~~Elgtunge~~ VÅRRULL – Newtons avkjølingslov

Hensikt:

Hensikten med forsøket er å undersøke og analysere hvor godt Newtons avkjølingslov beskriver temperaturendringen i de valgte gjenstandene når de kjøles ned i romtemperatur. Vi skal også bestemme proporsjonalitetskonstanten α avhengig av gjenstandenes egenskaper og nedkjølingsforholdene.

Teori:

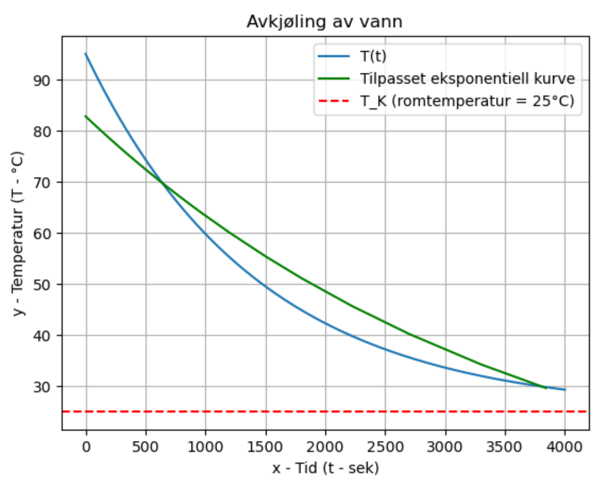
Newtons avkjølingslov: $\dot{T}(t) = \alpha(T(t) - T_K)$ $\qquad T(0) = T_0$

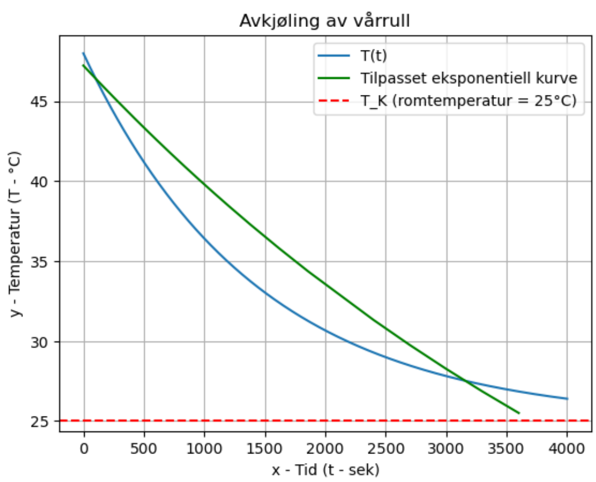
Modellen beskriver hvordan temperaturen til et objekt endrer seg over tid når den utsettes for et annen temperatur ($T_K \ = romtemperateratur = 25^{\circ}C$). Loven sier at avkjølingshastigheten α er proporsjonal med temperaturforskjellen mellom objektet og omgivelsene: $\alpha * (T(t) - T_K)$

I praksis vil dette gi et større varmetap dersom $T(t) - T_K$ er høy. Når temperaturen nærmer seg T_K avtar avkjølingen, som fører til en eksponentiell nedgang i temperaturen.

Resultat:

Grafene viser sammenligning av newtons avkjølingslov og eksperimentell data vist gjennom regresjon. Vi har testet ulike proporsjonalitetskonstanter ut ifra gjenstandene og nedkjølingsforholdene, og fant ut at $\alpha \ = -0.007$ ga best resultat.





Drøfting:

Newtons avkjølingslov tar ikke høyde for om det fordampning eller ulike materialer med forskjellig varmekapasitet. Måleinstrument har også innvirkning på resultatet, både på grunn av dets nøyaktighet og avlesningsfeil. Dette ser vi i modellen, hvor man ser at Newtons avkjøingslov gir en slakkere kurve enn realiteten.

For avkjøling av vårrullen er den tilpassende eksponentielle grafen får ikke med den bråe starten, som følger en eksponentiell funksjon. Årsaken til dette er at den raske temperaturendringer i starten ikke ble tatt med, fordi vårrullens temperatur i $T(0) = T_0 = 200^{\circ}C$, noe termometeret ikke kunne måle. Vi spiste og startet derfor målingene for sent. Sammenligner man kurvene for vann, ser man en bedre tilnærming.

Pythonkode for grafene

Punktene under regresjon er utførte målinger.

```python
#Vann

import numpy as np
import matplotlib.pyplot as plt

import scipy
from scipy.integrate import odeint

# Definer konstanter
alpha = -0.0007    # Velg en passende verdi for alfa
T_K = 25       # Konstant temperaturverdi, kan være romtemperatur
T_0 = 95      # Startverdi for T(0)

# Definer differensiallikningen dT/dt = alpha * (T - T_K)
def dT_dt(T, t):
    return alpha * (T - T_K)

# Tidsskala for simulering
t = np.linspace(0, 4000, 1000)  # Fra 0 til 50 sekunder, med 500 punkter

# Løs differensiallikningen med initialbetingelsen T(0) = T_0
T = odeint(dT_dt, T_0, t)


# REGRESJON
x = [0, 40, 70, 100, 130, 160, 190, 220, 270, 300, 330, 360, 420, 480, 540, 600, 660, 720, 780, 840, 900, 960, 1200, 1500, 1800, 2220, 2700, 3300, 3840]
y = [95, 91, 89, 87, 85, 83, 82, 81, 78, 77, 76, 75, 73, 71, 69, 67, 65.5, 64.5, 63, 62, 60.5, 59.5, 55, 51, 44.5, 44, 40, 37, 35]

# Ta den naturlige logaritmen av y
log_y = np.log(y)

# Utfør lineær regresjon på (x, log_y)
reg = np.polyfit(x, log_y, 1)
a = np.exp(reg[1])
b = reg[0]

# Definer den eksponentielle funksjonen
def exp_func(x):
    return a * np.exp(b * x)

# Plot løsningen
plt.plot(t, T, label="T(t)")
plt.plot(x, exp_func(np.array(x)), label='Tilpasset eksponentiell kurve', color='green')
plt.axhline(T_K, color='r', linestyle='--', label="T_K (romtemperatur = 25°C)")
plt.xlabel("x - Tid (t - sek)")
plt.ylabel("y - Temperatur (T - °C)")
plt.legend()
plt.title("Avkjøling av vann")
plt.grid()
plt.show()
```

```python
#Vårrull

import numpy as np
import matplotlib.pyplot as plt

import scipy
from scipy.integrate import odeint

# Definer konstanter
alpha = -0.0007    # Velg en passende verdi for alfa
T_K = 25       # Konstant temperaturverdi, kan være romtemperatur
T_0 = 48      # Startverdi for T(0)

# Definer differensiallikningen dT/dt = alpha * (T - T_K)
def dT_dt(T, t):
    return alpha * (T - T_K)

# Tidsskala for simulering
t = np.linspace(0, 4000, 1000)  # Fra 0 til 50 sekunder, med 500 punkter

# Løs differensiallikningen med initialbetingelsen T(0) = T_0
T = odeint(dT_dt, T_0, t)



# REGRESJON
x = [0, 15, 60, 120, 180, 240, 300, 420, 480, 540, 600, 720, 840, 960, 1080, 1200, 1380, 1620, 1860, 2400, 2700, 3000, 3300, 3600]
y = [48, 46, 45, 45, 44.5, 44, 43, 48, 47, 45.5, 45, 43, 40.5, 40, 38.5, 37.5, 37, 35, 33, 30, 29, 28.5, 28, 26]


# Ta den naturlige logaritmen av y
log_y = np.log(y)

# Utfør lineær regresjon på (x, log_y)
reg = np.polyfit(x, log_y, 1)
a = np.exp(reg[1])
b = reg[0]

# Definer den eksponentielle funksjonen
def exp_func(x):
    return a * np.exp(b * x)

# Plot løsningen
plt.plot(t, T, label="T(t)")
plt.plot(x, exp_func(np.array(x)), label='Tilpasset eksponentiell kurve', color='green')
plt.axhline(T_K, color='r', linestyle='--', label="T_K (romtemperatur = 25°C)")
plt.xlabel("x - Tid (t - sek)")
plt.ylabel("y - Temperatur (T - °C)")
plt.legend()
plt.title("Avkjøling av vårrull")
plt.grid()
plt.show()
```