1i-phi2i)*(dphi1i-dphi2i)-m2*l1*l2*dphi1i*dphi2i*np.sin(phi1i-phi2
    a[1][0] = m2*l1*l2*dphi1i*np.sin(phi1i-phi2i)*(dphi1i-dphi2i)+m2*l1*l2*dphi1i*dphi2i*np.sin(phi1i-phi2
    ddphi[i] = np.linalg.pinv(M)@a

    dphi[i + 1] = dphi[i] + ddphi[i]*dt
    phi[i + 1] = phi[i] + dphi[i+1]*dt

    t[i + 1] = t[i] + dt

plt.subplot(3,1,1)
plt.plot(t,phi[:,0,0], "b", label="Angle of phi_1")
plt.plot(t,phi[:,1,0], "r", label="Angle of phi_2")
plt.xlabel("Time (s)")
plt.ylabel("Angle (rad)")
plt.legend()
plt.subplot(3,1,2)
plt.plot(t,dphi[:,0,0], "b", label="Angular velocity of phi_1")
plt.plot(t,dphi[:,1,0], "r", label="Angular velocity of phi_2")
plt.xlabel("Time (s)")
plt.ylabel("Angular velocity (rad/s)")
plt.legend()
plt.subplot(3,1,3)
plt.plot(t,ddphi[:,0,0], "b", label="Angular acceleration of phi_1")
plt.plot(t,ddphi[:,1,0], "r", label="Angular acceleration of phi_2")
plt.xlabel("Time (s)")
plt.ylabel("Angular acceleration  (rad/s^2)")
```

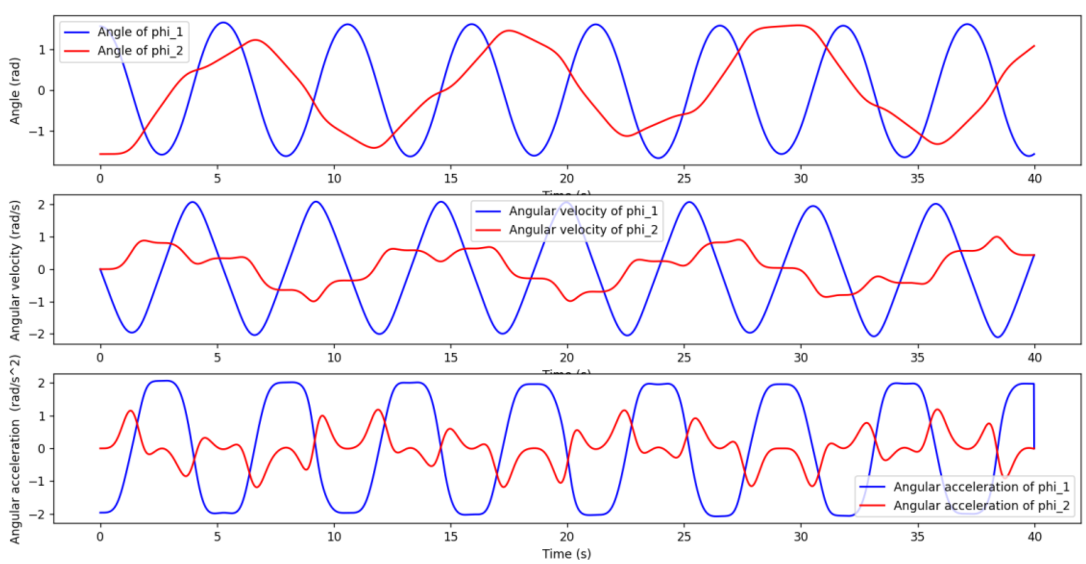
Plt.show()

#Først ser vi på et tilfelle der massen på m_1 er mye større enn m_2



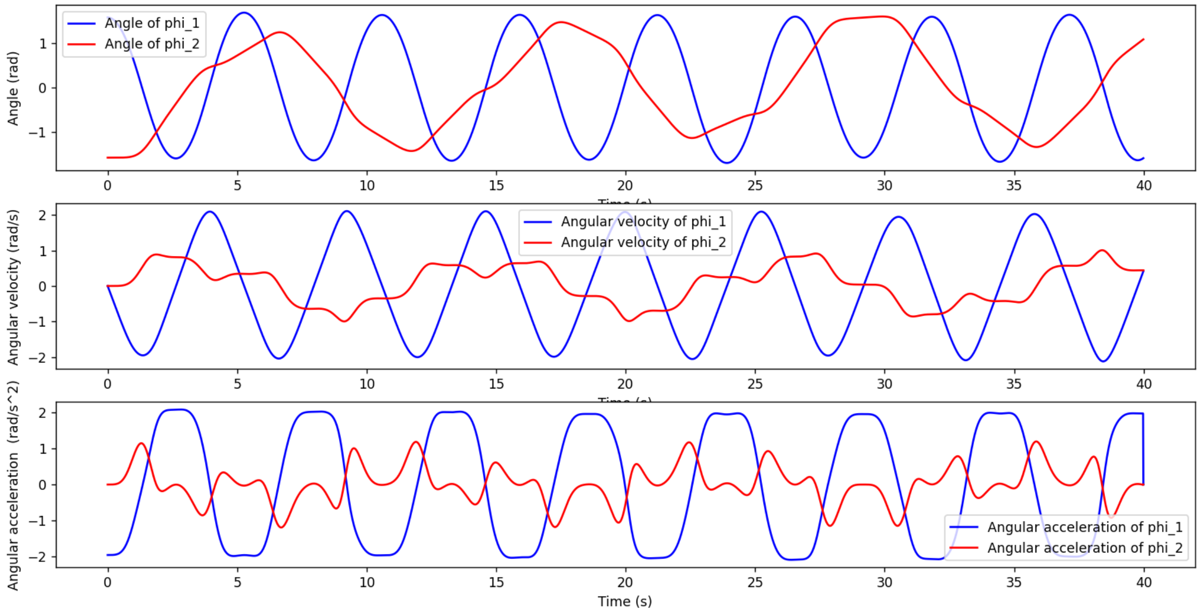
Når m_1>>m_2 så vil den ha få store mengder energi, så de små utfallene i m_2 vil nesten ikke røre m_1, dette vil føre til at m_2 blir slengt rundt. Dette ser vi skje på grafen når vinkelen til phi_2 stiger og går over 2pi, som da betyr at den har gjort en rotasjon.

#Andre tilfelle vi ser på er ved l_1= 1, l_2=5 og m_2=5, m_1=1

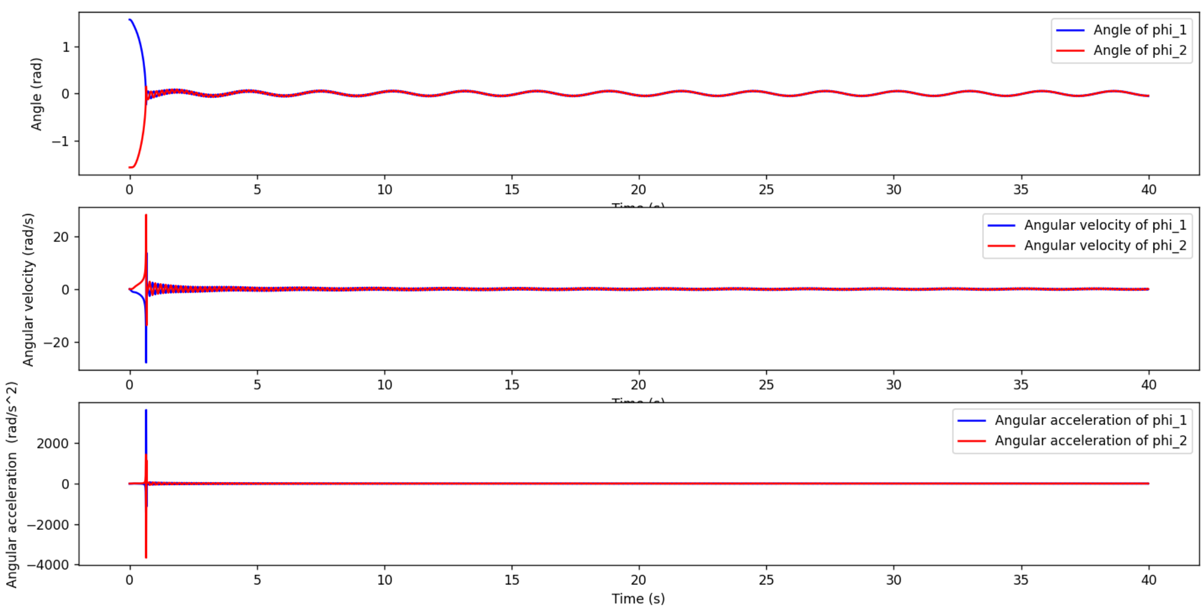


Denne grafen gir ingen mening for meg, og jeg tror det er et problem med koden

#For teste denne teorien gjør jeg at m_2<<m_1 og l_1 = 5. I teorien skal føre til at hele systemet vil nesten fungere som en enkelt pendel.



Koden viser en identisk graf som den tidligere. Jeg sjekker med å beholde massene, men endrer l_1 =1 så l_1=l_2



Der fant vi syndebukken , så da må det være en feil i koden når l_1>1

Konklusjon

Koden fungerte bra for l =1 i hvert fall, og jeg skulle gjerne fikset på koden, men klokken er 22 på søndagskvelden så rekker ikke. Dette prosjektet var SVÆRT interessant og læringsrikt. Skulle ønske jeg fikk mer tid på det så jeg kunne ha valgt et mer avansert system.