Eksperiment 2: Fall med luftmotstand

Hensikt:

I denne oppgaven skal vi utføre et enkelt eksperiment og behandle måleresultatene numerisk.



Figure 1: Oppsett for måling av posisjonen til en bordtennisball ved et utvalgt punkt i en video.

Utstyr:

Til dette eksperimentet vil vi trenge muffinsformer, en bordtennisball, mobiltelefonkamera og en papirremse med avstandsmarkeringer tegnet på (se figur 1). Bordtennisballer, muffinsformer og papirremser vil være tilgjengelig når eksperimentet skal gjennomføres.

E2.1

Vi ser først på hvordan muffinsformen påvirkes av luftmotstand.
Hold én muffinsform i én hånd og to muffinsformer (en opp i den andre) i den andre hånda, og slipp dem samtidig. Finn ut hvilket forhold det må være mellom starthøydene for at de skal treffe bakken samtidig. Prøv gjerne også med tre eller flere muffinsformer inni hverandre.

Bruk resultatene til å finne ut hvordan luftmotstandskraften på en muffinsform avhenger av farten til muffinsformen, dvs. vurder om luftmotstandskraften er proporsjonal med farten, $F \propto v$, eller med kvadratet av farten, $F \propto v^2$.

Det er ikke tilstrekelig å tippe hva som er rett form på luftmotstanden, det skal tydelig vises og begrunnes ut fra eksperimentene.

E2.2

Nå skal vi måle luftmotstanden på en bordtennisball.
Ved hjelp av kameraet på mobiltelefonen, film bordtennisballen mens den slippes langs papirremsen. Papirremsen har markeringer, og det er 10 cm mellom hver markering. Overfør videoen til en datamaskin for videre analyse.
Merk at en høyere bildefrekvens (engelsk: fps = frames per second) vil være hensiktsmessig. Ved bruk av høyere bildefrekvens vil ballen som slippes bli mindre "strukket ut" på bildet, og det vil være lettere å få en nøyaktig måling av ballens posisjon. Uansett er det nødvendig å vite kameraets bildefrekvens, slik at man kan konvertere mellom enhetene "frames" og sekunder i påfølgende analyser.

E2.3

Når vi nå skal analysere videoen ønsker vi å kunne bla igjennom bilde for bilde for å notere posisjonen til bordtennisballen som en funksjon av tid. Det er flere måter å konvertere en video til bilder, se tillegg A. For hvert bilde (evt hopp over noen bilder dersom det er veldig mange bilder pga høy bilderate), finn posisjonen til ballen. Diskuter hvordan gjøre det på en mest presis måte, og anslå usikkerheten.

E2.4

Konverter måledataene fra forrige punkt til SI-enheter og plott resultatet sammen med løsningen for fritt fall uten luftmotstand (som du kan regne ut enten analytisk eller numerisk etter som du selv ønsker). Diskuter plottet.

E2.5

Diskuter kort kilder til usikkerheter i målingene og gi et estimat for usikkerheten i posisjon og tid.

A Konvertering av videofil til enkeltbilder

Det er flere måter å konvertere en film til enkletbilder. Hvis man skal gjøre det mange ganger er det klart en fordel å kunne gjøre det direkte på egen pc med en kommando enten i kommandovindu eller i python (eller et annet språk). Vi har erfart at dette ofte er litt krevende å få til å virke, fordi ulike operativsystemer og python-installasjoner ikke fungerer likt og vi ikke vet hvordan det skal gjøres i mange tilfeller. Derfor har vi også funnet ut at man kan gjøre det ganske enkelt med nettbaserte konverteringssider. Det finnes flere av disse, vi har testet denne: https://mconverter.eu/convert/mp4/jpg/ men det kan godt være at andre er minst like gode (men vi har også funnet en, https://image.online-convert.com/ som dropper noen av bildene, slik at vi ikke vet hvor mye tid som går mellom hvert bilde. Dette er ubrukelig for oss) . Dere må da laste opp videofila, konvertere og så laste ned alle bildene. Det tar litt tid, men for en enkelt gang er det helt greit. For de som har lyst til å prøve å gjøre det på egen pc er det er par alternativer under. Når de virker, går disse fortere og det er

A.1 Konvertering med `opencv-python`

```
1  try:
2      import cv2
3  except ImportError as e:
4      import subprocess
5      subprocess.call(['python', '-m', 'pip', 'install', 'opencv-python']) # bytt fra ↩
           python til python3 hvis dette ikke virker
6      import cv2
7
8  cap = cv2.VideoCapture('bordtennisball.mov')
9  fps = cap.get(cv2.CAP_PROP_FPS)
10 print (f'fps: {fps}')
11
12 count = 0
13 while True:
14     ret,frame = cap.read()
15     if ret:
16         cv2.imwrite(f'frame{count}.jpg', frame)
17         count = count + 1
18     else:
19         break
20 cap.release()
21 cv2.destroyAllWindows() # destroy all opened windows
```

Ved å endre tekststrengen i linje 8 kan du angi navnet på videoen du ønsker å konvertere. Merk at videoen må være i samme mappe som programmet, eller at rett filplassering legges til i linje 8.

Dette skriptet vil generere en stor mengde bildefiler i nummerert rekkefølge. Ved å markere alle bildene og så åpne dem, så skal alle bildene kunne åpnes i ett vindu slik at man kan bla igjennom dem manuelt. Videre så undersøker koden på linjene 1-6 om den nødvendige pakken er installert, dersom den ikke er det så vil den forsøke å installere den.

A.2 Alternativ: Jupyter Notebook

Sammen med koden over, ligger det på emnesiden en Jupyter Notebook som fungerer på samme måte. Dersom metoden over ikke fungerer på din maskin kan en Jupyter Notebook

være et alternativ. For å minimere arbeidet og for at det skal fungere likt for alle, så anbefaler vi at den kjøres på en UiO-server (`jupyterhub.uio.no`). Se figur 2 for et eksempel på hvordan dette gjøres, bildene som produseres kan da lastes ned.

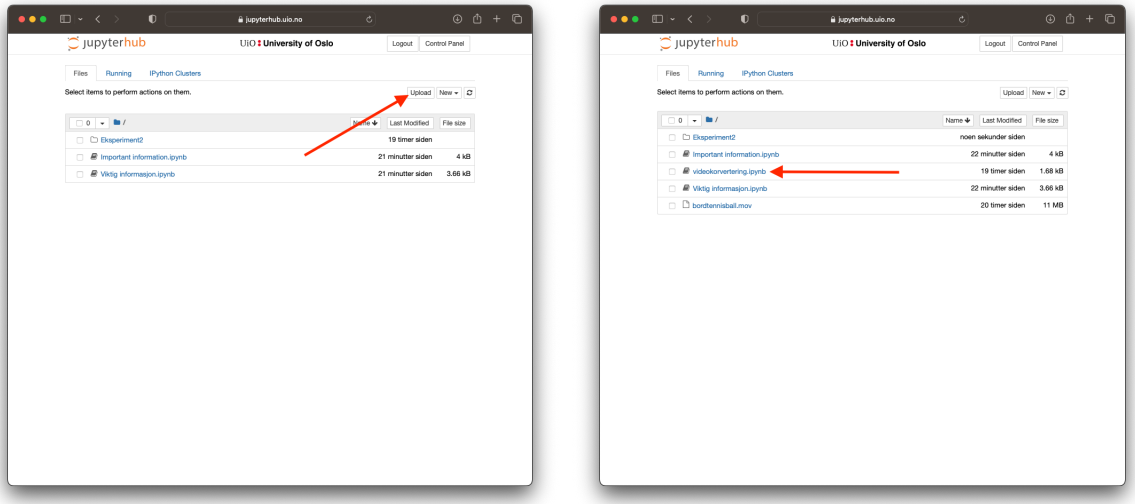
B Konvertering med `FFmpeg`

Se lenke over for nedlastningsguide for ditt operativsystem.

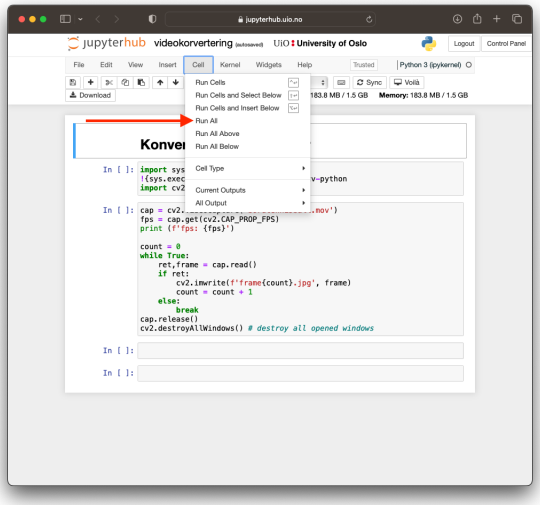
Ved hjelp av terminalen, naviger deg til mappen der hvor du har plassert videoen. For å konvertere videoen kjør følgende kjør følgende kommando i terminalen

```
ffmpeg -i filnavn.mp4 -qscale:v 0 thumb%04d.jpg
```

Dette vil konvertere videoen til enkeltbilder i `.jpg`-format. Ved å markere alle bildene og så åpne dem så skal alle bildene kunne åpnes i et vindu slik at man kan bla igjennom dem manuelt.



(a) (b)



(c)

Figure 2: Logg på `jupyterhub.uio.no`, last opp Nootebook og videofil (a). Åpne Nootebook ved å klikke på den(b), kjør så alle celler(c).