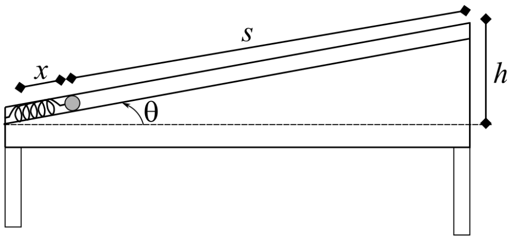
Løsning øving 4

Oppgave 1

Vi kan bruke bevaring av mekanisk energi: i det kula skytes ut, går potensiell energi i fjæra over til kinetisk og potensiell energi for kula (ettersom kula glir friksjonsfritt mot underlaget, vil den ikke rulle). Idet kula når sitt høyeste punkt en høyde h over punktet der fjæra var maksimalt sammenpresset, har all mekanisk energi gått over til potensiell energi for kula. Se figuren under.

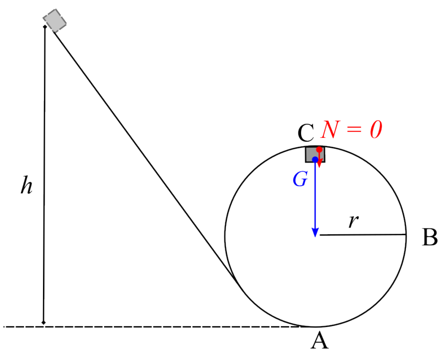


Hvis vi velger nullnivå i punktet der fjæra er maksimalt sammenpresset, får vi:

$$\begin{aligned}
\frac{1}{2}kx^2 &= mgh \\
\frac{1}{2}kx^2 &= mg\,(s+x)\sin\theta \\
k &= \frac{2mg\,(s+x)\sin\theta}{x^2} \\
&= \frac{2\cdot 0,080\,\mathrm{kg}\cdot 9,81\,\mathrm{m/s^2}\cdot(0,78\,\mathrm{m}+0,070\,\mathrm{m})\sin 15^\circ}{(0,070\,\mathrm{m})^2} \\
&= 70,47\,\mathrm{N/m} \\
&\approx \underline{\underline{70\,\mathrm{N/m}}}
\end{aligned}$$

Oppgave 2

a) Skal bestemme den minste høyden h vogna kan slippes fra for at vogna skal fullføre en hel loop uten å miste kontakten med underlaget. Se figuren under.



Det kritiske punktet er toppunktet C, ettersom farten her er minst. I grensetilfellet at vogna akkurat klarer å fullføre en loop, er normalkrafta $N = 0$ i dette punktet (med andre ord: det er tyngden G som alene besørger den nødvendige sentripetalkraften for å opprettholde sirkelbevegelsen).

Newtons 2. lov i toppunktet:

$$\sum F = ma = m\frac{v^2}{r}$$
$$mg = \frac{mv^2}{r}$$
$$\underline{v = \sqrt{gr}}$$

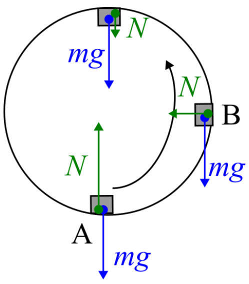
Dette er altså den minste farten vogna må ha i C for å klare en hel loop. Vi kan da bruke energibevaring til å finne den minimale starthøyden h som gjør dette mulig: med nullnivå i punkt A (nederst i loopen) får vi

$$mgh = \frac{1}{2}mv^2 + mg \cdot 2r,$$

ettersom punkt C befinner seg i høyde $2r$ over punkt A. Får at

$$gh = \frac{1}{2}v^2 + 2gr$$
$$gh = \frac{1}{2}\left(\sqrt{gr}\right)^2 + 2gr$$
$$gh = \frac{1}{2}gr + 2gr$$
$$h = \underline{\underline{\frac{5}{2}r}}$$

b) Når vogna slippes fra en høyde $h = 3r$, vil farten være større enn i a). Figuren under viser kreftene på vogna i dette tilfellet:



Nå vil altså $N > 0$ i det øverste punktet i loopen, der Newtons 2. lov gir:

$$\sum F = ma = m\frac{v^2}{r}$$
$$mg + N = m\frac{v^2}{r}$$
$$N = m\frac{v^2}{r} - mg$$

Ettersom vi skal finne N uttrykt ved $G = mg$, kan vi skrive

$$\frac{N}{G} = \frac{N}{mg} = \frac{\frac{mv^2}{r}}{mg} - \frac{mg}{mg}$$
$$= \underline{\frac{v^2}{gr} - 1}$$

Vi bruker akkurat samme energibevaringslikning som i forrige oppgave til å finne farten v i toppunktet:

$$\underbrace{mgh}_{\text{potensiell energi i startpunkt}} = \underbrace{\frac{1}{2}mv^2}_{\text{kinetisk energi i toppunkt}} + \underbrace{mg \cdot 2r}_{\text{potensiell energi i toppunkt}}$$

$$\begin{aligned} gh &= \frac{1}{2}v^2 + 2gr \\ v^2 &= 2\left(gh - 2gr\right) \\ v &= \sqrt{2\left(g \cdot 3r - 2gr\right)} \\ &= \underline{\sqrt{2gr}} \end{aligned}$$

Det gir:

$$\begin{aligned} \frac{N}{G} &= \frac{v^2}{gr} - 1 \\ &= \frac{\left(\sqrt{2gr}\right)^2}{gr} - 1 \\ &= \frac{2gr}{gr} - 1 \\ &= \underline{\underline{1}} \end{aligned}$$

c) I punkt B er det kun normalkrafta som bidrar til sentripetalakselerasjonen - tyngden $G = mg$ virker tangentielt og bidrar ikke. Newtons 2. lov for sentripetalakselerasjonen gir da

$$\begin{aligned} \sum F &= ma = m\frac{v^2}{r} \\ N &= m\frac{v^2}{r} \\ \frac{N}{G} &= \frac{N}{mg} = \frac{\frac{mv^2}{r}}{mg} = \underline{\frac{v^2}{gr}} \end{aligned}$$

Bruker energibevaring til å finne farten i punkt B, som ligger en høyde lik r over nullnivået:

$$\underbrace{mgh}_{\text{potensiell energi i startpunkt}} = \underbrace{\frac{1}{2}mv^2}_{\text{kinetisk energi i B}} + \underbrace{mg \cdot r}_{\text{potensiell energi i B}}$$

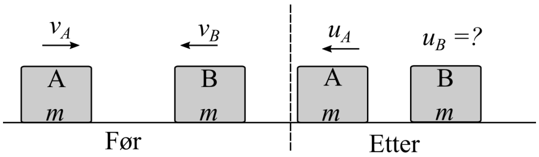
$$\begin{aligned} gh &= \frac{1}{2}v^2 + gr \\ v^2 &= 2\left(gh - gr\right) \\ v &= \sqrt{2\left(g \cdot 3r - gr\right)} \\ &= \underline{\sqrt{4gr}} \end{aligned}$$

Det gir følgende for normalkrafta i B:

$$\begin{aligned} \frac{N}{G} &= \frac{v^2}{gr} \\ &= \frac{\left(\sqrt{4gr}\right)^2}{gr} \\ &= \frac{4gr}{gr} \\ &= \underline{\underline{4}} \end{aligned}$$

Oppgave 3

a) Figuren under viser kollisjonen:



Velger positiv retning mot høyre, slik at alle farter mot venstre er negative.

Skal finne sluttfarten u_B til den ene steinen når $v_A = 3,0\,\text{m/s}$ og $v_B = -5,0\,\text{m/s}$, og $u_A = -4,5\,\text{m/s}$. Bevaring av bevegelsesmengde gir:

$$\begin{aligned} \sum_{\text{før}} p &= \sum_{\text{etter}} p \\ mv_A + mv_B &= mu_A + m \cdot u_B \\ v_A + v_B &= u_A + u_B \\ u_B &= v_A + v_B - u_A \\ u_B &= 3,0\,\text{m/s} + (-5,0\,\text{m/s}) - (-4,5\,\text{m/s}) \\ &= \underline{\underline{2,5\,\text{m/s}}} \end{aligned}$$

Det positive indikerer at stein B går mot høyre etter støtet.

b) I et elastisk støt er kinetisk energi bevart. Beregner kinetisk energi før og etter støtet:

Før:

$$\begin{aligned} K_{\text{før}} &= \frac{1}{2}mv_A^2 + \frac{1}{2}mv_B^2 \\ &= \frac{1}{2}m\left(v_A^2 + v_B^2\right) \\ &= \frac{1}{2} \cdot 19\,\text{kg} \cdot \left((3,0\,\text{m/s})^2 + (-5,0\,\text{m/s})^2\right) \\ &= \underline{323\,\text{J}} \end{aligned}$$

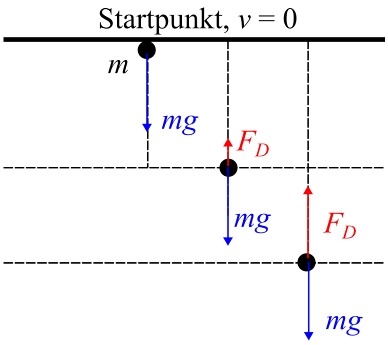
Etter:

$$\begin{aligned}
K_{\text{etter}} &= \frac{1}{2}mu_A^2 + \frac{1}{2}mu_B^2 \\
&= \frac{1}{2}m\left(u_A^2 + u_B^2\right) \\
&= \frac{1}{2}\cdot 19\,\text{kg}\cdot\left((-4,5\,\text{m/s})^2 + (2,5\,\text{m/s})^2\right) \\
&= \underline{252\,\text{J}}
\end{aligned}$$

Ettersom noe kinetisk energi har gått tapt i støtet, var det **ikke** elastisk.

Oppgave 4

a) Figuren under viser kreftene som virker på en fallskjermhopper i de ulike fasene av hoppet, fra utspranget (der $v = 0$), til punktet der terminalfarten er nådd: tyngden $G = mg$ (som kan antas konstant under hoppet), samt luftmotstanden F_D, som øker kvadratisk med farten v:



I punktet der terminalfarten er nådd, er

$$\begin{aligned}
F_D &= mg \\
\frac{1}{2}\rho A C_d v^2 &= mg \\
v &= \underline{\sqrt{\frac{2mg}{\rho A C_d}}}
\end{aligned}$$

For de de hopperne blir terminalhastighetene hhv.

For A:

$$\begin{aligned}
v &= \sqrt{\frac{2\cdot 70\,\text{kg}\cdot 9,81\,\text{m/s}^2}{1,29\,\text{kg/m}^3\cdot 0,17\,\text{m}^2\cdot 0,70}} \\
&= 94,6\,\text{m/s} \\
&\approx \underline{\underline{340\,\text{km/h}}}
\end{aligned}$$

For B:

$$\begin{aligned}
v &= \sqrt{\frac{2\cdot 70\,\text{kg}\cdot 9,81\,\text{m/s}^2}{1,29\,\text{kg/m}^3\cdot 1,0\,\text{m}^2\cdot 1,0}} \\
&= 32,6\,\text{m/s} \\
&\approx \underline{\underline{117\,\text{km/h}}}
\end{aligned}$$

b) Python-kode for funksjon som beregner luftmotstanden som funksjon av frontareal A, dragkoeffisient C og fart v, for en gitt (konstant) verdi av luftas massetetthet:

```
def drag(A,C,v):
    rho=1.28
    k=0.5*rho*A*C
    return k*v**2
```

c) Under er Jupyter notebook-kode for å bestemme falltid t og falt høyde s idet farten er en viss prosentandel av terminalfarten.

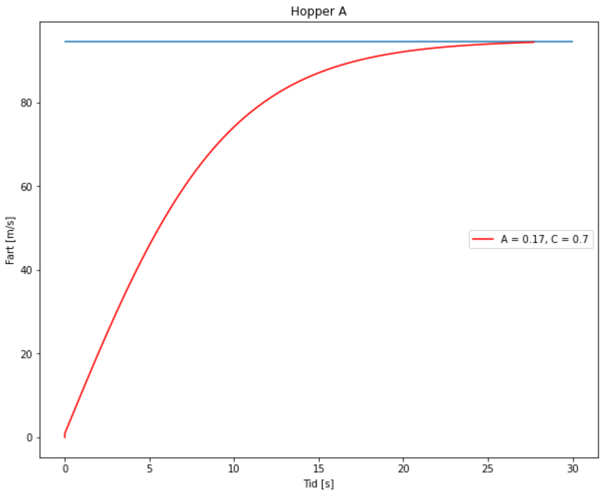
```
#Rutiner for simulering av vertikalt fall med luftmotstand. I dette
    eksemplet er positiv retning nedover.

import numpy as np
import matplotlib.pyplot as plt

# Globale konstanter
m=70 #Legemets masse i kg
g=9.81 #Tyngdeakselerasjonen i m/s^2

def drag(A,C,v):
    rho=1.29
    k=0.5*rho*A*C
    return k*v**2

def dXdt(X):
   """Funksjonen dXdt beregner høyresiden f(X) i
      differensiallikningssystemet; dX/dt=f(X).
   Input:
   X:  X=[y,v] en vektor som inneholder posisjon y og (vertikal) fart
       v. Med positiv retning nedover er v > 0 og y < 0
   for et legeme som faller vertikalt mot bakken, der y = 0.

   Output:
   [dydt,dvdt]: Array med nye verdier for hastighet (dydt) og
      akselerasjon (dvdt)
   """
   y , v =X          #Koordinater y og v hentes fra inndatavektor X
   f=drag(A,C,v)     #Luftmotstand i N
   dydt=v            #Sammenhengen mellom y og v er at v = dy/dt
   dvdt=-f/m+g       #Akselerasjonen a=dv/dt, fra Newtons 2. lov
   return np.array([dydt,dvdt])

def euler(t0,y0,v0,dt):
    """Funksjon som bruker Eulers metode til å løse et system av
       differensiallikninger dX/dt = f(X),
    der X =[y,v] er en vektor som inneholder posisjons- og
       hastighetsvariable.
    Input:
    t0: Starttid [s]
    y0: Startverdi for y [m]
    v0: Vertikal startfart [m/s]
```

```
    dt: Tidssteg [s]

    Output:
    t_liste: array med t-verdier,[t0,...tn]
    y_liste: array med y-verdier, [y0,...,yn]
    v_liste: array med v-verdier, [v0,...,vn]
    """

    X0=np.array([y0,v0]) #X0 er en vektor med posisjon og fart ved t=
        t0
    t_liste=[0.0]# Liste med t-verdier
    y_liste=[y0]# liste med y-verdier
    v_liste=[v0] # liste med v-verdier
    X=X0  # initierer loop
    t=t0
    y=y0
    while y<=0: #Loop kjøres inntil legemet treffer bakken; med pos.
        retning nedover er y0 < 0
        Xn=X+dt*dXdt(X) #Beregner neste steg Xn i Euler-metoden
        y=Xn[0] #Henter ut y-koordinat fra array
        v=Xn[1] #Henter ut fart v fra array
        t_liste.append(t) # t-verdi legges til liste
        y_liste.append(y)# y-verdi legges til liste
        v_liste.append(v)# v-verdi legges til liste
        t=t+dt #Ny tidsverdi
        X=Xn #Ny verdi for X
    return t_liste,y_liste,v_liste

def tid_falt_hoyde(p,v_term,t_liste,y_liste,v_liste):
    """Funksjonen beregner tid og falt høyde idet hastigheten er p*
        terminalfarten, 0 <= p <= 1
    Input:
    p: Prosentandel av terminalfart, f.eks. 0.98
    v_term: Terminalfart [m/s]
    t_liste,y_liste,v_liste: Lister med verdier for hhv, t, y og v

    Output:
    tid: Tiden det tar til farten er p*terminalfart [s]
    hoyde: Vertikal fallhøyde idet farten er p*terminalfart [m]
    """
    indeks=np.argmax(np.array(v_liste)>p*v_term) #Gir første indeks i
         array der betingelse er oppfylt
    tid=t_liste[indeks]-t0 #Beregner falltiden fram til betingelsen
        er oppfylt
    hoyde=y_liste[indeks]-y0 #Beregner falt høyde
    return tid,hoyde
```

```
#Initialiserer variable
t0=0.0 #t = 0 i startpunktet
v0=0 #Startfart
y0=-2000 #Starthøyde over bakken
```

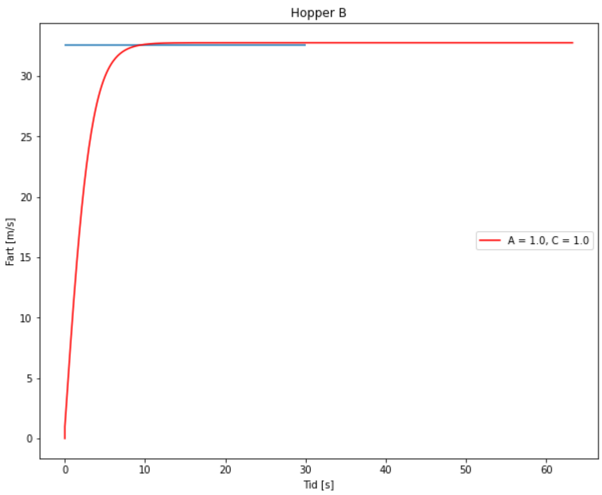
```
dt=0.1 #Tidssteg

#Konstanter for hopper A
A1=0.17 #Frontareal
C1=0.7 #Drag-koeffisient
v_term1=94.6 #Terminalfart [m/s]

#Konstanter for hopper B
A2=1.0 #Frontareal
C2=1.0 #Drag-koeffisient
v_term2=32.6 #Terminalfart [m/s]

#Simulering for hopper A
#Bestemmer v(t) og plotter
A=A1
C=C1
v_term=v_term1
t_liste,y_liste,v_liste=euler(t0,y0,v0,dt)

plt.figure(figsize = (10, 8))
plt.plot(t_liste,v_liste,color="red",label='A = '+str(A)+", C = "+str
    (C)) #Diagrammet angir hvilke verdier for A og C som er tilhører
    grafen
plt.hlines(v_term,0,30)
plt.xlabel("Tid [s]")
plt.ylabel("Fart [m/s]")
plt.title("Hopper A")
plt.legend(loc='right')
plt.show()

#Finner strekningen hopperen har falt idet v er 98 % av v_term. Angir
     verdier med 2 desimaler.
t,h=tid_falt_hoyde(0.98,v_term,t_liste,y_liste,v_liste)
print("Falltid [s]: ","%.2f" % t)
print("Falt høyde [m]: ","%.2f" % h)

#Simulering for hopper B
#Bestemmer v(t) og plotter
A=A2
C=C2
v_term=v_term2
t_liste,y_liste,v_liste=euler(t0,y0,v0,dt)
#np.array=euler(t0,y0,v0,dt)

plt.figure(figsize = (10, 8))
plt.plot(t_liste,v_liste,color="red",label='A = '+str(A)+", C = "+str
    (C)) #Diagrammet angir hvilke verdier for A og C som er tilhører
    grafen
plt.hlines(v_term,0,30)
plt.xlabel("Tid [s]")
plt.ylabel("Fart [m/s]")
```

```
plt.title("Hopper B")
plt.legend(loc='right')
plt.show()

#Finner strekningen hopperen har falt idet v er 98 % av v_term. Angir
    verdier med 2 desimaler.
t,h=tid_falt_hoyde(0.98,v_term,t_liste,y_liste,v_liste)
print("Falltid [s]: ","%.2f" % t)
print("Falt høyde [m]: ","%.2f" % h)
```



Falltid [s]: 22.00

Falt høyde [m]: 1465.34



Falltid [s]: 7.40

Falt høyde [m]: 169.93

Kommentar: Vi ser at hopper B, som hopper med utstrakte armer og har stort frontareal og drag-koeffisient, når sin terminalfart mye raskere enn den «stupende» hopper A.