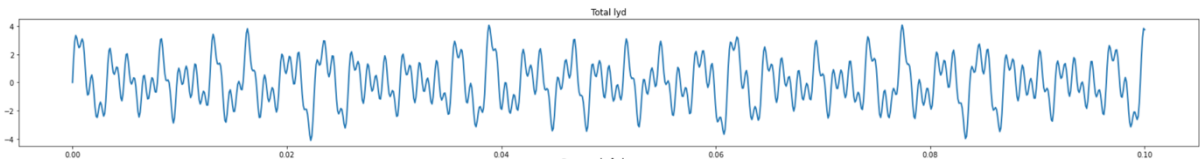


# Oblig

Dette prosjektet tar for seg følgende oppgave:  
 Du har sett Chemikalen, og tatt opp lyden fra et mellomspill. Du ønsker å finne ut hvilke toner som ble spilt på et tilfeldig slag.

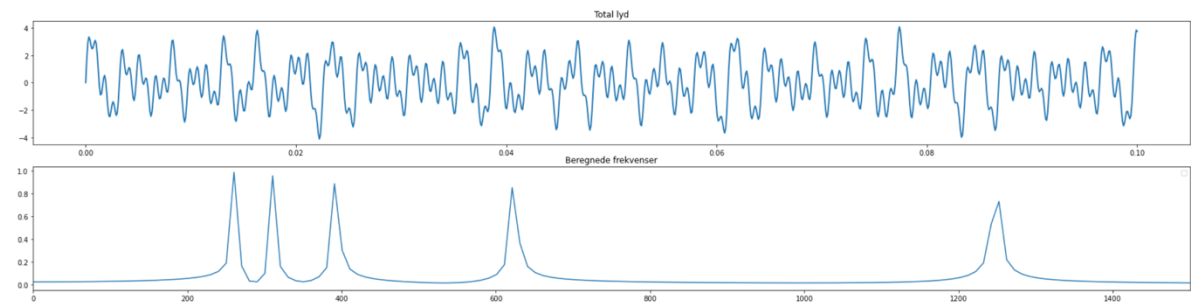
I denne oppgaven brukes ikke faktiske data hentet inn. Det har i stedet blitt laget data basert på partituret, som gjør at man kan se bort ifra bakgrunnsstøy fra f.eks. rekvisitter som flyttes rundt på scenen. Det antas også at instrumentene var stemt riktig, og at det ble spilt med god intonasjon. Mellomspillet som brukes i oppgaven ble spilt av ei tverrfløyte, en klarinett, to trompeter, og en trombone. Det sees bort ifra at messingen egentlig brukte cupmute.

For å løse denne oppgaven brukes fouriertransformasjon. Dataene som skulle analyseres ble fremstilt i et plott (Figur 1).



**Figur 1** Plott av den registrerte lyden. Tid er her målt i minutter, hvor det er omtrent ett slag per sekund, som er omtrent 0.02 minutter.

For å finne frekvensene som bidro til den totale mengden lyd, ble fouriertransformasjon brukt (Figur 2).



**Figur 2** Plott med total mengde lyd oppfattet, og hvilke frekvenser som i størst grad medvirket.

Det man kan se hos figuren er at de viktigste frekvensene ligger på rundt 250Hz, 300Hz, 400Hz, 650Hz og 1300Hz. Disse observerte verdiene passer godt med de teoretiske verdiene man bør kunne forvente (Tabell 1). Siden datasettet er laget basert på at det kun var disse tonene som ble oppfattet, hadde det vært rart om det var noen andre dominerende frekvenser.

**Tabell 1** Oversikt over tonene som ble spilt, og hvilken frekvens de i teorien har (MixButton, 2025).

Instrument	Tone	Frekvens (Hz)
Fløyte	Eb3	1244.51
Klarinett	Eb2	622.25
Trompet 1	G	392
Trompet 2	Eb	311.13
Trombone	C	261.63

Fouriertransformasjon har mange bruksområder, og det er mange alternative eller tilknyttede oppgaver som kunne blitt lagd om samme tema. Det kunne f.eks. ha vært å behandle cupmutene til messingen som lydfilter, for å finne ut hvilke frekvenser som stanses da. Man kunne også kuttet ut stemmen til et instrument man ikke liker ved å bruke fouriertransformasjon for å finne ut hvilken frekvens som skal fjernes, trekke fra frekvensen fra uttrykket for den fouriertransformerte, og så ta den inverse fouriertransformasjonen for å få et signal som kan gjøres om til lyd.

God påske!

## Kilder

MixButton. (27.01.2025). Music note to frequency chart [Tabell]. Hentet 30.03.2025 fra <https://mixbutton.com/music-tools/frequency-and-pitch/music-note-to-frequency-chart>

Sanderson, G. [3Blue1Brown]. (26.01.2018). *But what is the Fourier Transform? A visual introduction* [Video]. Youtube. <https://www.youtube.com/watch?v=spUNpyF58BY>

Fourier transform. (2025, 1. april). *Wikipedia*. Hentet 06.04.2025 fra [https://en.wikipedia.org/wiki/Fourier\\_transform](https://en.wikipedia.org/wiki/Fourier_transform)

## Vedlegg

### Vedlegg A

All kode

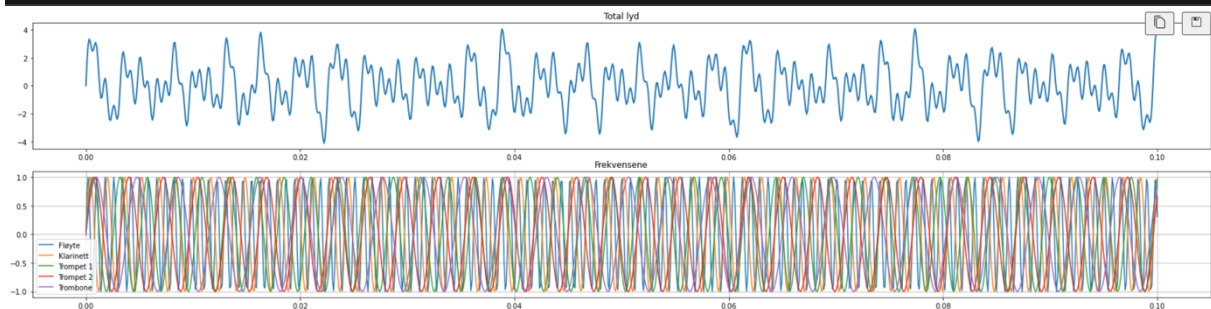
#### Data lages

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 # Definerer relevante funksjoner
5 def fløyte(t):
6     return np.sin(2*np.pi*t*1244.51)
7
8 def klarinett(t):
9     return np.sin(2*np.pi*t*622.25)
10
11 def trompet1(t):
12     return np.sin(2*np.pi*t*392)
13
14 def trompet2(t):
15     return np.sin(2*np.pi*t*311.13)
16
17 def trombone(t):
18     return np.sin(2*np.pi*t*261.63)
19
20 def total_lyd(t):
21     return fløyte(t) + klarinett(t) + trompet1(t) + trompet2(t) + trombone(t)
22
23 # Lager data
24 t1 = 0.0
25 T = 1/10000 # Sample spacing
26 N = 1000 # Sample points
27 t = np.linspace(t1, N*T, N)
28
29
```

```

28
29
30 # Plotting av alt dette
31 fig, ax = plt.subplots(2,1, figsize = (24,6))
32 fig.tight_layout()
33
34 ax[0].plot(t, total_lyd(t), linewidth=2.0, label = 'Total lyd')
35 ax[1].plot(t, fløyte(t), label = 'Fløyte')
36 ax[1].plot(t, klarinett(t), label = 'Klarinett')
37 ax[1].plot(t, trompet1(t), label = 'Trompet 1')
38 ax[1].plot(t, trompet2(t), label = 'Trompet 2')
39 ax[1].plot(t, trombone(t), label = 'Trombone')
40
41 ax[0].set_title('Total lyd')
42 ax[1].set_title('Frekvensene')
43
44 plt.legend()
45 plt.grid()
46 plt.show()
47
48

```



## Data analyseres

```

1  fouriertransformerte = np.fft.fft(total_lyd(t))
2  frekvenser = np.linspace(0.0, 1.0 / (2.0 * T), N // 2)
3
4  # Plotting av alt dette
5  fig, ax = plt.subplots(2,1, figsize = (24,6))
6  fig.tight_layout()
7
8  ax[0].plot(t, total_lyd(t), linewidth=2.0, label = 'Total lyd')
9  ax[1].plot(frekvenser, 2.0 / N * np.abs(fouriertransformerte[0:N // 2]))
10
11 ax[0].set_title('Total lyd')
12 ax[1].set_title('Beregnete frekvenser (Hz)')
13 plt.xlim(0,1500)
14 plt.show()

```

