

TMA4101 Oblig – Benjamin Herlofson Nilssen, Maria Thuy-an Truc Tran, Hannah Aspaas, Hannah Marie Levorsen

I dette forsøket tenkte vi å slå to fluer i en smekk ved å bruke Newtons avkjølingslov til å undersøke nedkjølingen av boller – og samtidig mette en veldig hangry Hannah. (Hun blir faktisk voldelig)

«**Newton's law of cooling** is a physical law which states that the rate of heat loss of a body is directly proportional to the difference in the temperatures between the body and its environment.» (*How many laws does this guy have? Is he a lawyer or something?*)

Newton's avkjølingslov kan uttrykkes som:

$$T'(t) = \alpha (T(t) - T(K)), T(0) = T_0$$

Løst for $T(t)$:

$$T(t) = T_k + (T_0 - T_k)e^{-\alpha t}$$

- $T(t)$ vil gi temperaturen til bollen ved tid t
- $T(k)$ vil tilsvare romtemperaturen i en 16-manns hybel på moholt (så hakket klammere enn i leiligheten min på frogner)
- $T(0)$ tilsvarer temperatur ved start
- α (alpha) «inneholder informasjon om varmekapasiteten til elgtungen og hvor fort varmeflyten går mellom elgtungen og omgivelsene.», (glemte jeg å nevne elgtungen?)

I vårt tilfelle:

$$T(k) = 25.2 \text{ C}$$

$$T(0) = 74.3 \text{ C}$$

Hannah hadde kjøpt termometeret på jysk (?) av alle steder så den var ikke helt som andre termometer, men hadde heller litt personlighet i form av humørsvingninger (temperatursvingninger) som ga oss litt usikkerhet i målingene samt en migrrene.

Vi brmmf... bruker det følgmff...ende progrmff...rammet for å plotte temm...peraturen ved forskjellige tidsppunker og samm...mmenligne den teoretiske modellen fra Newtons avkjølingslov... nomf... ("Jeg spmmf...spiser bmmff...boller... nomf..."):

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import curve_fit

# Omgivelsestemperatur
Tk = # f.eks. romtemperatur i grader Celsius

# Målte data (erstatt med dine egne målinger)
tid = np.array([, , , , ]) # tid i minutter
temperatur = np.array([, , , , ]) # eksempelverdier i grader Celsius

# Modellfunksjonen for Newtons avkjølingslov
def newtons_cooling_law(t, T0, alpha):
    return Tk + (T0 - Tk) * np.exp(-alpha * t)

# Finn den beste tilpasningen for \(\ T0 \) og \(\ \alpha \)
popt, _ = curve_fit(newtons_cooling_law, tid, temperatur, p0=[temperatur[0], 0.1])
T0_fit, alpha_fit = popt

# Plott måledata og modell
tid_plot = np.linspace(0, max(tid), 100)
temperatur_teoretisk = newtons_cooling_law(tid_plot, T0_fit, alpha_fit)

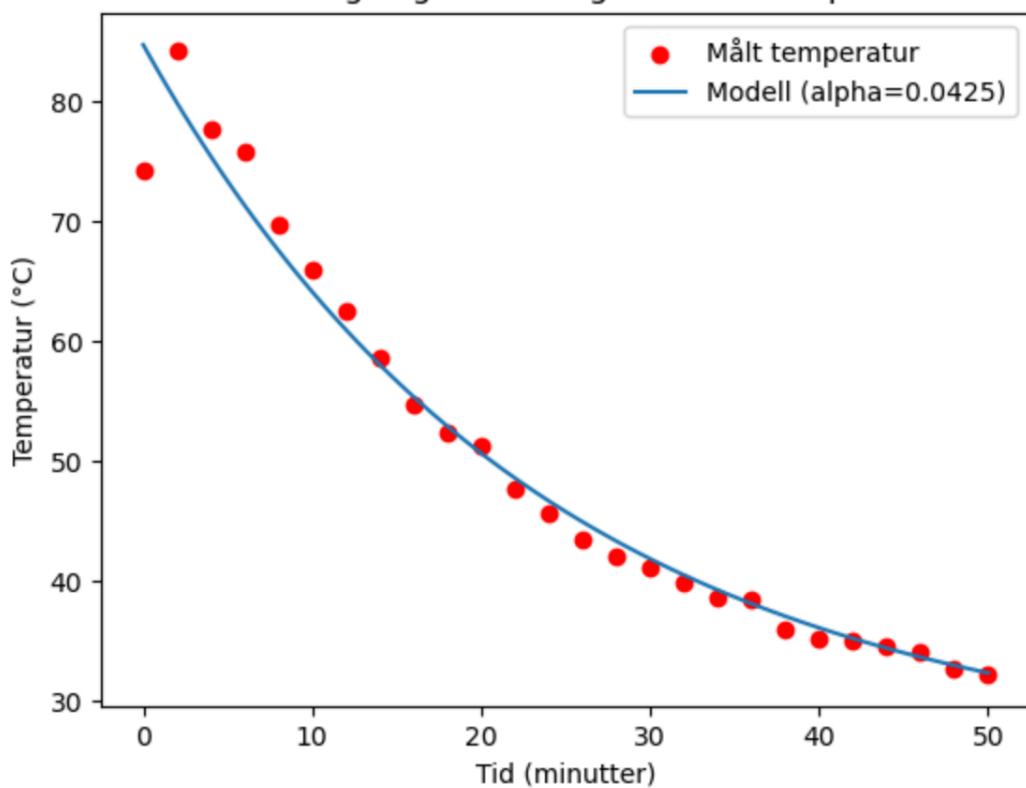
plt.figure()
plt.scatter(tid, temperatur, color='red', label='Målt temperatur')
plt.plot(tid_plot, temperatur_teoretisk, label=f'Modell (\alpha={alpha_fit:.4f})')
plt.xlabel('Tid (minutter)')
plt.ylabel('Temperatur (°C)')
plt.title('Sammenligning av målt og teoretisk temperatur')
plt.legend()
plt.show()
```

Vi måtte kjøre et parallelt forsøk fordi Hannah spiste den første bollen (jeg sa jo at hun var hangry):



Etter 50 minutter av å skvette hver gang alarmen gikk hvert andre minutt for å så stikke ett hull i en bolle (det ble mange dårlige vitser her), så var vi klare til å sette inn verdiene i programmet og fikk følgende plott:

Sammenligning av målt og teoretisk temperatur



Oppsummering:

Vi ser på grafen at våre målinger er nærmere Newtons beregninger. Dette tyder på at teorien stemmer. Vi ser også at den første målingen vår var litt feil, dette er helt sikkert en feil med termometret (aldri vår feil).

Vi elsker matte og boller, takk for oss! :)