# Oppg1

## 1a: Gi en algoritme for å beregne en tilnærming til objektets aksellerasjon a(t) = v 0 (t) ut fra de beregnede verdiene (ti , vi) av farten.

Siden vi har hastighet gitt av tid. Må vi derivere for å finne akselerasjonen. Dette gjøres ved å bruke numerisk derivasjon. Siden vi har data fra punkt data og ikke en kontinuerlig funksjon, bruker vi en steglende gidd ved t\_i – t\_i+1. Da er algoritmen:

def aksellerasjon (x,y):

dy = [(y[i+1] - y[i])/ abs(x[i+1] - x[i]) for i in range(len(y)-1)]

return(dy)

Som da returnerer den deriverte av y, som i vår oppgave er v.

Eksempel:

t = list(range(0,10,2))

v = [2,4,8,12,24]

print(t[1:],aksellerasjon(t,v))

python .\1a.py

[2, 4, 6, 8] [1.0, 2.0, 2.0, 6.0]

## 1b: Gi en algoritme for å beregne en tilnærming til objektets avstand s(t) fra startpunktet ut fra de beregnede verdiene når v(t) = s 0 (t) og s(t0) = 0.

For å finne avstanden fra startpunktet bruker jeg numerisk integrasjon. Som i 1a må jeg bruke integrasjonen over punkter og ikke en funksjon. Det gjør jeg med å regne ut dx for hver element i y. Det gir meg algoritmen:

def avstand(x,y):

Y = [0]

for i in range(1,len(y)):

dx = x[i]-x[i-1]

Y.append(y[i]\*dx + Y[-1])

return Y

Eksempel:

t = list(range(0,10,2))

v = [2,4,8,12,24]

print(t, avstand(t,v))

python .\1b.py

[0, 2, 4, 6, 8] [0, 8, 24, 48, 96]

## 1c: Last ned fila running.txt og kjør denne koden, og bruk algoritmene fra a) og b) til å lage to plott: Ett der du plotter objektets akselerasjon mot tid, og ett der du plotter objektets avstand fra startpunktet mot tid.

from Oppg1a import \*

from Oppg1b import \*

import numpy as np

import matplotlib.pyplot as plt

t = []

v = []

infile = open('running.txt','r')

for line in infile:

tnext, vnext = line.strip().split(',')

t.append(float(tnext))

v.append(float(vnext))

infile.close()

V = np.asarray(avstand(t, v))

dv = np.asarray(aksellerasjon(t, v))

if \_\_name\_\_ == "\_\_main\_\_":

plt.subplot(211)

plt.plot(t[1:],dv,label='aksellerasjon')

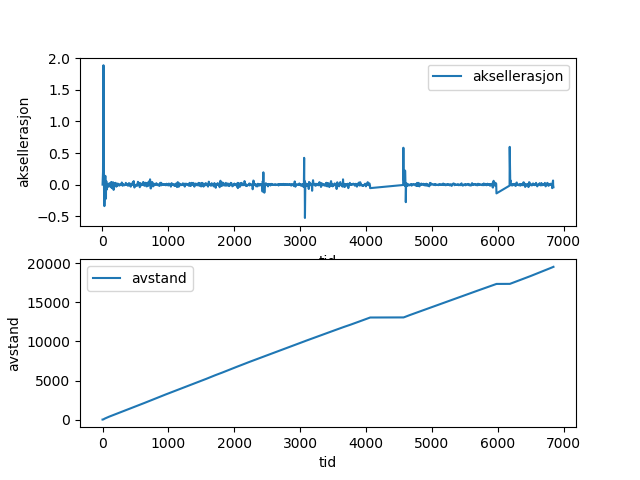
plt.legend()

plt.subplot(212)

plt.plot(t, V, label='avstand')

plt.legend()

plt.show()



# Oppg2

## 2a:

## 2b:

## 2c: Skriv en Python-funksjon lin\_pendel\_euler som med funksjonskallet v, theta = lin\_pendel\_euler(v0, theta0, g, L, N, h).

import numpy as np

import matplotlib.pyplot as plt

def lin\_pendel\_euler(v0, theta0, g, L, N, h=1e-14):

T = h\*N #siden vi har h = T/N

v, theta = [0]\*(N+1), [0]\*(N+1)

v[0], theta[0] = v0, theta0

for k in range(N):

v[k+1] = v[k] - g\*h\*theta[k]

theta[k+1] = theta[k] + h\*(v[k]/L)

return v, theta