Øving tma4101 høst24

Håvard Mykland

November 2024

Figur 1: Forsøkslegemer

Innhold

1 Bakeprosess **3**
1.1 Hvorfor boller? 3
1.2 Hvordan boller? 3

2 Forsøket **4**
2.1 Innsamliing data 4
2.2 Algebra 4

3 Plot **5**

4 Konklusjon **6**

5 Python kode **7**

1 Bakeprosess

1.1 Hvorfor boller?

I dette forsøket har jeg og noen kompiser undersøkt hvor godt newtons avkjølingslov spår avkjølingen til et valgt legeme. At dette legemet tilfeldigvis ble kanelsnurrer er da slettes ikke så tilfeldig i det hele tatt, da kanesnurrer er svært gode å inta.

1.2 Hvordan boller?

Man tar mel og gjær og alt det der, lar det lige en stund, romtemperert og voila, bolledeig.

Videre flater man ut deigen smører litt kanel, sukker og smør og ruller. Setter på ovnen og når man så varmer bolledeigen din som du har rullet sammen så vips, litt magi og kanelboller.

Figur 2: Feite, gode kanelsnurrer

2 Forsøket

2.1 Innsamliing data

Vi startet forsøket inni ovnen. Vi satt et steketermometer inni en av bollene mens den fortsatt var i ovnen og slik fikk vi vår T_0 Videre lot vi termometeret stå i og startet en stoppeklokke strakt bollen var ute av ovnen og leste av temperaturer og noterte tiden.

Fra disse målingene fant vi ved $t = tid\ i\ minutter$ at:

$$T_k = 23^oC\ ,\ T(0) = T_0 = 105^oC\ \&\ T(3) = T_1 = 93^oC \tag{1}$$

2.2 Algebra

Ved bruk av newtons avkjølingslov fant vi et utrykk for $T(t)$, vi vet også at $C = T(0) - T_k$

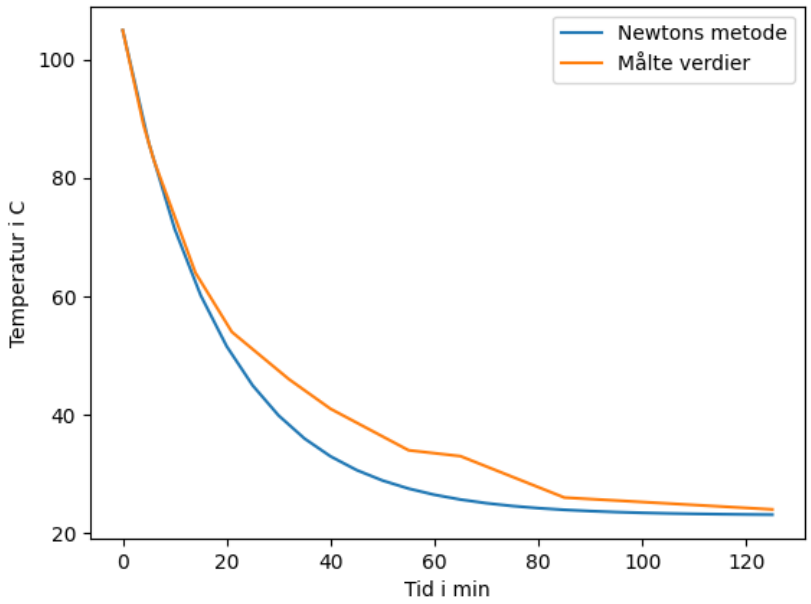
$$\begin{aligned}
\dot{T(t)} &= \alpha(T_k - T(t)), \quad T(0) = T_0 \\
\dot{T} + \alpha T &= \alpha T_k \quad | \cdot e^{\alpha t} \\
(\dot{T} + \alpha T)e^{\alpha t} &= \alpha T_k \cdot e^{\alpha t} \\
\frac{d}{dt} T e^{\alpha t} &= \alpha T_k \cdot e^{\alpha t} \quad | \int dt \\
T e^{\alpha t} + C_1 &= \alpha T_k \cdot \frac{1}{\alpha} e^{\alpha t} + C_2 \\
T e^{\alpha t} &= T_k \cdot e^{\alpha t} + C \quad | \cdot e^{-\alpha t} \\
T(t) &= C \cdot e^{-\alpha t} + T_k \\
T(t) &= (T(0) - T_k) \cdot e^{-\alpha t} + T_k
\end{aligned} \tag{2}$$

Om vi så løser likningen for T_1 for vi et uttrykk for α vi kan bruke for plotingen vår:

$$\begin{aligned}
T(3) = T_1 &= (T_0 - T_k) \cdot e^{-\alpha 3} + T_k \\
\frac{T_1 - T_k}{T_0 - T_k} &= e^{-3\alpha} \quad | \ ln \\
-3\alpha &= ln \left| \frac{T_1 - T_k}{T_0 - T_k} \right| \quad | \cdot -\frac{1}{3} \\
\alpha &= ln \left| \frac{T_1 - T_k}{T_0 - T_k} \right| \cdot -\frac{1}{3} \\
&*python + magi* \\
\alpha &= 0,0527
\end{aligned} \tag{3}$$

3 Plot

Vi fant alle de nødvendige variablene og utrykkene for plotting tidligere og bruker de så i plot:



Figur 3: Newton vs. virkeligheten

4 Konklusjon

Fra plotet kan vi se at newtons metode ikke samvarer helt med virkeligheten, men den gir oss et relativt bra estimat i dette tilfelle.
Fra dette kan vi konkluderer med at newtons avkjølingslov ikke inkluderer alle påvirkene parametere i avkjøling. Eksempler på parametere newtons avkjølingslov er:

- Fordampning
- Varierende varmekonstant α
- Vind

Konkluderende kan vi si at newtons metode er bra nok for å anta når boller er omtrent ferdig nedkjølt, men den er ikke god nok til å forutsi til å kunne brukes til prosesser som er svært avhengig av nøyaktige temperaturer til bestemte tidspunkt.

5 Python kode

```python
import numpy as np
import matplotlib.pyplot as plt

# definerer viktige variabler funnet ved målinger
T0 = 105
Tk = 23
alpha = -0.0527  # funnet ved å løse likningen for T(3) i forhold til målte verdier
steg = 5
i = 0
numeriskTemp = []
numeriskTid = []


# definerer funksjonen for temperaturen til bolla
def T(t):
    return (T0 - Tk) * np.e ** (alpha * t) + Tk


# while løkke som kjører til temperaturen er nådd Tk (romtemp)
while (
    T(i) > Tk + 0.1
):  # kjører ikke helt frem til T(i) = Tk av den årsak at det ville tatt veldig lang tid
    numeriskTemp.append(T(i))
    numeriskTid.append(i)
    i += steg

# definerere de faktiske verdiene som ble målt
faktiskTemp = [105, 93, 89, 83, 64, 54, 46, 41, 34, 33, 24]
faktiskTid = [0, 3, 4, 6, 14, 21, 32, 40, 55, 65, 120]

# ploter
plt.plot(numeriskTid, numeriskTemp, label="Newtons metode")
plt.plot(faktiskTid, faktiskTemp, label="Målte verdier")
plt.ylabel("Temperatur i C")
plt.xlabel("Tid i min")
plt.legend()
plt.savefig("kanelbolleplot")
plt.show()
```