Obligatorisk innlevering

TMA4101 – H2024

Cathrine Grønbech, Rebecca C. Kransberg, Hedda Krogh-Nilsen, Emma Aune Helmersen

Eksperiment

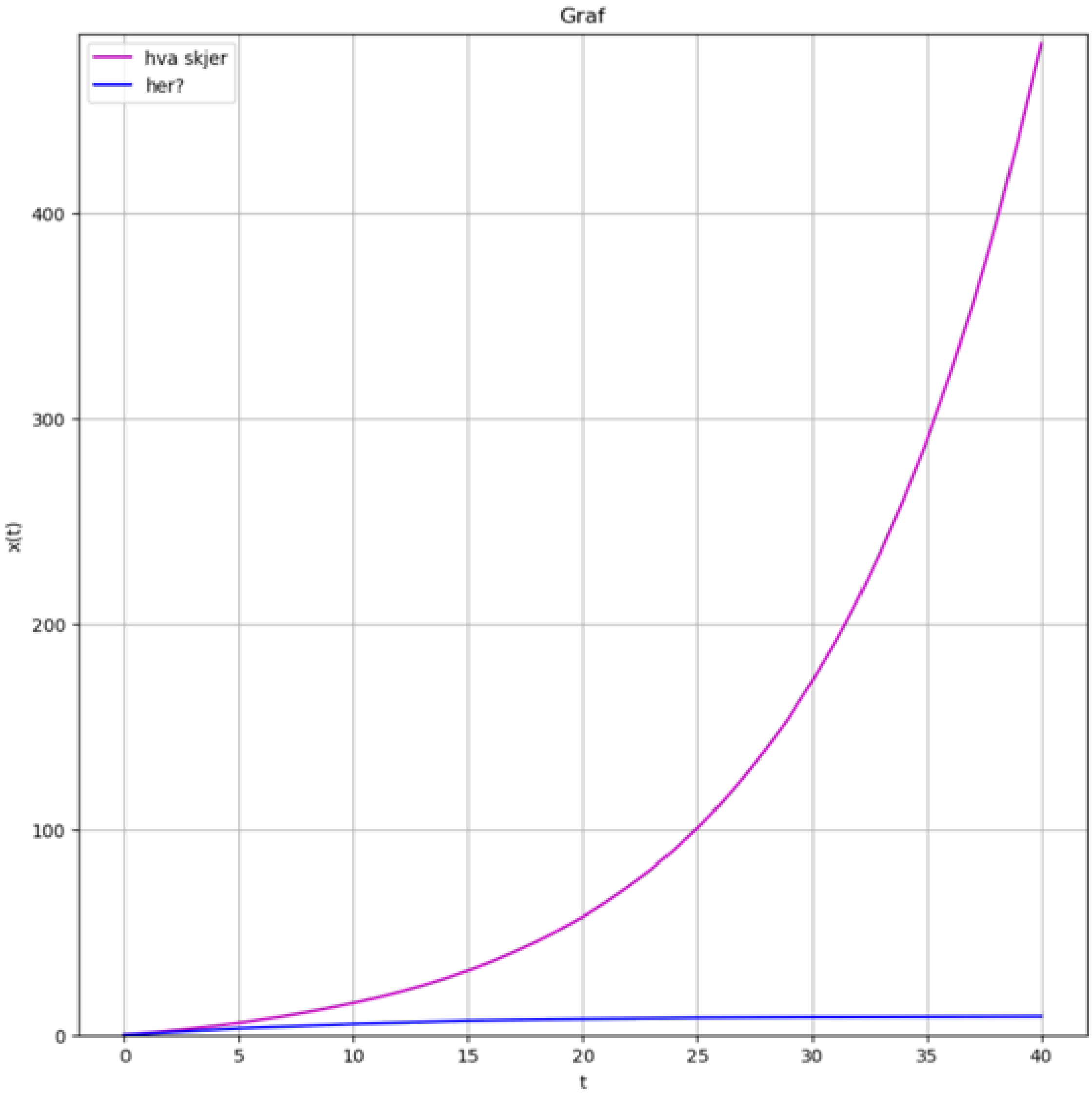
Oppdrag: Sammenlikne målte og beregnede verdier for spenningen over en kondensator under oppladning over 40 sekunder.

Utførelse:

Vi satte opp en RC-krets, der motstanden $R = 100k\Omega$, videre kalt Gunnar Sønsteby. Kondensatoren kalles for Kurt Kåre. Spenningen over Kurt Kåre ble målt med 5 sekunders intervall.

Vi prøvde først med Gunnar Sønsteby lik $3{,}3k\Omega$, men denne grafen ble veldig rar. Med manuelle målinger ble Kurt Kåre ladet opp for raskt. Vi trengte litt mer tid for å ta inn over oss informasjonen, så vi la inn litt mer motstand i kretsen. Gunnar Sønsteby ble oppgradert til $100k\Omega$.

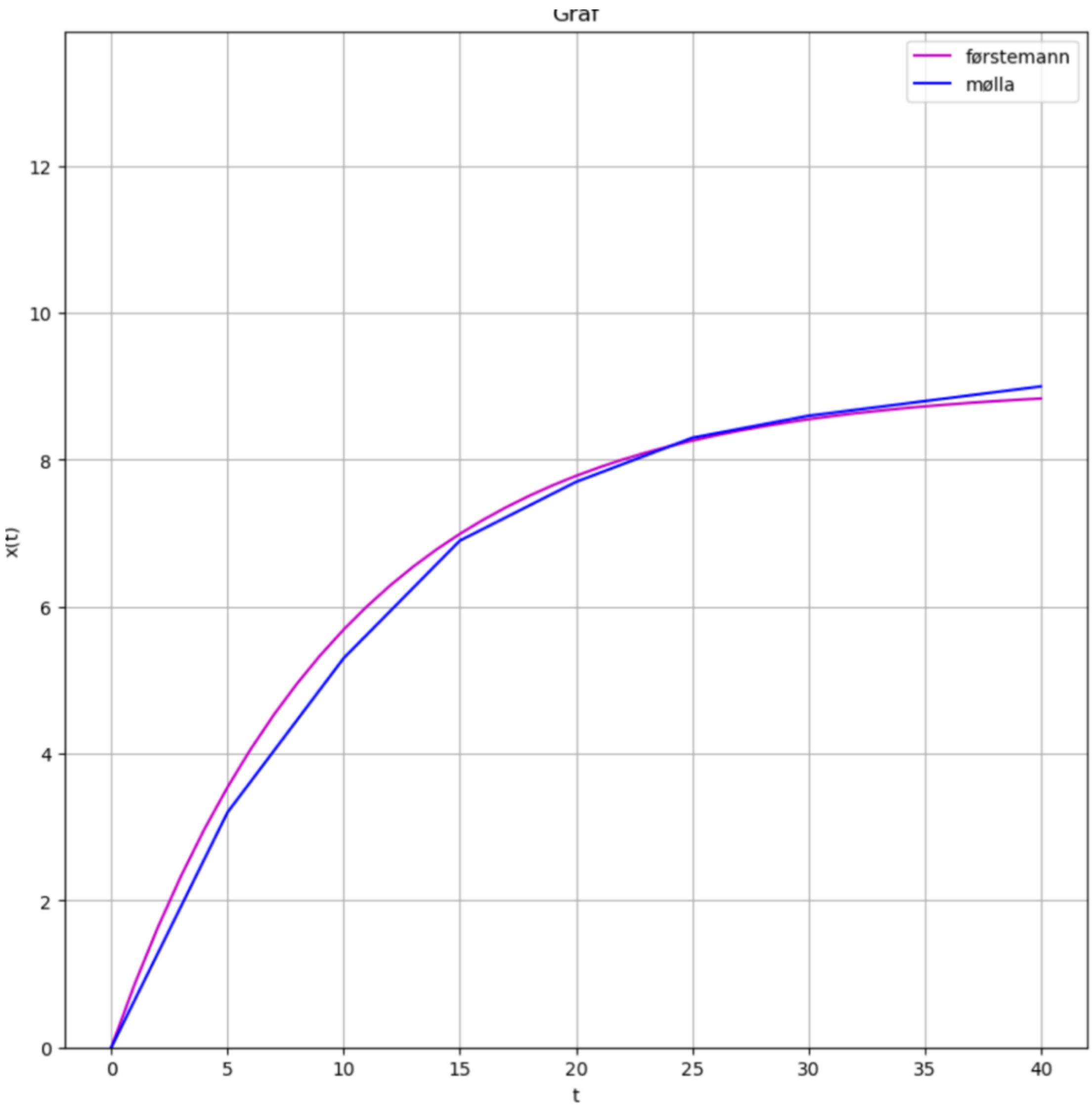
Den første grafen vi fikk var et sant mareritt, her markert med ‘hva skjer’ og ‘her’:



Figur 1

Det som hadde skjedd var Emma, og hennes fantastiske evne til å overse fortegn (et vedvarende problem, bare spør mattelærer fra 1.-4.: Marit S. ved Nyborg barneskole).

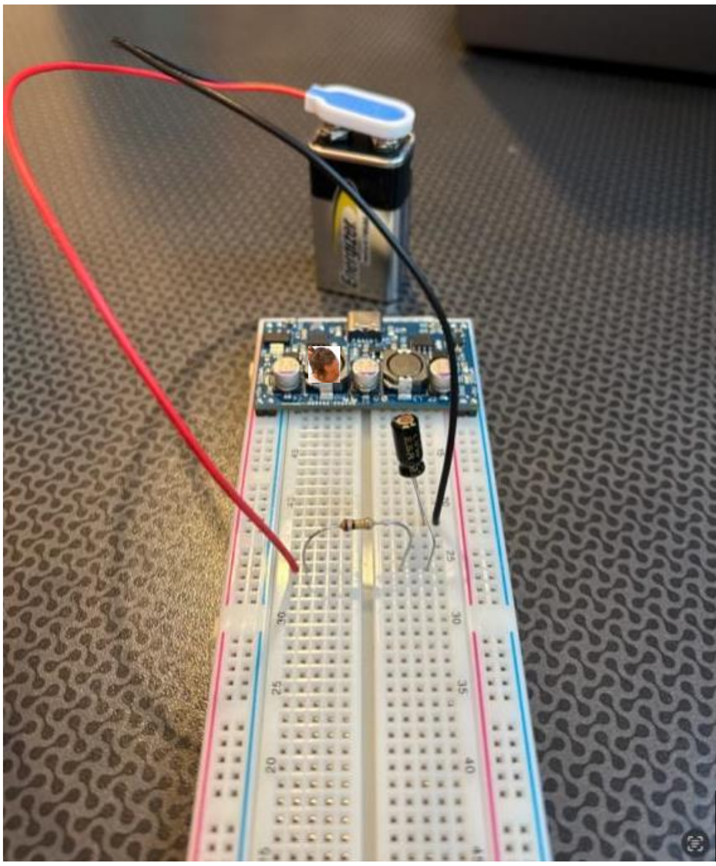
Litt **feilsøking** senere førte til oppdagelse av feilen, og vi fikk grafer som faktisk ga mening: ‘førstemann: og ‘mølla’:



Figur 2

Her ser vi at den målte spenningen (mølla) over Kurt Kåre er litt lavere enn i en ideell verden (førstemann). Det kan forklares med at Gunnar Sønsteby opptrer i ulike versjoner i den virkelige verden. Det er jo helt naturlig og noe vi alle gjør. Knut Kåre er veldig spent om dagen, så han klarer å ta opp litt mer spenning enn det han får tilført i den virkelige verden. Eller så er det kanskje vi som må ta oss en tur til Specsavers for å få sjekket synet. Kanskje de har studenttilbud nå før jul.

~~Kretsen vår?~~ – unnskyld, hadde noe i halsen. Kretsen vår:



Figur 3

Nome har fått superkrefter og kvantifisert seg. Klarer du å gjette hvilke deler av ham som er gjemt i bildet?

Script for plotting i Python

```python
import numpy as np
from scipy.integrate import odeint
import matplotlib.pyplot as plt


#start- og stoppverdier
t_start = 0
t_stopp = 40
t = np.arange(t_start, t_stopp+1, 1)


#verdier for motstand R og kondensator C:
R = 100*10**3
C = 100*10**(-6)

#funksjon som returnerer differensialligning
def diffligning(v, t):
    a = b = 1/(R*C)
    u = 9

    dvdt = -a*v + b*u
    return dvdt

#initialverdi
v_0 = 0

#finner funksjonen
v = odeint(diffligning, v_0, t)
#print(v)


#målte_verdier:

v_målt = [0, 3.2, 5.3, 6.9, 7.7, 8.3, 8.6, 8.8, 9.0]
t_målt = [0, 5, 10, 15, 20, 25, 30, 35, 40]


#plotting
plt.figure(figsize=(10, 10))
plt.plot(t, v, 'm', label='førstemann')
plt.plot(t_målt, v_målt, 'b', label='mølla')
plt.title("Graf")
plt.xlabel("t")
plt.ylabel("x(t)")
plt.ylim(0, max(v)+5)
plt.grid()
plt.legend()
plt.show()
```

Figur 4

Vi googlet hvordan man plotter differensiallingning i Python, og fikk veldig god hjelp av Hans-Petter Halvorsens blogg. (Halvorsen, u.d.).

Konklusjon:

Ting funker. Altså, de faktiske verdiene ligger tett opp mot den matematiske modelleringen basert på ideelle verdier. Det ser vi av grafene. Dette viser at å regne på ideelle kretser ikke er bortkastet tid – det er en god tilnærming for virkeligheten.

Figur 5

Takk for oss

Bibliografi

Halvorsen, H.-P. (u.d.). *halvorsen.blog.no*. Hentet fra Differential Equations in Python: https://www.halvorsen.blog/documents/programming/python/resources/powerpoints/Differential%20Equations%20in%20Python.pdf

PS: Er det sant at du sover i hengekøye? If True, print(svar = (input(«også om vinteren?»)))

PPS: vi avventer å lære oss LaTEX, men takk for tilbudet.