Prosjektoppgave besvarelse

Kadnidatnummer: 15267

May 6, 2018

Oppgave 1

Ved å løse differensial likningen

$$m\ddot{x}(t) + kx(t) = 0, \tag{1}$$

og plotte systemets bane i faserommet så får vi plottet i figur 1

Hvis vi varier initialverdiene for posisjonen og hastigheten, så vil det fortsatt være en ellipse. Hvis vi ser på totalenergien til systemet så ser vi hvorfor dette stemmer. Totalenergien er

$$\begin{aligned}
E_{tot} =& E_k + E_p \\
E_{tot} =& \frac{1}{2}m\dot{x}^2 + \frac{1}{2}kx^2 \\
\frac{2E_{tot}}{mk} =& \frac{\dot{x}^2}{k} + \frac{x^2}{m}, \\
Hvis\ vi\ setter\ C =& \frac{2E_{tot}}{mk} \\
\Rightarrow \frac{x^2}{m} + \frac{\dot{x}^2}{k} =& C \\
\frac{x^2}{mC} + \frac{\dot{x}^2}{kC} =& C \\
\frac{x^2}{\sqrt{mC}^2} + \frac{\dot{x}^2}{\sqrt{kC}^2} =& 1
\end{aligned}$$

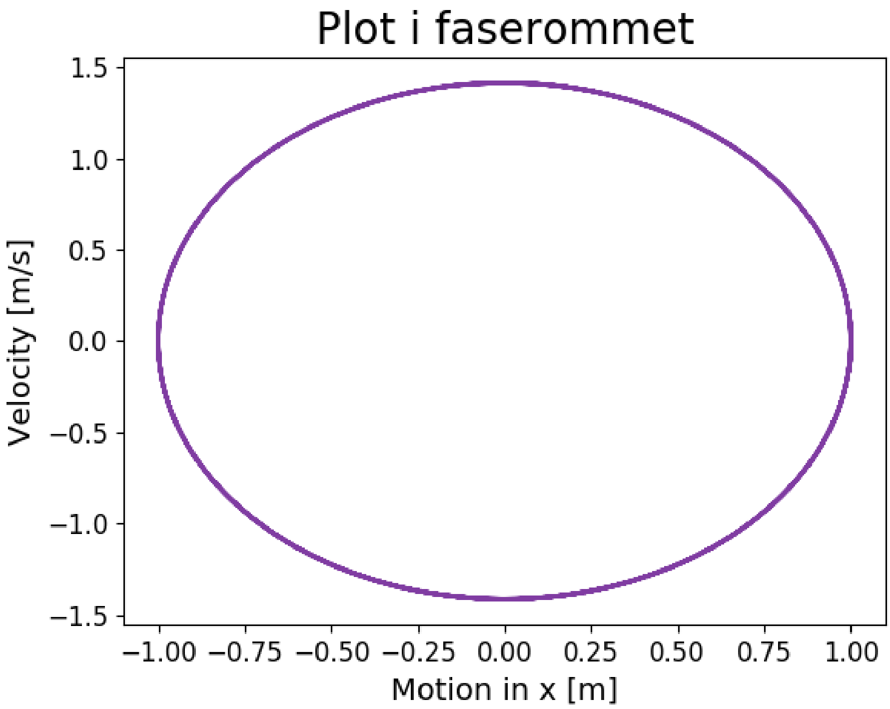

Figure 1: viser faseplottet for en fri harmonisk oscillator

Det er formelen for en ellipse.

Der hvor banen krysser x aksen og y aksen så er den vinkelrett på den aksen den krysser. Dette skyldes at i de punktene så er totalenergien lokalisert i bare det ene uttrykket, og det andre er null. Ved $|x| = max$ så endrer bevegelsen retning og $E_{tot} = E_p$. Når $x = 0$ så er $E_{tot} = E_k$

Hvis vi plotter en kortere tidsperiode slik at kurven ikke når til startposisjonen så ser vi at den beveger seg med klokken. Selv om vi varierer startposisjon og starthastighet så er det ikke mulig å få banen til å gå i motsatt retning på grunn av hvordan vi har satt positive retninger og hva vi har satt på x og y aksen.

Oppgave 2

Hvis vi inkluderer dempning i programmet så vil banen i faseplottet gå gradvis mot punktet $x = 0m$ og $\dot{x} = 0m/s$ som en spiral.

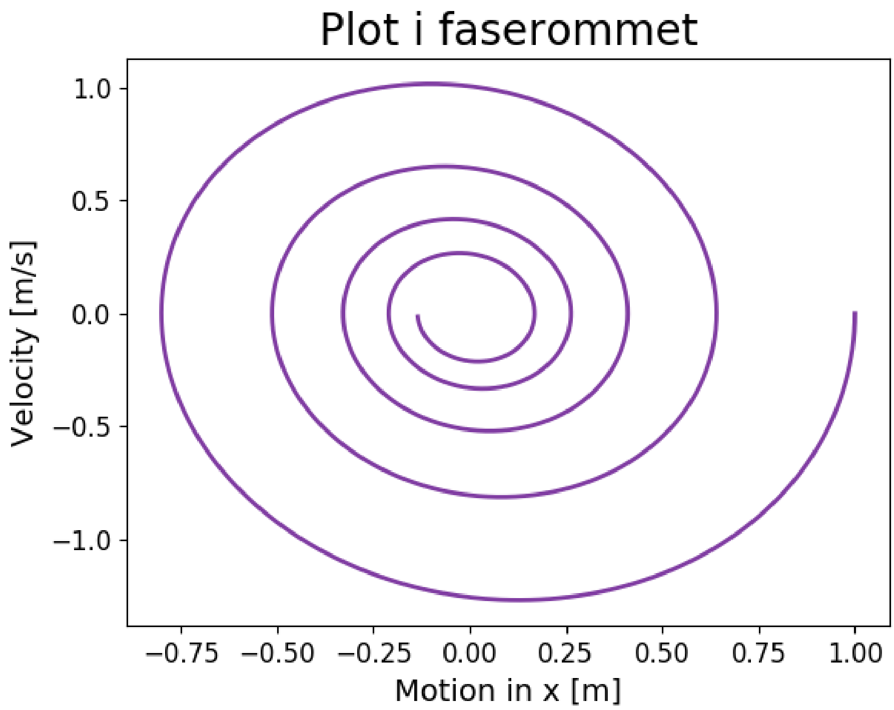

Figure 2:

Banen står vinkelrett bare når den krysser x aksen, ikk y aksen. Dette er på grunn av dempings leddet som vi satt inn.

En attraktor er et set med numeriske verdier som et dynamisk system har en tendens til å utvikle seg mot for et bredt utvalg av initialbetingelser. En attraktor kan være et punkt(0 dim), et set met punkter, en kurve(2 dim), og flere ting innenfor matematikk. Punktet $x = 0m$ og $\dot{x} = 0m/s$ er en attraktor av dimensjon 0, fordi det er et singulært punkt. Kurven i oppgave 1 er også en attraktor i følge definisjonen tror jeg, siden systemet alltid havner i den kurven uansett initialbetingelser. På den andre siden så kan det argumenteres for at det ikke er en attraktor fordi systemet faktisk ikke kan gjøre noe annet, og dermed ikke kan ha en tendens til å ende langs kurven. Altså siden den ikke begynner utenfor kurven og så går inn og blir der så er det ikke en attraktor. FINN UT AV DET.

Oppgave 3

Vi skal løse følgende likning alanytisk.

$$m\ddot{x}(t) + kx(t) = F_D cos(\omega_D t) \tag{2}$$

Begynner med å finne den generelle løsningen for den homogene likningen, og løser derfor den karakteristiske likningen $m\lambda^2 + k = 0$

$$\begin{aligned} m\lambda^2 + k &= 0 \\ \lambda^2 &= \frac{-k}{m} \\ \lambda &= \pm\sqrt{\frac{-k}{m}} \end{aligned}$$

Hvis vi har λ på formen $\lambda_\pm = \mu \pm i\eta$, så får vi den generelle løsningen

$$\begin{aligned} \lambda &= \pm i\sqrt{\frac{k}{m}} \\ x_h &= e^{\mu t}(Acos(\eta t) + Bsin(\eta t)) \\ x_h &= e^{0t}\left(Acos\left(\sqrt{\frac{k}{m}}t\right) + Bsin\left(\sqrt{\frac{k}{m}}t\right)\right) \\ x_h &= \underline{Acos\left(\sqrt{\frac{k}{m}}t\right) + Bsin\left(\sqrt{\frac{k}{m}}t\right)} \end{aligned}$$

Så den generelle løsningen for den homogene delen er

$$x_h = Acos\left(\sqrt{\frac{k}{m}}t\right) + Bsin\left(\sqrt{\frac{k}{m}}t\right) \tag{3}$$

Nå skal vi bruke metoden for ubestemte koeffisienter for å finne den partikulære løsningen. Siden høyre side av likningen er på formen $Acos(pt)$, så kan vi bruke x_p på formen Ce^{ipt}

$$\begin{aligned}
x_p =& Ce^{ipt}, \quad p = \omega_D, fra \;\; likningen \\
x_p =& Ce^{i\omega_D t} \\
\dot{x_p} =& iC\omega_D e^{i\omega_D t} \\
\ddot{x_p} =& -C\omega_D^2 e^{i\omega_D t}
\end{aligned}$$

Dette setter vi inn i difflikningen og løser for C

$$\begin{aligned}
m\ddot{x} + kx = F_D cos(\omega_D t) \\
m\left(-C\omega_D^2 e^{i\omega_D t}\right) + kCe^{i\omega_D t} =& F_D cos(\omega_D t) \\
-mC\omega_D^2 e^{i\omega_D t} + kCe^{i\omega_D t} =& F_D cos(\omega_D t) \\
Ce^{i\omega_D t}\left(-m\omega_D^2 + k\right) =& F_D cos(\omega_D t) \\
C =& \frac{F_D cos(\omega_D t)}{e^{i\omega_D t}\left(-m\omega_D^2 + k\right)}
\end{aligned}$$

Så setter vi uttrykket for C inn i x_p og får at

$$x_p = \frac{F_D cos(\omega_D t)}{(k - m\omega_D^2)} \tag{4}$$

Så den fullstendige generelle løsningen er

$$Acos\left(\sqrt{\frac{k}{m}}t\right) + Bsin\left(\sqrt{\frac{k}{m}}t\right) + \frac{F_D cos(\omega_D t)}{(k - m\omega_D^2)} \tag{5}$$

Vi skulle løse den for $x(0) = 2m$ og $\dot{x}(0) = 0m/s$, så da gjør vi det og får

$$\begin{aligned}
A =& 2 - \frac{F_D}{k - m\omega_D^2} \\
B =& 0
\end{aligned}$$

Dette setter vi inn i løsningen og får

$$x(t) = 2cos\left(\sqrt{\frac{k}{m}}t\right) + \frac{F_D}{k - m\omega_D^2}\left(cos(\omega_D t) - cos\left(\sqrt{\frac{k}{m}}t\right)\right) \tag{6}$$

Så bruker vi en trigonometrisk identitet funnet på side 42 i Rottmann, som ser slik ut

$$cos\alpha - cos\beta = -2sin\frac{\alpha + \beta}{2}sin\frac{\alpha - \beta}{2} \tag{7}$$

og da får vi

$$x(t) = 2cos\left(\sqrt{\frac{k}{m}}t\right) - \frac{2F_D}{k - m\omega_D^2}\left(sin\frac{\left(\omega_D + \sqrt{\frac{k}{m}}\right)t}{2}sin\frac{\left(\omega_D - \sqrt{\frac{k}{m}}\right)t}{2}\right) \tag{8}$$

Hvordan kan vi tolke en slik løsning?

Oppgave 4

Ved å løse likning 2 nummerisk og plotte den nummeriske løsningen minus den analytiske så får vi.
Så det nummeriske resultatet er ganske likt det analytiske. Plotter vi bevegelsen i faserommet får vi figur **??**
Bevegelsen i figur **??** er periodisk. Hvis vi bytter ut drivefrekvensen med $\omega_D = \frac{2}{(\sqrt{5}-1)\omega_0}$ så får vi
Bevegelsen i figur **??** også en pariodisk bevegelse.

Oppgave 5

mer ting
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla sed massa urna. Fusce placerat, nulla id aliquam varius, nulla risus commodo metus, in fringilla velit massa eu odio. Aliquam vitae eros at orci volutpat pulvinar eu a sapien. Proin ullamcorper tincidunt orci vel vehicula. Vestibulum venenatis eget magna at pharetra. Aenean ut rutrum urna. Aliquam

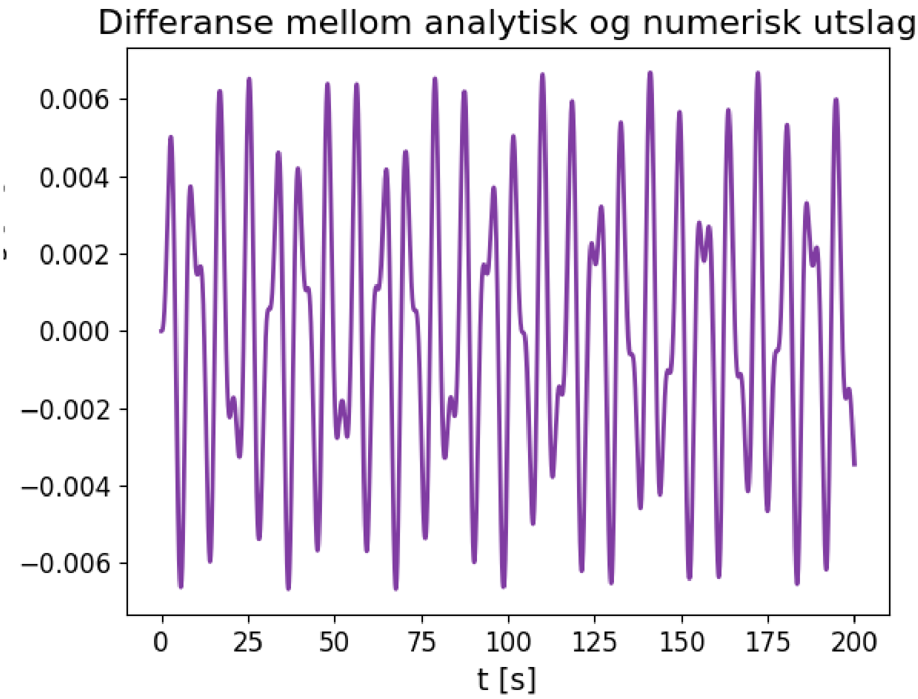

Figure 3: viser differansen mellom den nummeriske og analytiske løsningen

venenatis leo viverra, fermentum massa vel, maximus lectus. Pellentesque pulvinar sodales massa non venenatis. Praesent lobortis consequat erat ut semper. Nulla tincidunt ac enim sed imperdiet. Nulla maximus dui eget eros laoreet, vitae eleifend magna vulputate. Sed ut molestie neque. Sed sagittis sagittis metus. Fusce placerat enim ut augue ornare mattis.
ting

Oppgave 6

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla sed massa urna. Fusce placerat, nulla id aliquam varius, nulla risus commodo metus, in fringilla velit massa eu odio. Aliquam vitae eros at orci volutpat pulvinar eu a sapien. Proin ullamcorper tincidunt orci vel vehicula. Vestibulum venenatis eget magna at pharetra. Aenean ut rutrum urna. Aliquam venenatis leo viverra, fermentum massa vel, maximus lectus. Pellentesque

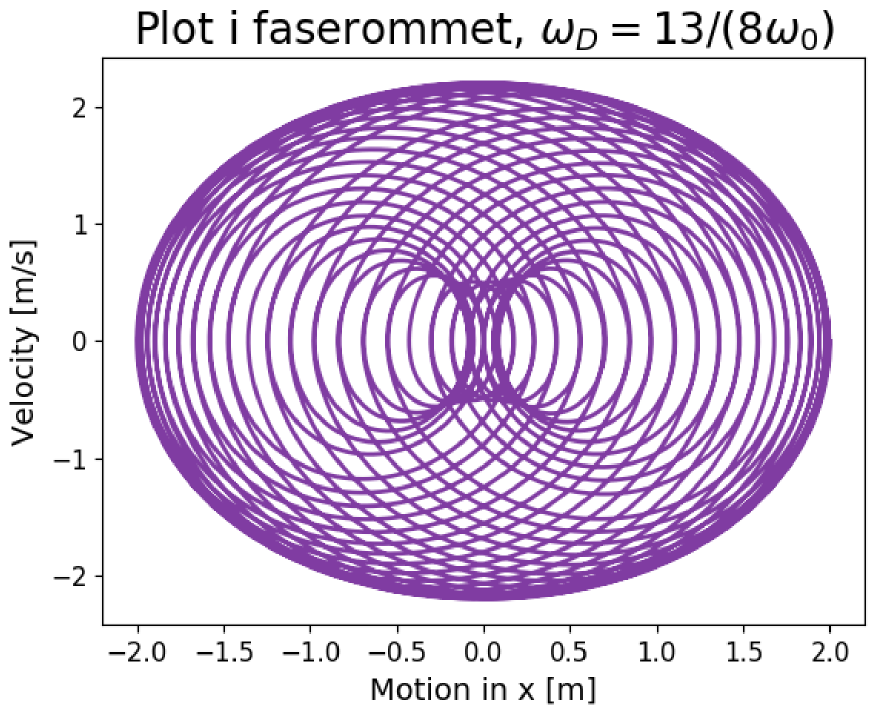

Figure 4: Viser faseplottet med påtrykt kraft og $\omega_D = \frac{13}{8\omega_0}$

pulvinar sodales massa non venenatis. Praesent lobortis consequat erat ut semper. Nulla tincidunt ac enim sed imperdiet. Nulla maximus dui eget eros laoreet, vitae eleifend magna vulputate. Sed ut molestie neque. Sed sagittis sagittis metus. Fusce placerat enim ut augue ornare mattis.

Oppgave 7

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla sed massa urna. Fusce placerat, nulla id aliquam varius, nulla risus commodo metus, in fringilla velit massa eu odio. Aliquam vitae eros at orci volutpat pulvinar eu a sapien. Proin ullamcorper tincidunt orci vel vehicula. Vestibulum venenatis eget magna at pharetra. Aenean ut rutrum urna. Aliquam venenatis leo viverra, fermentum massa vel, maximus lectus. Pellentesque pulvinar sodales massa non venenatis. Praesent lobortis consequat erat ut

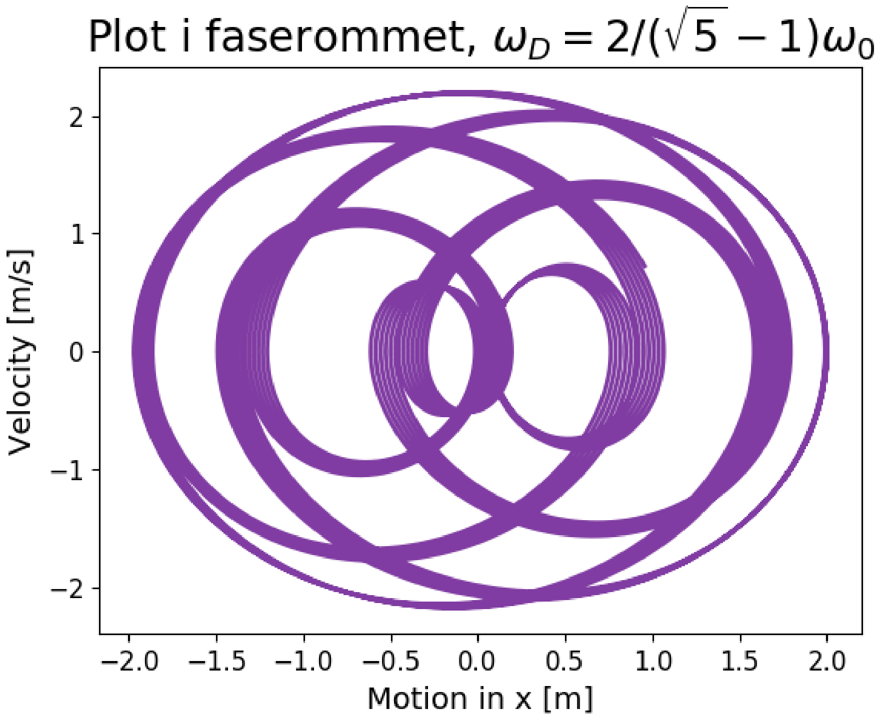

Figure 5: Viser faseplottet med påtrykt kraft og $\omega_D = \frac{2}{(\sqrt{5}-1)\omega_0}$

semper. Nulla tincidunt ac enim sed imperdiet. Nulla maximus dui eget eros laoreet, vitae eleifend magna vulputate. Sed ut molestie neque. Sed sagittis sagittis metus. Fusce placerat enim ut augue ornare mattis.

Oppgave 8

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla sed massa urna. Fusce placerat, nulla id aliquam varius, nulla risus commodo metus, in fringilla velit massa eu odio. Aliquam vitae eros at orci volutpat pulvinar eu a sapien. Proin ullamcorper tincidunt orci vel vehicula. Vestibulum venenatis eget magna at pharetra. Aenean ut rutrum urna. Aliquam venenatis leo viverra, fermentum massa vel, maximus lectus. Pellentesque pulvinar sodales massa non venenatis. Praesent lobortis consequat erat ut semper. Nulla tincidunt ac enim sed imperdiet. Nulla maximus dui eget

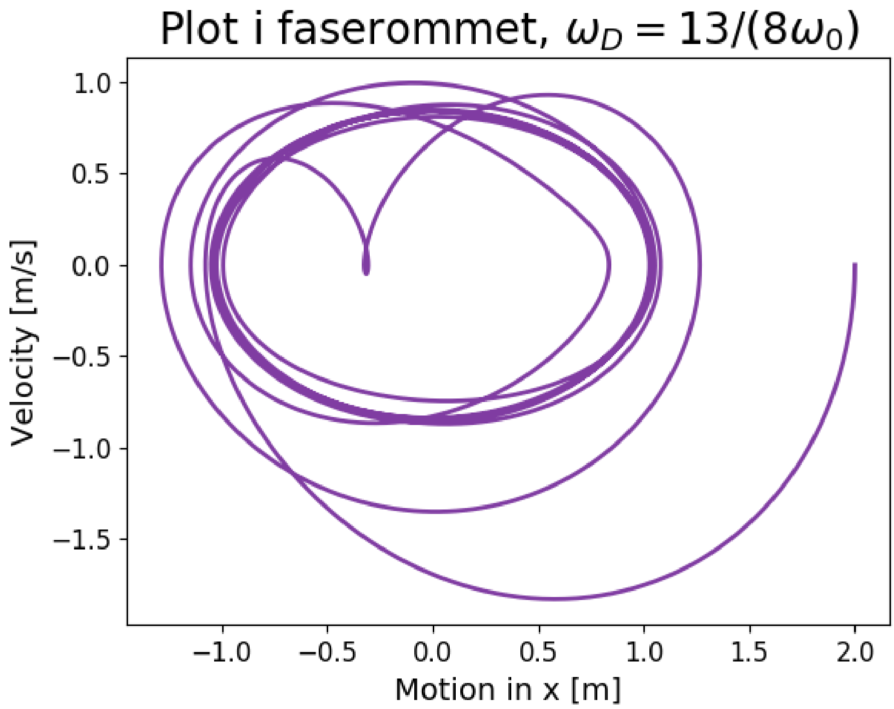

Figure 6: noe

eros laoreet, vitae eleifend magna vulputate. Sed ut molestie neque. Sed sagittis sagittis metus. Fusce placerat enim ut augue ornare mattis.

Oppgave 9

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla sed massa urna. Fusce placerat, nulla id aliquam varius, nulla risus commodo metus, in fringilla velit massa eu odio. Aliquam vitae eros at orci volutpat pulvinar eu a sapien. Proin ullamcorper tincidunt orci vel vehicula. Vestibulum venenatis eget magna at pharetra. Aenean ut rutrum urna. Aliquam venenatis leo viverra, fermentum massa vel, maximus lectus. Pellentesque pulvinar sodales massa non venenatis. Praesent lobortis consequat erat ut semper. Nulla tincidunt ac enim sed imperdiet. Nulla maximus dui eget eros laoreet, vitae eleifend magna vulputate. Sed ut molestie neque. Sed

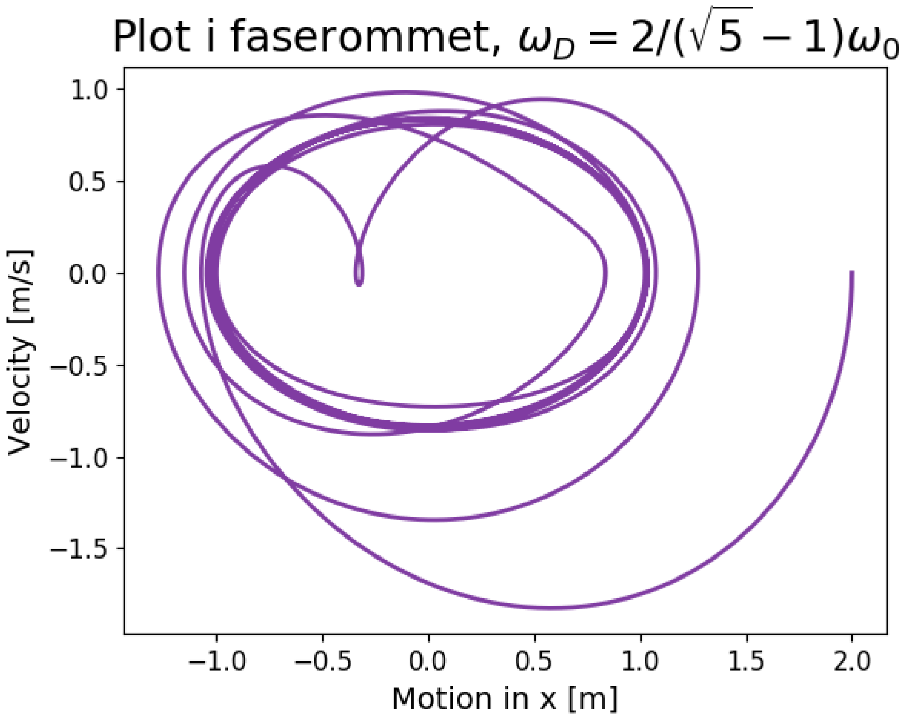

Figure 7: noe

sagittis sagittis metus. Fusce placerat enim ut augue ornare mattis.

Appendix

RK4.py

```
"""
The function for the Runge Kutta method

"""

# The Runge Kutta method
def RK4(diffEQ,xStart,vStart,tStart,dt):
    a1 = diffEQ(xStart,vStart,tStart)
    v1 = vStart

    xHalf1 = xStart + v1 * dt/2.0
```

```
    vHalf1 = vStart + a1 * dt/2.0

    a2 = diffEQ(xHalf1,vHalf1,tStart+dt/2.0)
    v2 = vHalf1

    xHalf2 = xStart + v2 * dt/2.0
    vHalf2 = vStart + a2 * dt/2.0

    a3 = diffEQ(xHalf2,vHalf2,tStart+dt)
    v3 = vHalf2

    xEnd = xStart + v3 * dt
    vEnd = vStart + a3 * dt

    a4 = diffEQ(xEnd,vEnd,tStart+dt)
    v4 = vEnd

    aMiddle = 1.0/6.0 * (a1+2*a2+2*a3+a4)
    vMiddle = 1.0/6.0 * (v1+2*v2+2*v3+v4)

    xEnd = xStart + vMiddle*dt
    vEnd = vStart + aMiddle*dt

    return xEnd, vEnd
```

oppgave1.py

```
#imports
import numpy as np
import matplotlib.pyplot as plt
from RK4 import RK4
import seaborn


#declare constants
dt = 10.**-2        #some timestep [s]
m = 0.500       #mass of 500 g [kg]
k = 1.          #stiffness constant [N/m]
x0 = 1.             #start position [m]
v0 = 0.             #start velocity [m/s]
T = 20.0            #total time [s]
N = int(T/dt)

#setting initialconditions
x = np.zeros(N); v = np.zeros(N); t = np.zeros(N)
x[0] = x0; v[0] = v0

"""
the function we have is
```

```
mx''(t) + kx(t) = 0
x''(t) = kx(t)/m
"""
#making functions

def diffEQ(xNow,vNow,tNow):
    aNow = -k*xNow/m
    return aNow

#running the loop
for i in range(N-1):
    x1 = x[i]; v1 =v[i]
    x[i+1],v[i+1] = RK4(diffEQ,x1,v1,t,dt)
    t[i+1] = t[i] + dt



#plotting

"""
#plotting the motion in x against time
plt.plot(t,x)
plt.xlabel("time")
plt.ylabel("motion")
plt.show()
"""

#plotting the phaseplot
plt.plot(x,v, '#803CA2', linewidth=2.0)
plt.title('Plot i faserommet', fontsize=20)
plt.xlabel("Motion in x [m]", fontsize=14)
plt.ylabel("Velocity [m/s]", fontsize=14)
plt.tick_params(axis = 'both', which = 'major', labelsize =
    12)
#plt.savefig('Oppgave1.png')
plt.show()
```

oppgave2.py

```
#imports
import numpy as np
import matplotlib.pyplot as plt
from RK4 import RK4
import seaborn



#declare constants
dt = 10.**-2        #some timestep [s]
```

```python
m = 0.500              #mass of 500 g [kg]
k = 1.                 #stiffness constant [N/m]
x0 = 1.                #start position [m]
v0 = 0.                #start velocity [m/s]
T = 20.                #total time [s]
N = int(T/dt)          #number of things
b = 0.1                #dampening constant[kg/s]


#setting initialconditions
x = np.zeros(N); v = np.zeros(N); t = np.zeros(N)
x[0] = x0; v[0] = v0

"""
the function we have is
mx''(t) +bx'(t) + kx(t) = 0
x''(t) = -bx'(t)/m -kx(t)/m
"""
#making functions

def diffEQ(xNow,vNow,tNow):
    aNow = -b*vNow/m-k*xNow/m
    return aNow

#running the loop
for i in range(N-1):
    x1 = x[i]; v1 =v[i]
    x[i+1],v[i+1] = RK4(diffEQ,x1,v1,t,dt)
    t[i+1] = t[i] + dt



#plotting

#plotting the motion in x against time
plt.plot(t,x, '#803CA2', linewidth=2.0)
plt.title('Plot av utslag', fontsize=20)
plt.xlabel("Time [s]", fontsize=14)
plt.ylabel("Motion in x [m]", fontsize=14)
plt.tick_params(axis = 'both', which = 'major', labelsize =
    12)
plt.show()

#plotting the phaseplot
plt.plot(x,v, '#803CA2', linewidth=2.0)
plt.title('Plot i faserommet', fontsize=20)
plt.xlabel("Motion in x [m]", fontsize=14)
plt.ylabel("Velocity [m/s]", fontsize=14)
plt.tick_params(axis = 'both', which = 'major', labelsize =
```

```
    12)
#plt.savefig('Oppgave2.png')
plt.show()
```

oppgave4.py

```
#imports
import numpy as np
import matplotlib.pyplot as plt
from RK4 import RK4
import seaborn



#declare constants
dt = 10.**-2          #some timestep [s]
m = 0.500             #mass of 500 g [kg]
k = 1.                #stiffness constant [N/m]
x0 = 2.               #start position [m]
v0 = 0.               #start velocity [m/s]
T = 200.              #total time [s]
N = int(T/dt)         #number of things
omega0 = k/m          #Svingefrekvens for HO
F_D = 0.7             #[N]
omega_D = 13./(8*omega0)
#omega_D = 2./((np.sqrt(5)-1)*omega0)


#setting initialconditions
x = np.zeros(N); v = np.zeros(N); t = np.zeros(N)
x[0] = x0; v[0] = v0
F = np.zeros(N)
F[0] = F_D



"""
the function we have is
mx''(t) + kx(t) = F(t)
x''(t) = F(t)/m -kx(t)/m
"""
#making functions

def diffEQ(xNow,vNow,tNow):
    aNow = (F[i]-k*xNow)/m
    return aNow

#running the loop
for i in range(N-1):
```

```
    x1 = x[i]; v1 =v[i]
    F[i] = F_D *np.cos(omega_D*t[i])
    x[i+1],v[i+1] = RK4(diffEQ,x1,v1,t,dt)
    t[i+1] = t[i] + dt


#The analytic solution

def anal_x_solution(t):
    return 2*np.cos(np.sqrt(k/m)*t) + F_D/(k-m*omega_D**2)*(
       np.cos(omega_D*t)-np.cos(np.sqrt(k/m)*t))


#plotting

#plotting the motion in x against time
plt.plot(t,x, '#803CA2', linewidth=2.0)
plt.title('Plot av utslag', fontsize=20)
plt.xlabel("Time [s]", fontsize=14)
plt.ylabel("Motion in x [m]", fontsize=14)
plt.tick_params(axis = 'both', which = 'major', labelsize =
   12)
plt.show()

#plotting the phaseplot
plt.plot(x,v, '#803CA2', linewidth=2.0)
plt.title('Plot i faserommet, $\omega_D = 13/(8\omega_0)$',
   fontsize=20)
plt.xlabel("Motion in x [m]", fontsize=14)
plt.ylabel("Velocity [m/s]", fontsize=14)
plt.tick_params(axis = 'both', which = 'major', labelsize =
   12)
#plt.savefig('Oppgave4del1.png')
plt.show()


#plotting the difference between the numerical and analytical
    result

plt.plot(t, x-anal_x_solution(t), '#803CA2', linewidth=2.0)
plt.xlabel('t [s]', fontsize=14)
plt.ylabel('Differanse utslag [m]',fontsize=14)
plt.title('Differanse mellom analytisk og numerisk utslag',
   fontsize=16)
plt.tick_params(axis = 'both', which = 'major', labelsize =
   12)
#plt.savefig('Oppgave4_differanse.png')
plt.show()
```

oppgave5.py

```
#imports
import numpy as np
import matplotlib.pyplot as plt
from RK4 import RK4
import seaborn


#declare constants
dt = 10.**-2            #some timestep [s]
m = 0.500               #mass of 500 g [kg]
k = 1.                  #stiffness constant [N/m]
x0 = 2.                 #start position [m]
v0 = 0.                 #start velocity [m/s]
T = 100.                #total time [s]
N = int(T/dt)           #number of things
b = 0.1                 #[kg/s]
omega0 = k/m            #Svingefrekvens for HO
F_D = 0.7               #[N]
omega_D = 13./(8*omega0)
#omega_D = 2./((np.sqrt(5)-1)*omega0)


#setting initialconditions
x = np.zeros(N); v = np.zeros(N); t = np.zeros(N)
x[0] = x0; v[0] = v0
F = np.zeros(N)
F[0] = F_D



"""
the function we have is
mx''(t) +bx'(t) + kx(t) = F(t)
x''(t) = (F(t) -bx'(t) -kx(t))/m
"""
#making functions

def diffEQ(xNow,vNow,tNow):
    aNow = (F[i]-b*vNow-k*xNow)/m
    return aNow

#running the loop
for i in range(N-1):
    x1 = x[i]; v1 =v[i]
    F[i] = F_D *np.cos(omega_D*t[i])
    x[i+1],v[i+1] = RK4(diffEQ,x1,v1,t,dt)
    t[i+1] = t[i] + dt
```

```
#plotting

#plotting the motion in x against time
plt.plot(t,x, '#803CA2', linewidth=2.0)
plt.title('Plot av utslag', fontsize=20)
plt.xlabel("Time [s]", fontsize=14)
plt.ylabel("Motion in x [m]", fontsize=14)
plt.tick_params(axis = 'both', which = 'major', labelsize =
    12)
plt.show()

#plotting the phaseplot
plt.plot(x,v, '#803CA2', linewidth=2.0)
plt.title('Plot i faserommet, $\omega_D = 13/(8\omega_0)$',
    fontsize=20)
plt.xlabel("Motion in x [m]", fontsize=14)
plt.ylabel("Velocity [m/s]", fontsize=14)
plt.tick_params(axis = 'both', which = 'major', labelsize =
    12)
#plt.savefig('Oppgave5del1.png')
plt.show()
```

oppgave6.py

```
#imports
import numpy as np
import matplotlib.pyplot as plt
from RK4 import RK4
import seaborn


#declare constants
dt = 10.**-4          #some timestep [s]
m0 = 0.00001          #mass of waterdrop [kg]
k = 0.475             #stiffness constant [N/m]
x0 = 0.001            #start position [m]
v0 = 0.001            #start velocity [m/s]
T = 3.            #total time [s]
N = int(T/dt)         #number of things
b = 0.001             #[kg/s]
#omega0 = k/m         #Svingefrekvens for HO
#F_D = 0.7            #[N]
#omega_D = 13./(8*omega0)
#omega_D = 2./((np.sqrt(5)-1)*omega0)
g = 9.81              #gravitational acceleration [m/s^2]
psi = 0.00055         #m'(t) [kg/s]
dpsi =3*psi/N
```

```python
#setting initialconditions
x = np.zeros(N); v = np.zeros(N); t = np.zeros(N)
x[0] = x0; v[0] = v0
m = np.zeros(N)
m[0] = m0

#F = np.zeros(N)
#F[0] = F_D



"""
the functions we have are
m(t)x''(t)+(b+psi)x'(t)+kx(t)=m(t)g
x''(t) = (m(t)g - (b+psi)x'(t) - kx(t))/m(t)
"""
#making functions

def diffEQ(xNow,vNow,tNow):
    aNow = (m[i]*g-(b+psi)*vNow-k*xNow)/m[i]
    return aNow

#running the loop
for i in range(N-1):
    x1 = x[i]; v1 =v[i]
    x[i+1],v[i+1] = RK4(diffEQ,x1,v1,t,dt)
    t[i+1] = t[i] + dt
    m[i+1] = m[i] + dpsi


#plotting

#plotting the motion in x against time
plt.plot(t,x, '#803CA2', linewidth=2.0)
plt.title('Plot av utslag', fontsize=20)
plt.xlabel("Time [s]", fontsize=14)
plt.ylabel("Motion in x [m]", fontsize=14)
plt.tick_params(axis = 'both', which = 'major', labelsize =
    12)
plt.show()

#plotting the phaseplot
plt.plot(x,v, '#803CA2', linewidth=2.0)
plt.title('Plot i faserommet', fontsize=20)
plt.xlabel("Motion in x [m]", fontsize=14)
plt.ylabel("Velocity [m/s]", fontsize=14)
plt.tick_params(axis = 'both', which = 'major', labelsize =
    12)
#plt.savefig('Oppgave5del1.png')
```

```
plt.show()
```