FYS1120 Elektromagnetisme

Labøving

Hall-effekt.

Hilde Solesvik Skeie

3. november 2014

Sammendrag

Sei noko her.

Innhold

Oppgåve 1

PRELAB-Oppgåve 1

Me vil lage eit program som tek imot datapunkter x og y og gjer oss eit plott over punkta mot eit polynom som interpolerer dei.

Eg har nytta Python kor metodane `polyval` og `polyfit` ligg i numpy-biblioteket. Me er då interesserte i eit program som les datapunkt frå ei fil og prøver å finne eit polynom som gjer oss ein god modell.

```
# coding: utf-8
from numpy import polyfit, polyval, zeros, array
from matplotlib.pylab import plot, show, title, xlabel, ylabel,\
        legend, hold, savefig

"""
Klasse som les punkter frå ei fil, interpolerer med eit polynom og
plottar resultatet mot einannan.
"""
class EvaluatePoints:

    """
    Konstruktør som tek imot ei fil med x- og y-verdiar.
    """
    def __init__(self, datapoints):
        self.datapoints = datapoints
        # Finn antal punkter i fila.
        with open(self.datapoints, 'r') as f:
            self.n = sum(1 for line in f)

    """
    Metode som les verdiane frå fila og lagrar dei i klassa.
    """
    def storeValues(self):
        self.points = zeros((2, self.n))
        with open(self.datapoints, 'r') as f:
            counter = 0
            for values in f:
                self.points[0, counter] = float(values.split()[0])
                self.points[1, counter] = float(values.split()[1])
                counter += 1

    """
    Metode som nyttar numpy-biblioteket til å interpolere punkta
```

```
med eit polynom.
"""
def interpolate(self, deg, evalPoint):
    # Lager eit polynom av grad 'deg' ut frå punkta.
    self.poly = polyfit(self.points[0, :], self.points[1, :], deg)
    self.evaluated = zeros(self.n)
    for i in range(self.n):
        self.evaluated[i] = polyval(self.poly, self.points[0, i])

"""
Metode som plottar datapunkta og polynomet mot einannan.
"""
def plotPoints(self, TITLE):
    plot(self.points[0, :], self.points[1, :], '.')
    hold('on')
    plot(self.points[0, :], self.evaluated, '-')
    hold('off')
    title(TITLE)
    xlabel('x [m]')
    ylabel('B [T]')
    legend(('datapunkter', 'polynomtilnærming',), loc=1)
    savefig('PRELAB3.png')
    show()
```

Oppgåve 2

PRELAB-Oppgåve 2.1

Me vil vise at straumtettheten j er identisk med magnetiseringa M.

Det enklaste er å sjå på einingane til j og M. Me får

$$j = \frac{dI}{dx} = \frac{I}{t} \sim \frac{\text{A}}{\text{m}},$$

kor straumen I er målt i Ampere og distansen dx er gjeve i meter. M er gjeve som magnetisk moment per volumeining. Då får me

$$M = \frac{\mu}{V} = \frac{IA}{V} = \frac{I\pi a^2}{\pi a^2 t} = \frac{I}{t} \sim \frac{\text{Am}^2}{\text{m}^3} = \frac{\text{A}}{m}.$$

I tillegg til å få dei same einingane veit me at me jobber med dei same "metrane".

PRELAB-Oppgåve 2.2

Vis at me ved å integrere opp overflatestraumen kan skrive magnetfeltet som

$$B_x(h) = \frac{\mu_0}{2} j \left[\frac{h+t}{\sqrt{(h+t)^2 + a^2}} - \frac{h}{\sqrt{h^2 + a^2}} \right].$$

Me set inn uttrykket for straumen I i uttrykket for $B_x(x)$.

$$j = \frac{dI}{dx} \qquad \Rightarrow \qquad I = j\ dx$$

Då får me

$$\begin{aligned}
B_x &= \frac{\mu_0}{2} I \frac{a^2}{(x^2 + a^2)^{3/2}} \\
&= \frac{\mu_0}{2} j \frac{a^2}{(x^2 + a^2)^{3/2}}\ dx \\
&= \frac{\mu_0}{2} j a^2 \int_h^{h+t} \frac{dx}{(x^2 + a^2)^{3/2}} \\
&= \frac{\mu_0}{2} j a^2 \left[\frac{x}{a^2 \sqrt{x^2 + a^2}} \right]_h^{h+t} \\
&= \frac{\mu_0}{2} j \left[\frac{h+t}{\sqrt{(h+t)^2 + a^2}} - \frac{h}{\sqrt{h^2 + a^2}} \right].
\end{aligned}$$

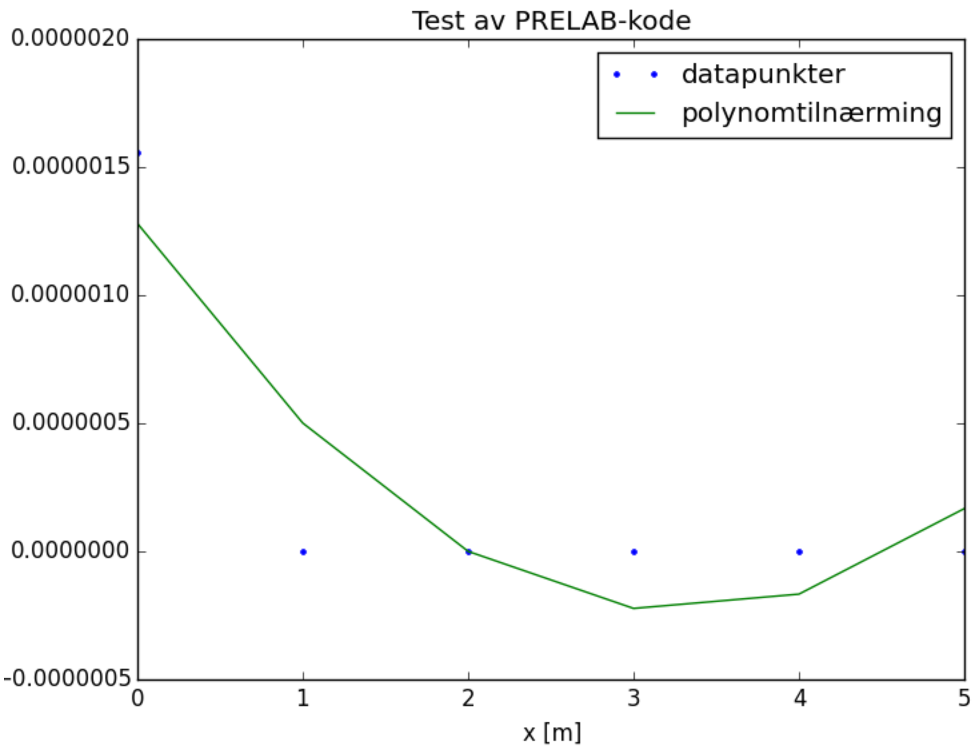
PRELAB-Oppgåve 2.3

I denne oppgåven vil me teste programmet frå **PRELAB-Oppgåve 1** med likninga frå **PRELAB-Oppgåve 2.2**. Me vil nytte 5-6 punkter for $t = 35$ mm og $a = 20$ mm.

Fyrst finner me vakuum permeabiliteten $\mu_0 = 4\pi \times 10^{-7}$ Vs/(Am) frå læreboka. For å finne j nyttar me resultatet frå **PRELAB-Oppgåve 2.1**.

$$j = M = \frac{IA}{V} = \frac{I\pi a^2}{\pi a^2 t} = \frac{I}{t}.$$

kor straumen I vert valgt til $I = 0.1$ A. Me får då resultatet



Figur 1: Her kan me sjå korleis polyfit prøver å tilpasse eit polynom til 6 punkter. Desverre vil det for så få punkter vere vanskeleg å gje eit nøyaktig resultat.

og utskrifta;

```
0   1.55867e-06
1   2.38584e-11
2   3.06056e-12
```

```
3   9.14742e-13
4   3.8759e-13
5   1.98966e-13
```

Programmet `PRELABOppgave3.py` er gjeve ved;

```
# coding: utf-8
from PRELABOppgave1 import EvaluatePoints
from numpy import pi, sqrt, zeros

my_0 = (4*pi)*10**(-7) # vakuum permeabiliteten.
a = 0.02 # radius i meter.
t = 0.035 # høgde i meter.
I = 0.1 # Ampere, denne må kunne endrast.
A = pi*a**2 # Areal over sida til magneten i kvadratmeter.
V = pi*a**2*t # Volumet til sylinderen i kubikkmeter.
j = I*A/float(V) # Straumtetthet i Ampere per meter.
Bx = lambda h: my_0/2.0*j*((h + t)/sqrt((h + t)**2 + a**2) - h/sqrt(h**2 + a**2))

h = zeros(6) # Evalueringspunkter.
evaluated = zeros(6) # Verdi av Bx i h.
for i in range(6):
    h[i] = i
    evaluated[i] = Bx(h[i])

with open('Oppgave3.txt', 'w') as f:
    for i in range(6):
        f.write('%g \t %g \n' % (h[i], evaluated[i]))

EP = EvaluatePoints('Oppgave3.txt')
EP.storeValues()
EP.interpolate(2, 1)
EP.plotPoints('Test av PRELAB-kode')
```

PRELAB-Oppgåve 2.4

Me er interesserte i å sjå på B-feltet i topp-flata av magneten og når $t \to \infty$. Då lurer me og på kva B-feltet midt i sylinderen vert for noko.

Me startar med å sette inn $h = 0$ for å finne B-feltet i topp-flata. Me får

$$\begin{aligned} B_x(0) &= \frac{\mu_0}{2} j \left[\frac{t}{\sqrt{t^2 + a^2}} - 0 \right] \\ &= \frac{\mu_0}{2} \frac{tj}{\sqrt{t^2 a^2}}. \end{aligned}$$

For $t \to \infty$ får me

$$\lim_{t\to\infty} B_x(h) = \lim_{t\to\infty} \frac{\mu_0}{2} j \underbrace{\left[\frac{t}{\sqrt{t^2}} - \frac{h}{\sqrt{h^2+a^2}}\right]}_{\approx 1}$$
$$= \frac{\mu_0}{2} j.$$

Midt i sylinderen har me at $h = -\frac{t}{2}$ det gjer oss

$$B_x(-\frac{t}{2}) = \frac{\mu_0}{2} j \left[\frac{t}{2\sqrt{(t/2)^2 + a^2}} + \frac{t}{2\sqrt{(-t/2)^2 + a^2}}\right]$$
$$= \frac{\mu_0}{2} \frac{t}{\sqrt{(t/2)^2 + a^2}}.$$