2022
høst

Fysikk og programmering (iPython)

15.1 Bevegelsesgrafer

Fall med luftmotstand, for en ball.
Tekstfil på RSTnett: ballslipp.txt

(i sekund) Tid (t)	(i meter) Fallengde (s)
0,0	0,0
0,5	1,3
1,0	5,1
osv.	osv.
10,0	274,7

a) Lag program som leser dataene og tegner en s-t-graf.

hvit tekst →

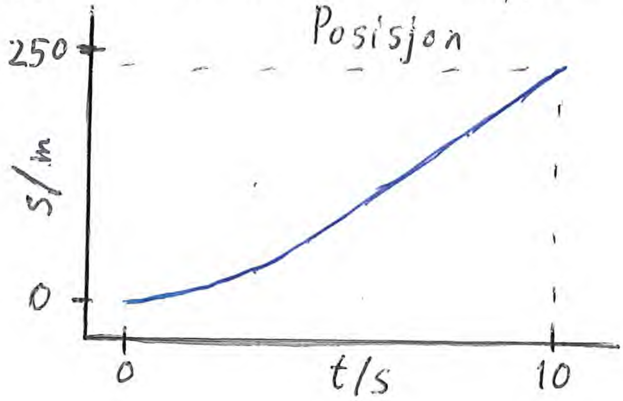
```
# Program ballposisjon
import numpy as np
import matplotlib.pyplot as plt

# Leser t- og s-verdier inn i lister
data=np.loadtxt('ballslipp.txt') #laster tekstfil
t=data[:,0]          #innleste t-verdier [s]
s=data[:,1]          #innleste s-verdier [m]

# Tegner posisjonsgrafen
plt.figure(1)
plt.plot(t,s)
plt.title('Posisjon')
plt.xlabel('$t$ / s')
plt.ylabel('$s$ / m')
plt.show()
```

grønn tekst

Husk å legge pythonprogrammet og txt-fila i samme mappe.
Klikk Plots i vinduet til høyre i Spyder for å se plottet.

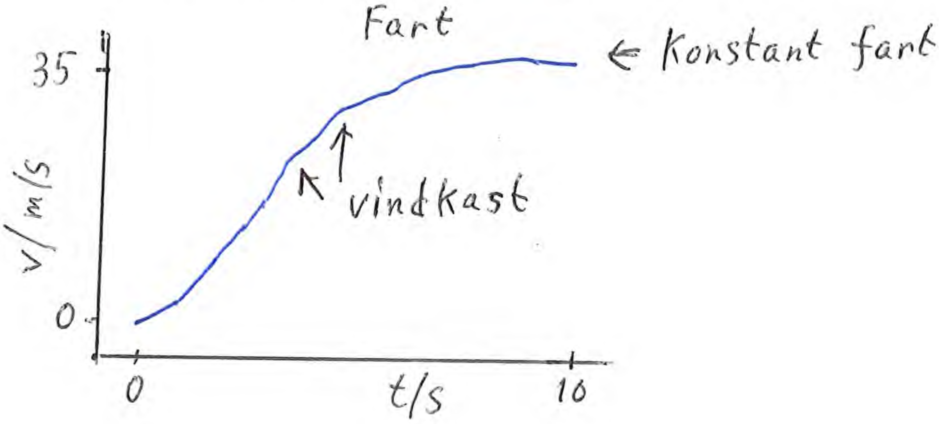


I lager liste over fartene med en for-løkke der vi bruker definisjonen $\bar{v} = \frac{s_2 - s_1}{t_2 - t_1}$

```
# Stegvis beregning av v
for i in range(1,n):
    v[i] =(s[i]-s[i-1])/(t[i]-t[i-1])
```

Tegner så grafen med nye navn på aksene og ny tittel:

```
plt.title('Fart')  osv.
```



15.2 Modellering av fall med luftmotstand

$$\sum F = ma$$

$+\Uparrow$ $$-G + L = ma$$

$$a = \frac{-mg + kv^2}{m}$$

$$a = -g + \frac{k}{m}v^2$$ dvs. $a = -g + rv^2$ med $r = \frac{k}{m}$

Vi splitter bevegelsen i bittesmå tidsintervall der akselerasjonen er konstant. Dette er nesten riktig.

Vi bruker programmet til å bestemme a i hver del for seg fra $a = -g + rv^2$.
Vi kan da beregne farten i hver del fra $v = v_0 + at$
og posisjonen fra $s = s_0 + vt$

I programmet blir dette til lister etter beregningen

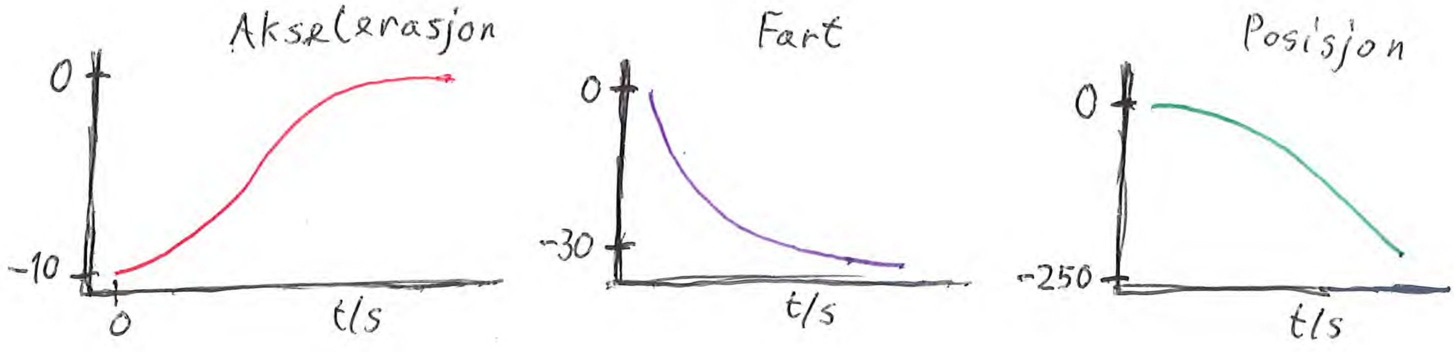
```
for i in range(n-1):
    a[i+1] = -g + r*v[i]**2
    v[i+1] = v[i] + a[i]*dt
    s[i+1] = s[i] + v[i]*dt
```

På forhånd har vi laget en liste for t-verdiene

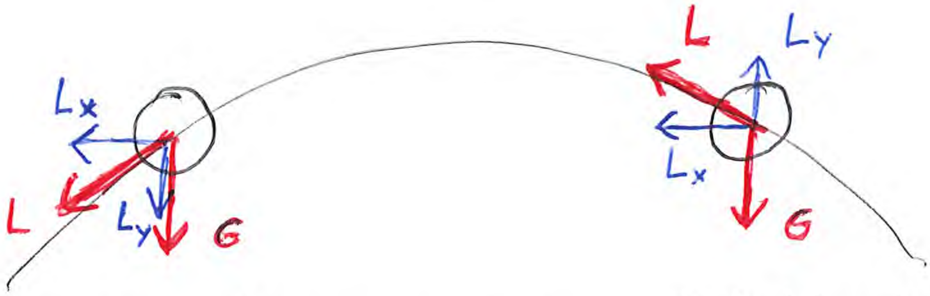
```
dt = 0.001                      # tidssteg
n = int((tMax/dt))              # antall steg
t = np.linespace(0,tMax,n)      # t-liste
```

Tegner så grafene i slutten av programmet

```
# Akselerasjonsgraf
plt.figure(1)
plt.plot(t,a,'-r')
plt.grid( )
plt.title('Akselerasjon')
plt.xlabel('$t$ / s')
plt.ylabel('$a$ / m/s^2')
   osv.
plt.show( )
```



15.3 Skrått kast med luftmotstand



Både størrelse og retning på luftmotstand varierer

$$\sum \vec{F} = m\vec{a}$$

$$\vec{G} + \vec{L} = m\vec{a} \quad \text{der } L = kv^2 \text{ og } \vec{L} = -kv \cdot \vec{v}$$

mot fartsretningen

Deler i komponenter og får:

$$G_x + L_x = ma_x \qquad G_y + L_y = ma_y$$

$$0 - kvv_x = ma_x \qquad -mg - kvv_y = ma_y$$

$$a_x = -\frac{k}{m}vv_x \qquad a_y = -g - \frac{k}{m}vv_y$$

$$a_x = -rvv_x \qquad a_y = -g - rvv_y \text{ med } r = \frac{k}{m}$$

Kan nå lage bevegelsesgraf.

Uten luftmotstand blir $x = v_{0x}t$

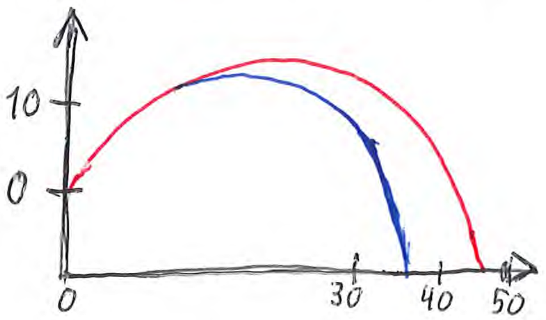
og $y = v_{0y}t + \frac{1}{2}a_y t^2$ der $a_y = -g$

Kan lage program som viser kastebanen for en ball med $r = 0{,}010\ m^{-1}$ med og uten luftmotstand.

$v_0 = 20\ \frac{m}{s}$ og startvinkel er 60° opp.

Bruker programmet i eks. 15,2, men nå i både x- og y-retning.

```
ax[i+1] = -r*v[i+1]*vx[i+1]          # akselerasjon i x-retn.
ay[i+1] = -g - r*v[i+1]*vy[i+1]      # aks. i y-retning
```



15.4 Svingninger. Harmonisk oscillator

Akselerasjonen i svingebevegelsen til en elastisk pendel er

$$a = -\frac{k}{m} \cdot x \qquad \text{(se side 376 2022bok)}$$

Vi deler bevegelsen i ørsmå tidsintervaller med tilnærmet konstant aks. i hvert intervall. Beregner da a, v og x steg for steg. med

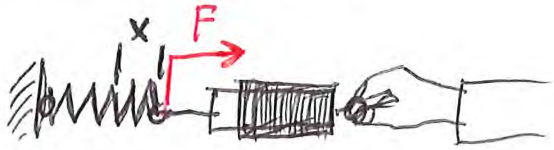
```
for i in range(n-1):
    x[i+1] = x[i] + v[i]*dt
    v[i+1] = v[i] + a[i]*dt
    a[i+1] = -k/m * x[i+1]
```

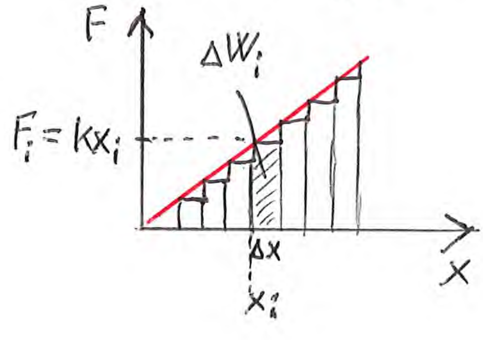
Kan så tegne grafene for akselerasjon, fart og posisjon.

15.5 Arbeid og numerisk integrasjon

Arbeid på elastisk fjær.

$W = Fs$ og $F = kx$





Splitter opp i

$$\Delta W_i = F_i \Delta x = kx_i \Delta x$$

Slik at

$$W = \Delta W_1 + \Delta W_2 + \dots \Delta W_n$$

$$= \sum \Delta W_i = \underbrace{\sum k x_i \Delta x}$$

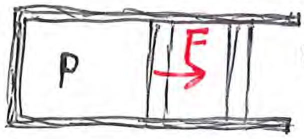
Å summere dette kalles numerisk integrasjon.

<u>Eks. 15.5</u> $k = 300 \frac{N}{m}$ og fjæra strekkes 5,0 cm fra likevekt. Beregn arbeidet!

```
# Program fjærarbeid.py
# Definerer størrelser
def F(x):
    k = 300
    return k*x
x1 = 0
x2 = 0.05
x = x1
n = 200
dx = (x2-x1)/n
W = 0
# Summerer opp ΔW-rektanglene
while x <= x2:
    W = W + F(x)*dx
    x = x + dx
# Skriver ut arbeidet W
print(f'W = {W:.4f} J')     # utskrift med 4 desimaler
```

Arbeid når en gass endrer volum:

$\Delta W = p\Delta V$



W på omgivelsene

$W = \Delta W_1 + \Delta W_2 + \cdots \Delta W_n = \sum \Delta W_i = \sum p_i \Delta V$

Eks 15.6 Isoterm prosess, idealgass

Program volumarbeid_isoterm.py

```
def p(V):
    n =
    R =
    T =
    return n*R*T/V
```

og V1 = 0.015, V2 = 0.030, V = V1, N = 10, dV = (V2-V1)/N, W = 0

og
```
while V <= V2:
    W = W + p(V)*dV
    V = V + dV
print(f'W=W:.1f} J')
```