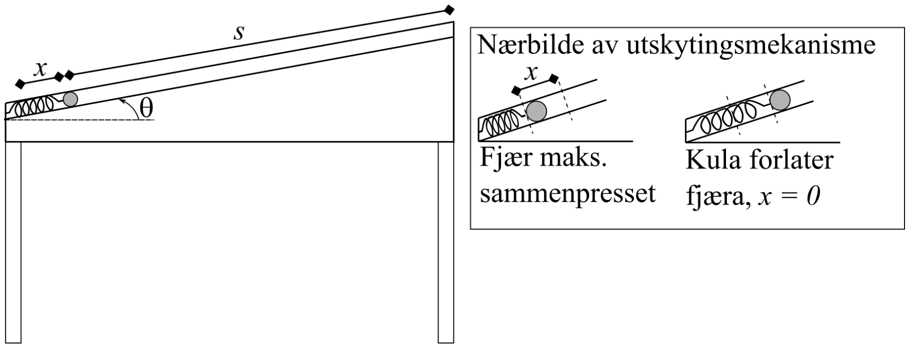
Øving 4

Oppgave 1

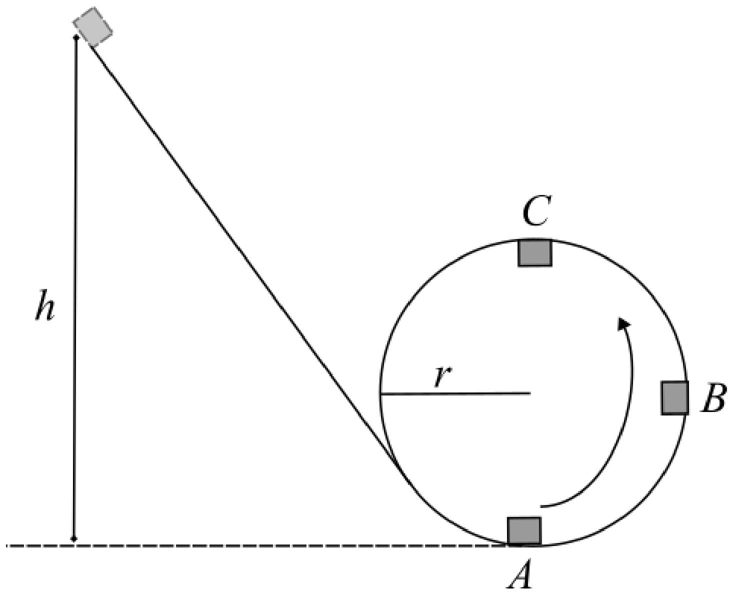
Kula i et flipperspill skytes ut fra en fjærbelastet avtrekker. Spilleren trekker i fjæra slik at den presses sammen en avstand $x = 7,0\ \mathrm{cm}$. Flipperspillet har en helningsvinkel $\theta = 15^\circ$. Se figuren under.



Hvor stor må fjærkonstanten k til fjæra være dersom kula akkurat skal nå toppen av flipperspillet, som ligger en avstand $s = 78\ \mathrm{cm}$ fra punktet der kula forlater fjæra (i punktet der fjæra er slapp)?

Oppgave 2

En vogn i en berg-og-dalbane starter i en viss høyde h over det laveste punktet A i en sirkulær loop med radius R. To andre punkter i loopen er markerte: B er midtveis oppe, og C er det høyeste punktet. Se figuren under.



I denne oppgaven skal vi se bort fra friksjon og luftmotstand.

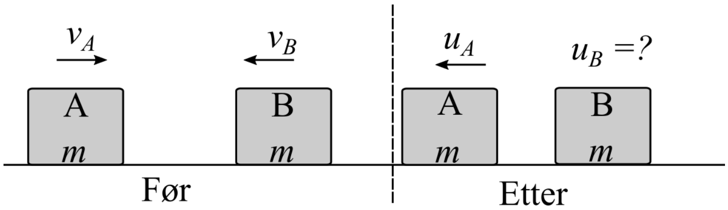
a) Fra hvilken høyde h over punkt A må vogna slippes for at den skal kunne fullføre en hel loop uten å miste kontakten med underlaget?

b) Hva er normalkrafta på vogna i det høyeste punktet i loopen dersom den slippes fra en starthøyde $h = 3R$?

Oppgave 3

To curlingsteiner A og B med identisk masse $m = 19\ \mathrm{kg}$ kolliderer i et rett, sentralt støt. Før støtet har stein A fart $v_\mathrm{A} = 3,0\ \mathrm{m/s}$ mot høyre og stein B fart $v_\mathrm{B} = 5,0\ \mathrm{m/s}$ mot venstre, og etter støtet har stein A fart

$v_A = 1,0 \text{ m/s}$ mot venstre. Se figuren under.



a) Finn farten til stein B etter støtet.

b) Var støtet elastisk?

Oppgave 4

To fallskjermhoppere A og B med identiske masser $m = 70 \text{ kg}$ faller vertikalt fra samme starthøyde. A faller med hodet først og har frontareal $A = 0,17 \text{ m}^2$ og "drag-koeffisient" $C_d = 0,70$; B faller liggende og har $A = 1,0 \text{ m}^2$ og $C_d = 1,0$. Vi forutsetter kvadratisk luftmotstand $F_D = \frac{1}{2}\rho A C_d v^2$, og tyngdeakselerasjonen g kan antas konstant over fallhøyden.

a) Finn ved regning terminalhastighetene for A og B.

I de to siste oppgavene skal vi bruke Python til å løse bevegelseslikningene numerisk.

b) Gitt følgende Python-funksjon som beregner luftmotstanden (drag) som funksjon av frontareal A, drag-koeffisient C_d og fart v, for en gitt (konstant) verdi av luftas massetetthet ρ:

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

Hvilken Python-kode skal stå i linja markert med ? for at funksjonen skal gi luftmotstanden i newton som funksjon av inndataene?

A. `return k*v**2` B. `return k*(v/3.6)**2` C. `return k*(v/3,6)**2` D. `return k*(v*3.6)**2` E. `return k*(v*3,6)**2`

c) Ta utgangspunkt i den ferdige rutinen for Eulers metode med tidssteg $\Delta t = 0,10 \text{ s}$ til å beregne tiden t det tar det før hopperne har nådd 98 % av sine terminalhastigheter, og strekningen s hopperen har falt på dette tidspunktet (målt fra startpunktet).

In [ ]:

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

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=1.28
    k=0.5*rho*A*C
    #? Her må du legge inn riktig kode for at funksjonen skal returnere luftmotstanden (drag) i

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 ned
   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 = 
    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
```

In [ ]:

```
#Eksempel på bruk av rutinene
#Initialiserer variable
t0=0.0 #t = 0 i startpunktet
v0=0 #Startfart
y0=-1000 #Med positiv retning nedover, er y-verdier negative over bakken. Her starter fallet 1,
dt=0.01 #Tidssteg
A=0.9 #Frontareal
C=0.8 #Drag-koeffisient

#Bruker Eulers metode til å generere sammenhengende verdier for t, y og v
t_liste,y_liste,v_liste=euler(t0,y0,v0,dt)

#Plotter fartsgraf v(t)
plt.figure(figsize = (10, 8))
plt.plot(t_liste,v_liste,color="red",label='A = '+str(A)+", C = "+str(C)) #Diagrammet angir hvi
plt.xlabel("Tid [s]")
plt.ylabel("Fart [m/s]")
plt.legend()
plt.show()
```