

HALVERINGSTID

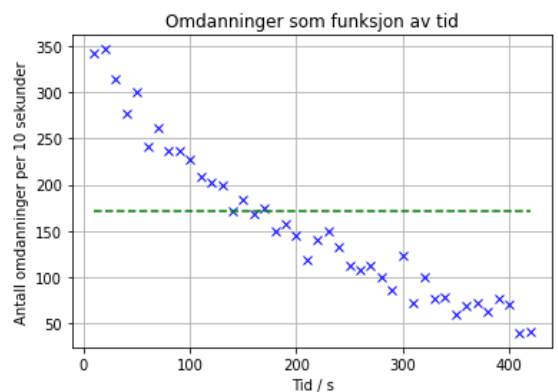
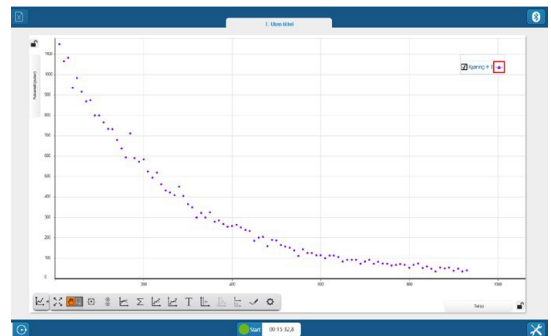
I dette forsøket skal vi måle radioaktiv stråling med en geigerteller. Vi gjør målinger på det radioaktive isotopen Ba-137 og ønsker å finne isotopens halveringstid. Dataene samles inn med Capstone eller SPARKvue m/nødvendig sensorikk, og dataene lastes over i Python for numerisk analyse.

Fest geigertelleren til stativet og koble sensoren til PC eller nettbrettet for innsamling av data. Husk å ta av evt. beskyttelse på geigertelleren. Gjør klar isotopgeneratoren ved å trekke opp ca. 5 ml saltløsning i sprøyta. Ta av beskyttelseshettene på generatoren (oppe og nede) og plasser sprøyta ned i generatoren. Ikke fyll metallkoppene før du er klar til å starte målingen. Når dataene er samlet, eksporteres de som en .txt-fil og lastes inn i Python.



Utstyr

Trådløs GM-teller (PS-2166S),
Isotopgenerator (108065) og
diverse stativmaterieill



Koden gir svaret på to måter, ved plott og med utskrift. Utskriften til skjermen kan f.eks. se slik ut:

```
Antall omdanninger start: 342.0
Antall omdanninger halveringspunkt: 171.0
Beste målepunkt: 171.0
Differanse fra halveringspunkt: 0.0
Halveringstid 130.0
```

```
# Deklarerer liste for lagring av data
# NB Deler av koden kortet ned for lesbarhet
tid_data = []
omdanninger_data = []
with open("Halveringstid.csv") as file:
    for line in file:
        # Lagrer tidsdataene
        tid_data.append(float(line[1]))
        # Lagrer registreringer
        omdanninger_data.append(float(line[2]))
```

I Python-koden som følger eksperimentet er det gjort to tilnærminger for å finne halveringstiden, én enkel og én avansert. Du velger den du synes passer best til din undervisning. Kun den enkle versjonen er gjengitt her, mens den avanserte er tilgjengelig i kodearkivet.

```
# Finn første måleverdi fra geigertelleren
N0 = omdanninger_data[0]
# Beregn halve verdien av oppstartsmålingen
halveringspunkt = N0/2
i = 1 # Let etter første data som er lavere
# i verdi enn halveringspunktet
while omdanninger_data[i] > halveringspunkt:
    i += 1 # Oppdater teller
# Løkken avslutter for første målte verdi som er
# lavere enn halveringspunktet. Tidspunkt og
# måleverdi lagres
halveringstid = tid_data[i]
beste_måling = omdanninger_data[i]
# Halveringstid beregnes som tiden siden oppstart
halveringstid = halveringstid - tid_data[0]
```