Oblig 2 – MAT-INF1100

Rune Hovde (runehovd)

Oppgave 1

a)

Vi finner ut akselerasjonen ved å dele endringen av fart på endring av tid:

$$\frac{\Delta v}{\Delta t} = v' = a$$

b)

For å beregne strekningen s(t) fra startpunktet:

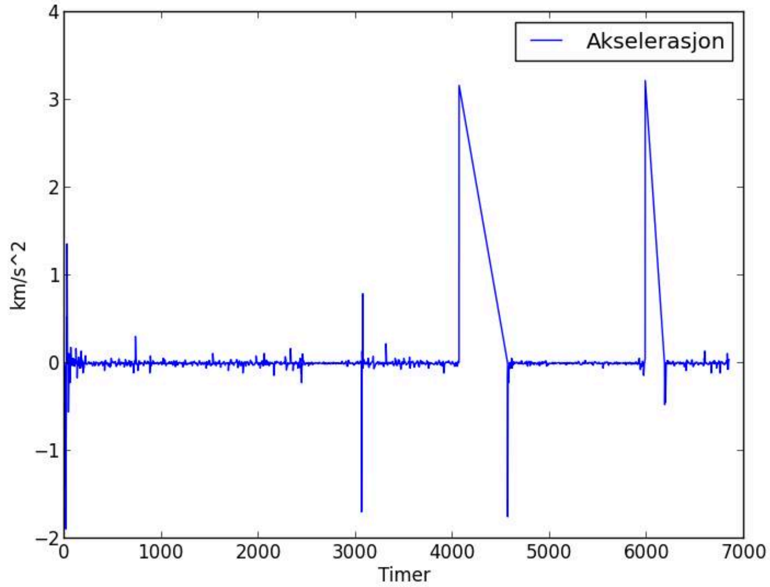
$$S = \Delta t * v$$

Siden vi da har den forrige verdien av s kan vi legge den til for å få neste verdi av s. Dette er gjort i koden.

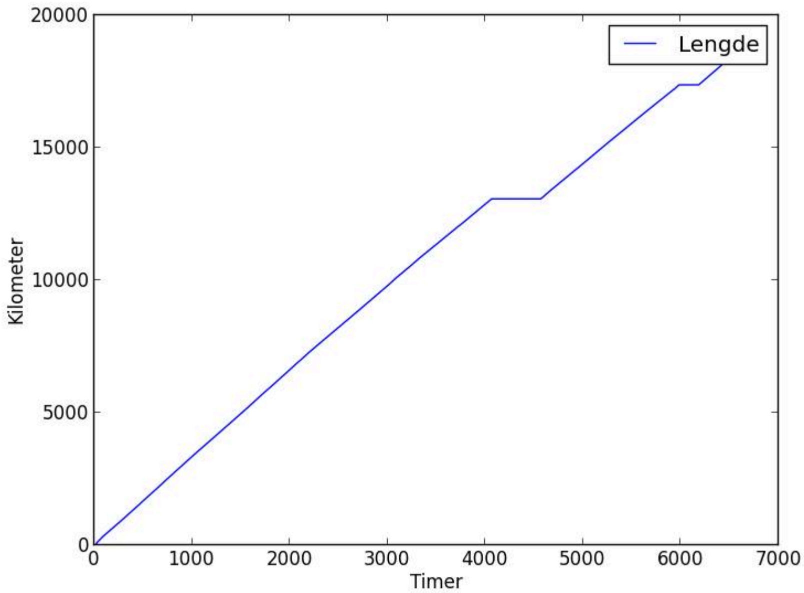
Koden:

```
from matplotlib.pyplot import *
from numpy import *

t = [] #tid
v = [] #fart
innfil = open('running.txt', 'r')
for line in innfil: #leser inn fila
   tneste, vneste = line.strip().split(',')
   t.append(float(tneste))
   v.append(float(vneste))
innfil.close() #lukker fila

a = [] #akselerasjon
for i in range(0, len(v)-1):
   h = t[i-1]-t[i] #definerer delta t
   a.append((v[i+1]-v[i])/h) #Den deriverte av farten. Dette for aa finne akselerasjonen
plot(t[0:-1], a) #tilpasser x-verdiene til y-verdiene for plottingen
legend(['Akselerasjon'])
xlabel('Timer')
ylabel('km/s^2')
show()

s = [] #posisjon
s0 = 0 #startverdi
s.append(s0)
for i in range(0, len(v)-1):
   s.append(s[i]+((t[i+1]-t[i])*v[i+1]))
   #Her bruker jeg formelen for lengde: s = v*t
   #Vi har den forrige lengden, og adderer derfor bare den forrige s med formelen
   #for s for aa faa neste s.
s = array(s); t = array(t)
plot(t,s)
legend(['Lengde'])
xlabel('Timer')
ylabel('Kilometer')
show()
```

Plottene:



Dette er en plott av [a(x)]



Dette er plottet til [v(x)]

Oppgave 2
a)

$$x' + x^2 = 1 \qquad x(0) = 0$$

$$x' = 1 - x^2$$

Ganger med (1-x^2) på begge sider

$$\frac{x'}{(1 - x^2)} = 1$$

$$x \int \frac{1}{1 - x^2} = t + C$$

Dette er arctanh x (stor i forsiden på kalkulus utgave 3)

$x \operatorname{arctanh} x = t$ + C

Ganger med tanh x på begge sider

$$x(t) = \tan(t + C)$$

For å finne ut hva C er bruker vi x(0) = 0

$$x(0) = 0 = \tanh(0 + C)$$
$$\tanh(c) = 0$$
$$c = 0$$
$$\mathrm{x(t)} = \tanh(t)$$

b) og c) på neste side

```python
# b og c
from matplotlib.pyplot import *
from numpy import *

# b)
xt = [] #Eulers metode
xt.append(0) #siden x(0) = 0
h = 0.4 #for aa finne en liten nok step
steps = linspace(0, h, 2)
for n in range(len(steps)-1):
    xt.append(xt[n] + h*(1-xt[n]**2)) #Bruker Eulers metode med f(x) = 1-x**2
eksakt = tanh(steps) #regnet ut for haand
# c)
xdt = []
xdt.append(0)
for i in range(len(steps)-1):
    xdt.append(xdt[i]+(((xdt[i]+(xdt[i]**2+1)*h/2)**2+1)*h))
Midtpunkt = array(xdt) #Bruker Eulers midtpunkt-metode med samme f(x)

plot(steps, eksakt)
plot(steps, xt)
plot(steps, Midtpunkt)
legend([Eksakt, 'Eulers metode', 'Eulers midtpunkt'])
xlabel('x')
ylabel('y')
show()
```

d)

Differensiallikningen (tanh(t)) er alltid stigende mellom 0 og 1. Dette vant jeg ut av ved å observere funksjonen tanh. Dette er en satt funksjon, og vil ikke endre seg.
Tanh vil aldri bli større eller lik 1, som da er dens begrensning, ettersom tanh konvergerer mot 1.

Plott fra oppgave b og c.