

Computational Essay

April 24, 2021

1 Introduksjon

Vi vil i dette prosjektet se nærmere på dispersjon fenomenet. Dispersjon kommer fra den latinske ordet «dispersio» som betyr fordeling eller oppdeling. Den elektromagnetiske bølgen, samt lys blir dispergert. Når lyset beveger seg gjennom et materialmediet, kan det dele seg i bølger med forskjellige frekvenser. Vi vil i dette prosjektet finne ut hvordan kan lys betraktes som en bølge, samtidig som vi skal se nærmere på dobbeltspalteeksperimentet utført av Thomas Young. Alle simuleringer skal vises med Python.

2 Bakgrunn

Isaac Newton er kjent for sin oppdagelse av at hvit lys består av et spekter av farger (Newtons fargespekter). I sin undersøkelse tok han utgangspunktet i brytningen av hvit lys når det passerte gjennom et prisme. Dispersjon av hvit lys skyldes variasjon i brytningsindeksen for prismet, dvs. at lysets hastighet i prismet avhenger av lysets bølgelengde. Hastigheten avtar med avtagende bølgelengde. Dette betyr at brytningsindeksen, som er omvendt proporsjonal med hastigheten, øker med avtagende bølgelengde. Med denne relasjon kan bl.a. plassering av farger i en regnbue forklares.

De fleste optiske fenomener i hverdagen kan forklares med en forenklet model, altså at lys er en samling av stråler som følger reglene for optikk. Men når dimensjonene til gjenstander og blenderåpninger nærmer seg bølgelengden til lys, kan ikke bølgekarakteren til lys ignoreres. Thomas Young viste at et interferensmønster produseres når lys fra to kilder overlapper hverandre. Dette demonstrerte tydelig at lys har bølgelignende egenskaper.

Et definierende kjennetegn ved bølger er superposisjon. Den beskriver oppførselen til overlappende bølger. Superposisjonsprinsippet sier at når to eller flere bølger overlapper i rommet, er den resulterende forstyrrelsen lik den algebraiske summen av individuelle forstyrrelser. Denne enkle underliggende atferden fører til en rekke effekter som kollektivt kalles interferensfenomener. Det er to grenser for interferenseffekter. Ved konstruktiv forstyrrelse faller to bølger sammen og bølgenes sies å være i fase med hverandre. Omvendt, i destruktiv forstyrrelse, faller toppen av en bølge sammen med dalen til en annen bølge, og de sies å være ute av fase.

3 Simuleringer

Vi tar utgangspunktet i bildet vist overfor. Vi ønsker å importere resultatene til Python. Vi starter med å importere alle nødvendige biblioteker for våre simuleringer:

```
[1]: # Importering av biblioteker
import numpy as np
import matplotlib.pyplot as plt
```

Vi definerer deretter plotens størrelse

```
[2]: #Definisjon av plotens størrelse
x_min = 0
x_maks = 10
y_min = -5
y_maks = 5
```

Vi definerer antall punkter vi ønsker å ha med i plotet

```
[3]: #Antall punkter
x_punkter = 200
y_punkter = 200
```

Vi lager en Array for videre simuleringer

```
[4]: #Lager en Array for plotting av resultater
x = np.linspace(x_min, x_maks, x_punkter)
y = np.linspace(y_min, y_maks, y_punkter)
```

Og ut fra disse lager vi en matrise

```
[5]: #Lager matrise
xv, yv = np.meshgrid(x, y)
```

Vi prøver nå å finne alle disse punktene. Vi omformer våre matriser

```
[6]: #Omforming
punkter= np.concatenate([xv.reshape(-1, 1), yv.reshape(-1, 1)], axis=-1)
```

Vi nå definerer “lys” kildene.

```
[7]: #Definering av kilder
kilde_1 = np.array([0, 0.5])
kilde_2 = np.array([0, -0.5])
```

Vi finner nå avstand

```
[8]: #Avstand
punkter_1 = punkter - kilde_1
```

```
punkter_2 = punkter - kilde_2
```

Vi definerer amplituden og bølgetallet. Foreløpig så har vi

```
[9]: #Amplitude  
a_1 = 4  
a_2 = 4  
  
#Bølgetall  
k = 20
```

Vi definerer nå bølgene. Vi bruker omregningen fra karteiske koordinater til poolarkoordinater, altså avstand $\sqrt{x^2 + y^2}$

```
[10]: #Bølge 1 og 2  
b_1 = a_1*(np.sin( k * (punkter_1[:, 0]**2 + punkter_1[:, 1]**2)**0.5))  
b_2 = a_2*(np.sin( k * (punkter_2[:, 0]**2 + punkter_2[:, 1]**2)**0.5))
```

Vi bruker superposisjonsprinsippet. For hver punktet vil amplituden bli lik

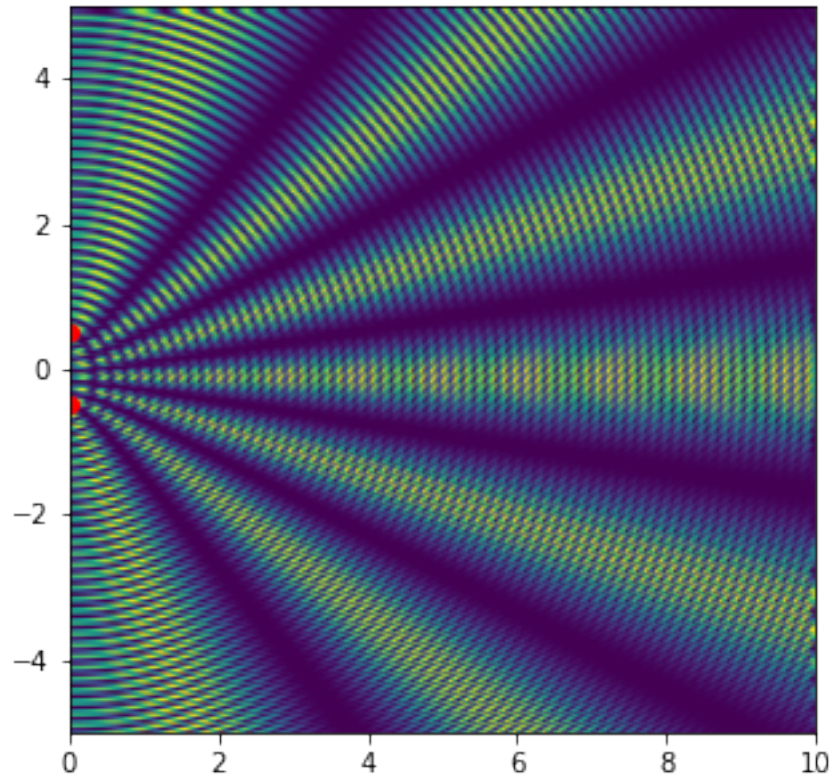
```
[11]: #Superposisjon  
A = (b_1 + b_2)
```

Intensitet til et punkt er direkte proporsjonal til kvadraten av amplituden A, altså

```
[12]: #Intensitet  
I = A**2
```

Vi ploter resultatene

```
[13]: plt.figure(figsize=(5, 5))  
plt.xlim(x_min, x_maks)  
plt.ylim(y_min, y_maks)  
plt.scatter(punkter[:, 0], punkter[:, 1], c = I)  
  
#Tegner opp spalter/lysemitter  
plt.scatter(*kilde_1, c='r')  
plt.scatter(*kilde_2, c='r')  
plt.show()
```



De fargende/lyse områder på plotter er når bølgene er i fase med hverandre, dvs. en konstruktiv forstyrrelse. Den mørke området på plotet er når bølgene er ute av fase, dvs. en destruktiv forstyrrelse. Vi lager nå animasjon av resultater. Vi velger variablene

```
[ ]: from IPython.display import display, clear_output

r = np.linspace(x_min, x_maks, int((x_maks-x_min)/0.5))
h = 0.1

fig, ax = plt.subplots(figsize=(5, 5))
plt.xlim(x_min, x_maks-1) #Vi velger x_maks for at animasjon skal se bedre ut
plt.ylim(y_min, y_maks)

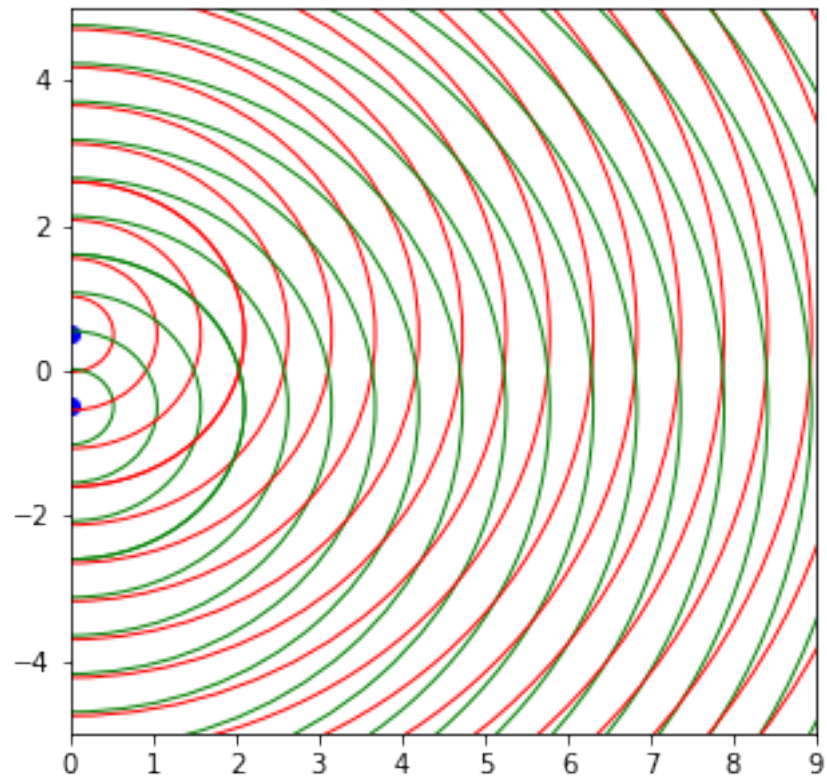
c_1 = []
c_2 = []

for i in r:
    sirkel_1 = plt.Circle(kilde_1, facecolor=(0.8, 0, 0, 0), edgecolor='r')
    c_1.append(ax.add_artist(sirkel_1))

    sirkel_2 = plt.Circle(kilde_2, facecolor=(0.8, 0, 0, 0), edgecolor='g')
    c_2.append(ax.add_artist(sirkel_2))
```

```
plt.scatter(*kilde_1, c='b')
plt.scatter(*kilde_2, c='b')

for j in range(10000):
    [c.set_radius(r[i]) for i, c in enumerate(c_1)]
    [c.set_radius(r[i]) for i, c in enumerate(c_2)]
    r = (r + h)%(x_maks-x_min)
    clear_output(wait=True)
    display(fig)
```



Her kan vi se den animerte versjonen av interferasjon av bølger.

4 Konklusjon

Vi kan se tydelig at lyset kan betraktes som en bølge. Young har vist med sin eksperimentet at interferasjon er en metode for å fremvise lys som en bølge. Våre simuleringer viser tydelig at hans eksperiment kan bli også bekreftet med datasimuleringer. Med dagens teori, vett vi at lyset er ikke en bølge, men kan på mange måter beholde seg som en bølge.

[]: