

Eksperiment 1: Akselerasjon

Hensikt:

I dette eksperimentet skal dere innhente egne måledata og behandle dem numerisk. Videre skal dere programmere en numerisk metode for integrasjon og anvende denne for å finne relevante størrelser, samt drøfte usikkerheter i egne resultater. Dere også skal konstruere enkle matematiske modeller for et fysisk problem og sammenlikne modellene med egne måledata.

Utstyr:

I dette eksperimentet ønsker vi å hente ut data fra akselerometeret i en mobiltelefon, og det er flere måter dette kan gjøres på. For enkelhetens skyld anbefaler vi at dere benytter dere av en app kalt phyphox som er tilgjengelig i App Store og Google Play. Denne appen lar oss eksportere data fra sensorene i mobiltelefonen til CSV-filer, som vi så kan laste inn i Python eller lignende. Se tillegg A for en kort veiledning i å eksportere til ønsket format.

E1.1

Gitt en heis som står i ro, tegn et frilegemediagram for en person som står inne i heisen. Husk å navngi alle krefter og definere et aksekors.

E1.2

Ved hjelp av akselerometeret i en mobiltelefon, ta en måling av akselerasjonen til mobilen når den ligger i ro (velg "Acceleration with g" i phyphox). Eksporter målingen som en CSV-fil med komma som skilletegn og punktum som desimalpunkt.

E1.3

Skriv et enkelt program som laster inn datafilen produsert i forrige oppgave, og plott akselerasjonen i z-retning som en funksjon av tiden. Et eksempel på hvordan en CSV fil kan leses med pakken CSV i Python er gitt i tillegg B. Diskuter plotet av akselerasjonen.

Beregn den midlere akselerasjonen fra målingen og angi standardavviket¹. Diskuter svaret. Samsvarer svaret med forventningene?

E1.4

La oss nå se på en heis som beveger seg oppover. For en person som står inne i heisen når den beveger seg med konstant fart, tegn et frilegemediagram med alle krefter som virker. Tegn deretter et nytt frilegemediagram for personen når heisen akselererer oppover. Husk å navngi alle krefter og definere et aksekors.

¹Standardavviket er et mål på variasjonen i de målte verdiene, og kan beregnes i python ved hjelp av numpy-funksjonen std(). Når mobilen er i ro bør akselerasjonen være tilnærmet konstant, så en god måling bør ha et standardavvik som er nært 0 m/s². Dersom mobilen var fullstendig i ro under målingen vil standardavviket si noe om usikkerheten i sensoren.

E1.5

Ved hjelp av akselerometeret i en mobiltelefon, ta en måling av akselerasjonen til mobilen når heisen beveger seg fra kjelleren til fjerde etasje i Fysikkbygget. Eksporter målingen som en CSV-fil med komma som skille tegn og punktum som desimalpunkt.

E1.6

Skriv et enkelt program som laster inn datafilen produsert i forrige oppgave. Fra filen plott akselerasjonen i z-retning som en funksjon av tiden. Hva ser du?

E1.7

Vi ønsker nå å finne ut hvor langt personen har beveget seg i løpet av heisturen beskrevet i forrige oppgave. Forklar hvordan den totale avstanden person tilbakelegger i løpet av heisturen kan beregnes.

Hint: Er verdien til tyngdeakselerasjonen vi fant tidligere relevant for avstanden personen tilbakelegger?

E1.8

Implementer løsningsmetoden fra forrige oppgave numerisk og plott farten og posisjonen i z-retning som en funksjon av tid. Hva ble den totale tilbakelagte avstanden?

Diskuter resultatene og sammenlign med den faktiske avstanden, som er blitt målt med lasermåler til å være omtrent 16.6 meter. Ser resultatene ut til å være fornuftige?

Ekstra: Bruk lasermåler til å måle avstanden fra gulvet i kjelleren til gulvet i fjerde etasje selv, og sammenlikn med resultatet fra heiseeksperimentet.

A Hvordan eksportere data fra phyphox

Figur 1 viser hvordan data kan eksporteres fra phyphox til ønsket filformat. Dataene kan sendes til deg selv per e-post, Facebook, AirDrop, etc.

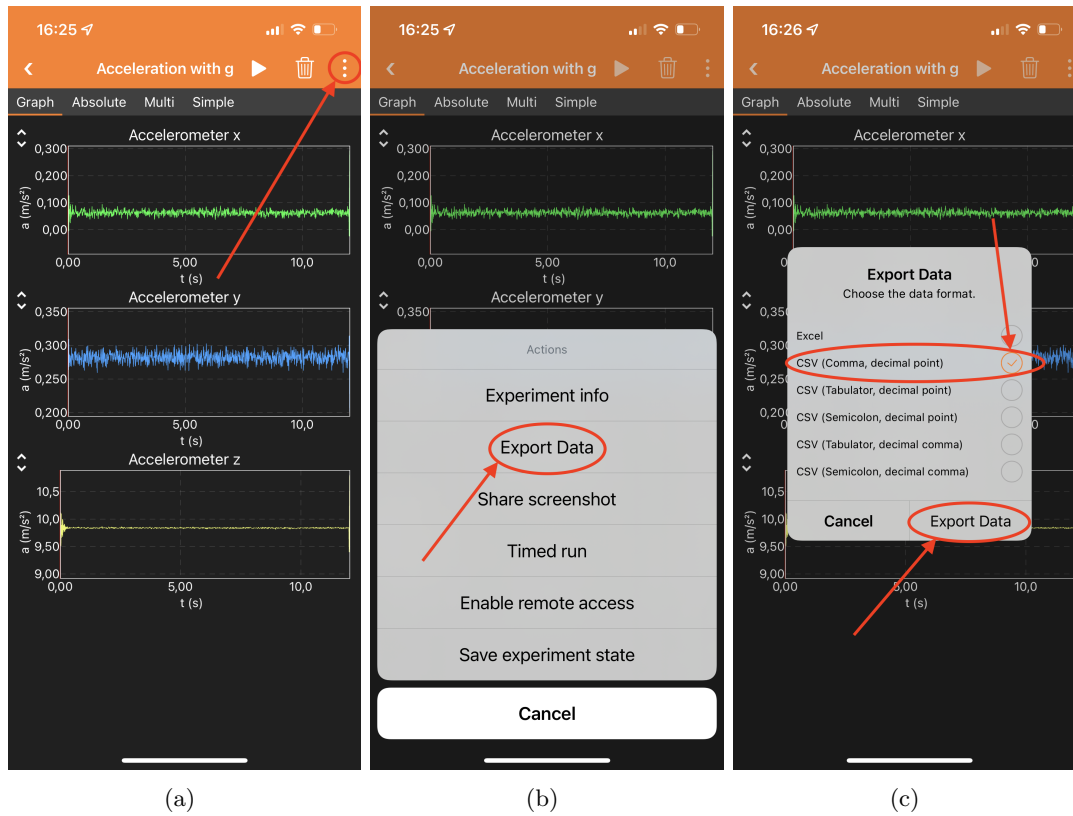


Figure 1: For å eksportere data fra phyphox følg steg a-c, deretter send filen til datamaskinen på ønsket måte.

B Hvordan lese .csv fil til Python

Kodesnippen under viser et eksempel på hvordan en .csv-fil kan leses med Python, og lagres som `numpy`-arrays. Merk at dette forutsetter at det er brukt komma som skilletegn og punktum som desimaltegn, slik som er standard for CSV-formatet.

```
1 import numpy as np
2 import csv
3
4 file = open('kalibrering.csv')
5 csvreader = csv.reader(file)
6
7 header = []
8 header = next(csvreader)
9 t = []
10 az = []
11 a = []
12
13 for row in csvreader:
14     t.append(float(row[0]))
15     az.append(float(row[3]))
16     a.append(float(row[4]))
17 file.close()
18
19 t = np.array(t)
20 az = np.array(az)
21 a = np.array(a)
```