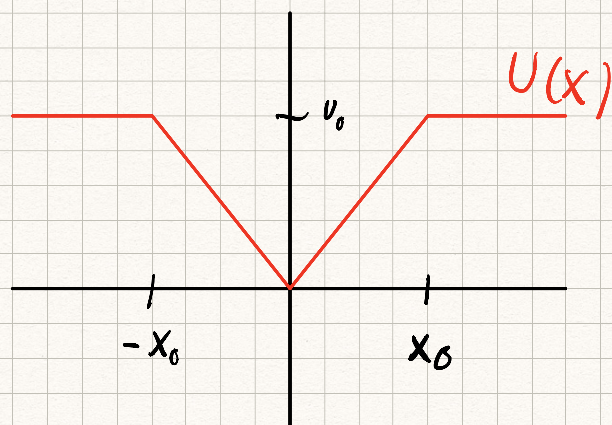
Oblig 4 - FYS-MEK1110

a)



Siden ingen andre krefter er med i bildet, har vi:

$$E_{mek} = K + U$$

(der K er den kinetiske og U den potensielle energien) fordi energien skal være bevart.

Likevektspunktene er $x = 0$, $x < -x_0$, eller $x > x_0$.

Stabiliteten i punktene som oppfyller de to sistnevnte er ganske lav siden det har mulighet til å falle til et lavere punkt dersom atomet ikke er i ro.

I $x = 0$ må atomet ha en fart som gir kinetisk energi: $K \geq U_0$. Retningen har ikke noe å si.

Dette er derfor et ganske stabilt likevekts-punkt.

b)

Kraften er den negative gradienten til potensialet:

$$F = -\nabla U$$

Ettersom vi bare regner i én dimensjon, tilsvarer dette å derivere potensialet:

$$F(x) = \begin{cases} \frac{U_0}{x_0} \cdot \frac{x}{|x|} & \text{for } 0 < |x| < x_0 \\ 0 & \text{for } x = 0, \quad |x| > x_0 \end{cases}$$

c)

Mekanisk energi er bevart:

$$E_{mek_0} = E_{mek_1}$$

$$K_0 + U_0 = K_1 + U_1$$

$$\boxed{|x| < x_0}$$

$$\frac{1}{2}m\left(\sqrt{\frac{4U_0}{m}}\right)^2 + 0 = \frac{1}{2}mv^2 + U_0\frac{|x|}{x_0}$$

$$2U_0 - U_0\frac{|x|}{x_0} = \frac{1}{2}mv^2$$

$$\frac{2}{m}\left(2U_0 - U_0\frac{|x|}{x_0}\right) = v^2$$

$$v = \sqrt{\frac{2\left(2U_0 - U_0\frac{|x|}{x_0}\right)}{m}}$$

Setter inn $x = \frac{x_0}{2}$ og får:

$$\underline{\underline{v = \sqrt{\frac{3U_0}{m}}}}$$

$$\boxed{|x| > x_0}$$

$$\frac{1}{2}m\left(\sqrt{\frac{4U_0}{m}}\right)^2 + 0 = \frac{1}{2}mv^2 + U_0$$

$$2U_0 - U_0 = \frac{1}{2}mv^2$$

$$\frac{2U_0}{m} = v^2$$

$$v = \sqrt{\frac{2U_0}{m}}$$

(Setter inn $x = 2x_0$)

$$v = \sqrt{\frac{2U_0}{m}}$$

d)

Bruker samme metode som i c):

$|x| > x_0$

$$v = \pm\sqrt{\frac{2\left(2U_0 - U_0\frac{|x|}{x_0}\right)}{m}}$$

setter inn $x = -\frac{x_0}{2}$

$$v = -\sqrt{\frac{3U_0}{m}}$$

$|x| < x_0$

$$v = \pm\sqrt{\frac{2U_0}{m}}$$

setter inn $x = -2x_0$

$$v = -\sqrt{\frac{2U_0}{m}}$$

e)

$$E_{mek_{slutt}} = E_{mek_{start}} + W$$

$K(0) = 0$	$K(0) = \frac{U_0}{2}$
$U_0 = 0 + W$	$U_0 = \frac{U_0}{2} + W$

$$\boxed{W = F_0 \cdot x_0}$$

$U_0 = F_0 \cdot x_0$	$\frac{U_0}{2} = F_0 \cdot x_0$
$F_0 = \frac{U_0}{x_0}$	$F_0 = \frac{U_0}{2x_0}$

f)

F er ikke konservativ fordi den avhenger av atomets kinetiske energi, og ikke bare posisjonen.

Dermed kan atomet bevege seg, og returnere til start-posisjonen med lavere energi enn den startet med, og F er per definisjon ikke konservativ.

g)

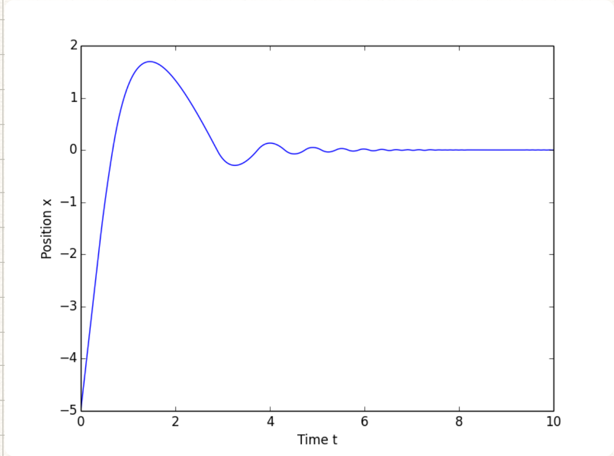
Initialbetingelsene er posisjon og fart.

$$a = \begin{cases} 0 & \text{for } |x| \geq x_0 \\ \dfrac{-\alpha v}{m} & \text{for } x = 0 \\ \dfrac{-\frac{U_0}{x_0} \cdot \frac{x}{|x|} - \alpha v}{m} & \text{for } 0 < |x| < x_0 \end{cases}$$

h)

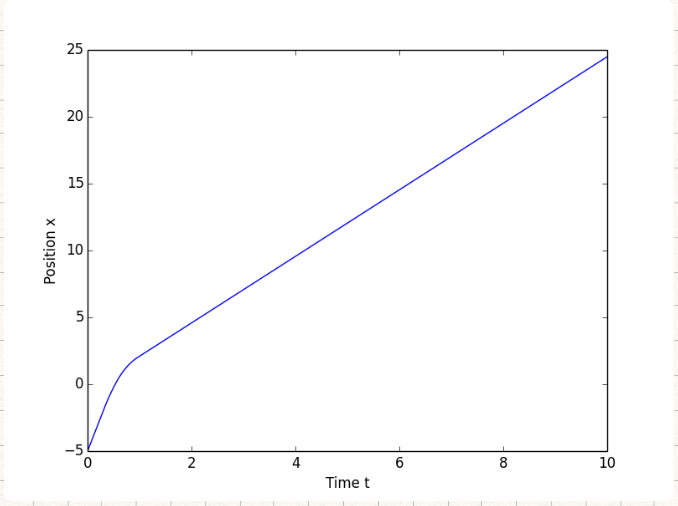
```python
import numpy as np
import matplotlib.pylab as plt

## Initial Conditions
U0 = 150
m = 23
x0 = 2
alpha = 39.48

T = 10
dt = 0.01
N = int(T/dt)

x = np.zeros(N)
v = np.zeros(N)
a = np.zeros(N)
t = np.linspace(0, T, N)

## Oppg i
v[0] = 8.0
x[0] = -5.0

## Oppg j
v[0] = 8.7
x[0] = -5.0

def U(x):
    if np.abs(x) >= x0:
        return U0
    else:
        return U0*(np.abs(x)/x0)

def F(x):
    if x > -x0 and x < 0:
        return U0/x0
    elif x < x0 and x > 0:
        return -U0/x0
    else:
        return 0

def F_photon(x, v):
    if np.abs(x) < x0:
        return -alpha * v
    else:
        return 0

for i in range(0, N-1):
    a[i+1] = (F(x[i]) + F_photon(x[i], v[i]))/m
    print a[i]
    v[i+1] = v[i] + a[i+1]*dt
    x[i+1] = x[i] + v[i+1]*dt
    t[i+1] = t[i] + dt


plt.xlabel("Time t")
plt.ylabel("Position x")
plt.plot(t, x)
#plt.savefig("oppg_j.png")
plt.show()
```

i)



Starthastigheten på 8 er lav nok til at MOT klarer å fange opp atomet, og etter den er fanget opp vil farten bare minke siden F avhenger av farten til atomet.

j)



Her er starthastigheten stor nok til at MOT ikke klarer å fange den opp, og atomet fortsetter uendelig i en retning etter at $|x| > x_0$, ettersom ingen krefter fra vårt system virker på atomet utenfor $[-x_0, x_0]$

k)

```python
for v0 in np.linspace(8, 10, 1000):
    v[0] = v0


    for i in range(0, N-1):
        a[i+1] = (F(x[i]) + F_photon(x[i], v[i]))/m
        v[i+1] = v[i] + a[i+1]*dt
        x[i+1] = x[i] + v[i+1]*dt
        t[i+1] = t[i] + dt


    if np.abs(x[-1]) < x0:
        trapped_velocity = v0
    else:
        print trapped_velocity
        break
```

$\underline{\underline{V_0 \approx 8,75}}$

Lagde enda en for-loop rundt hovedlokken som kjører hele scenarioet for 1000 forskjellige verdier for V_0.
Første gang atomet ender opp utenfor MOT brytes lokken, og forrige verdi for V_0 printes.