Skråplan

Simon Bakken-Jantasuk

21. desember 2022

Innhold

1 Introduksjon **2**
1.1 Hensikt 2
1.2 Oppsummering 2

2 Teori **2**

3 Fremgangsmåte **4**
3.1 Målinger 4

4 Resultat **5**
4.1 Python: akselerasjon 5
4.2 Python: bevegelse 5

Tabeller

1 strekning (cm) 4

1 Introduksjon

1.1 Hensikt

1. Undersøke bevegelsen til en boks som beveger seg opp et skråplan

1.2 Oppsummering

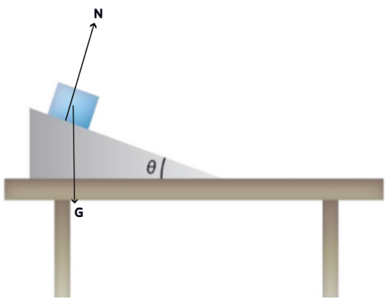
Undersøkte bevegelsen til en boks, og har funnet ut at

Utstyr

1. Gradskive
2. Boks
3. Fjær
4. Linjal

2 Teori

a)



Figur 1: I ro, så har vi normalkraft og gravitasjonskraft

b)

Vi vet at,

$$\mu = \frac{R}{N}$$

Hvor $G_x = R$

$$R = G \sin\phi \wedge N_y = G \cos\phi$$

Det vil si,

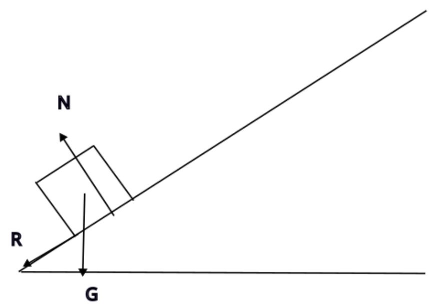
$$\mu = \frac{mg \sin\phi}{mg \cos\phi}$$

Vi vet at,

$$\tan\phi = \frac{\sin\phi}{\cos\phi}$$

Da vet vi at,

$$\mu = \tan\phi, Q.E.D$$



Figur 2: I bevegelse opp: normalkraft, gravitasjonskraft og friksjonkraft

c)

Uttrykk for akselerasjonen blir,

$$\Sigma F = ma$$
$$\vec{G_x} + \vec{R_x} = ma_x$$
$$a_x = \frac{\vec{G_x} + \vec{R_x}}{m}$$

Vi vet at,

$$G_x = mg\sin\phi \wedge R_x = \mu mg\cos\phi$$

Da blir,

$$a_x = \frac{mg\sin\phi + \mu mg\cos\phi}{m}$$
$$a_x = g(\sin\phi + \mu\cos\phi)$$

d)

For et legeme som beveger seg ned et skråplan,

$$a_x = g(\sin\phi - \mu\cos\phi)$$

Her så blir R negativ.

e)

Ifølge formelen så er $a_{opp} - a_{ned} = \mu 2g\cos\phi$, men dersom vi ikke bruker dette som utgangspunkt

Men a_{opp} og a_{ned}, så får vi,

$$g(\sin\phi + \mu\cos\phi) - g(\sin\phi - \mu\cos\phi) = 2\mu g\cos\phi$$
$$\mu = \frac{a_{opp} - a_{ned}}{2g\cos\phi}, Q.E.D$$

Dette er lov fordi vi har brukt a_{opp} og a_{ned} som utgangspunkt.

3 Fremgangsmåte

1. Velger en vinkel ϕ
2. Bruker fjæren for å skyte opp boksen
3. Gjentar (2) flere ganger
4. Tar målinger av (3)

3.1 Målinger

Vinkel $\phi = 9^\circ \pm 0.1^\circ$ Lengden av boksen er 25×10^{-2} cm

Tabell 1: strekning (cm)

S_1	81
S_2	84
S_3	82
S_4	81
S_5	85
S_6	84
S_7	85
S_8	84
S_9	81
S_{10}	85
S_{11}	85

4 Resultat

4.1 Python: akselerasjon

```
from math import *

g = 9.81 # m/s^2
friksjonsKonstant = 0.17

def akselerasjon(vinkel, retning):
  if retning == "opp":
    return g * (sin(vinkel) + friksjonsKonstant * cos(vinkel))
  elif retning == "ned":
    return g * (sin(vinkel) - friksjonsKonstant * cos(vinkel))
  else:
    raise ValueError("Retningene må være opp eller ned")

akselerasjonOpp = round(akselerasjon(radians(9), "opp"),20)
akselerasjonNed = round(akselerasjon(radians(9), "ned"),20)

print(akselerasjonOpp)
print(akselerasjonNed)

def friksjonTall(vinkel, akselerasjonOpp, akselerasjonNed):
    return tan(vinkel), (akselerasjonOpp - akselerasjonNed)/(2*g*
        ↪ cos(vinkel))

print(friksjonTall(radians(9),akselerasjonOpp, akselerasjonNed))
```

```
Output:
        3.1817899476551763
        -0.11254574356584655
        (0.15838444032453627, 0.17)
```

4.2 Python: bevegelse

```
from pylab import *

g = 9.81
angle = radians(9)
mu = 0.17

x = 0
v = 0


strekning = []
tid = []
```

```
t = 1

while t <= 10:
    a = g * (sin(angle) + mu * cos(angle))
    v = v + a * t
    x = v * t + 1/2 * a * t ** 2
    strekning.append(x)
    tid.append(t)
    t = t + 1

plot(tid, strekning)
show()
```