

## Afleverin 2

Vi har givet et signal

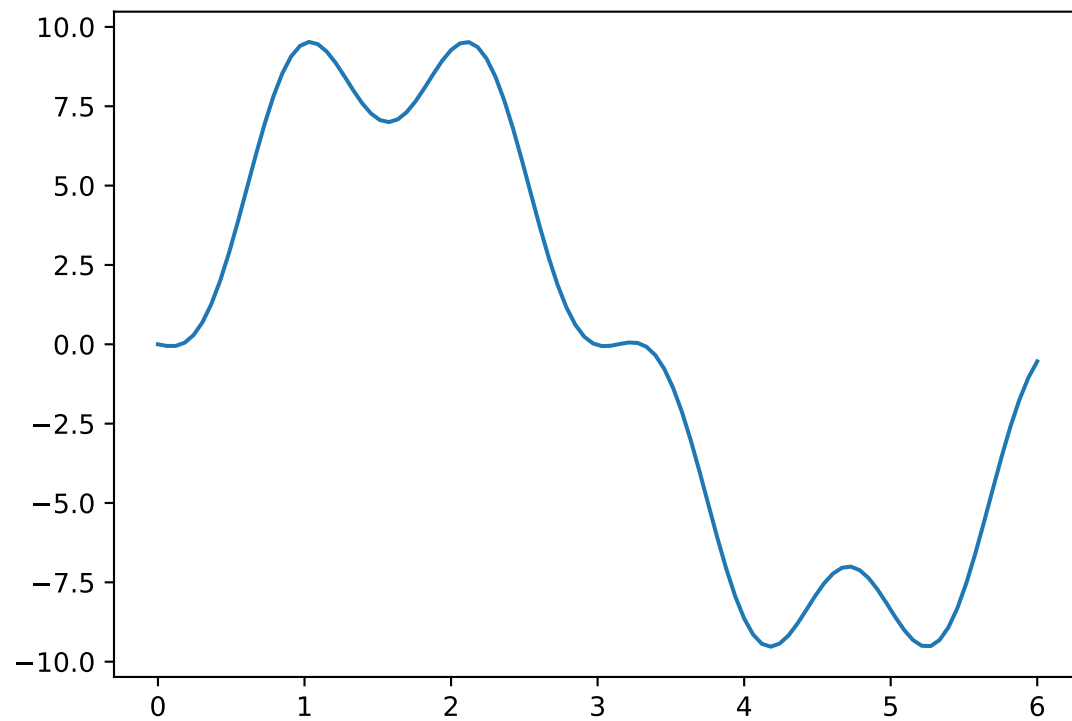
$$y(x) = 9\sin(x) - 2\sin(5x)$$

a) Lav en python plot af funktionen foroven med  $n = 100$  jævnt fordelt over intervallet  $[0, 6]$ .

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

Til at lave plottet af den ovenstående funktino benytter jeg mig af modulet `matplotlib`.

```
x = np.linspace(0, 6, 100)
y = 9 * np.sin(x) - 2 * np.sin(5*x)
fig, ax = plt.subplots()
ax.plot(x, y)
plt.show()
```



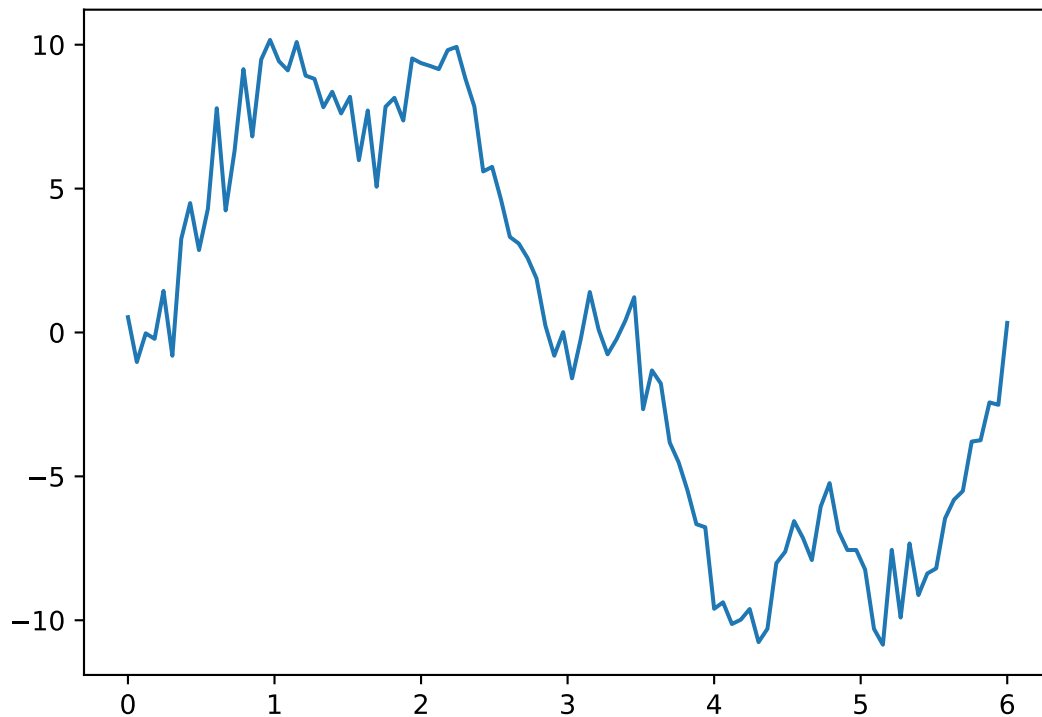
**b) Tilføj støj til funktionen og plot det.**

Her skal vi i y formelen tilføje følgende:

$$y_{støj} = y + rng.standardnormal(n)$$

```
n = 100
rng = np.random.default_rng()
støj = rng.standard_normal(n)
```

```
y_støj = y + støj
fig, ax = plt.subplots()
ax.plot(x, y_støj)
plt.show()
```



c)

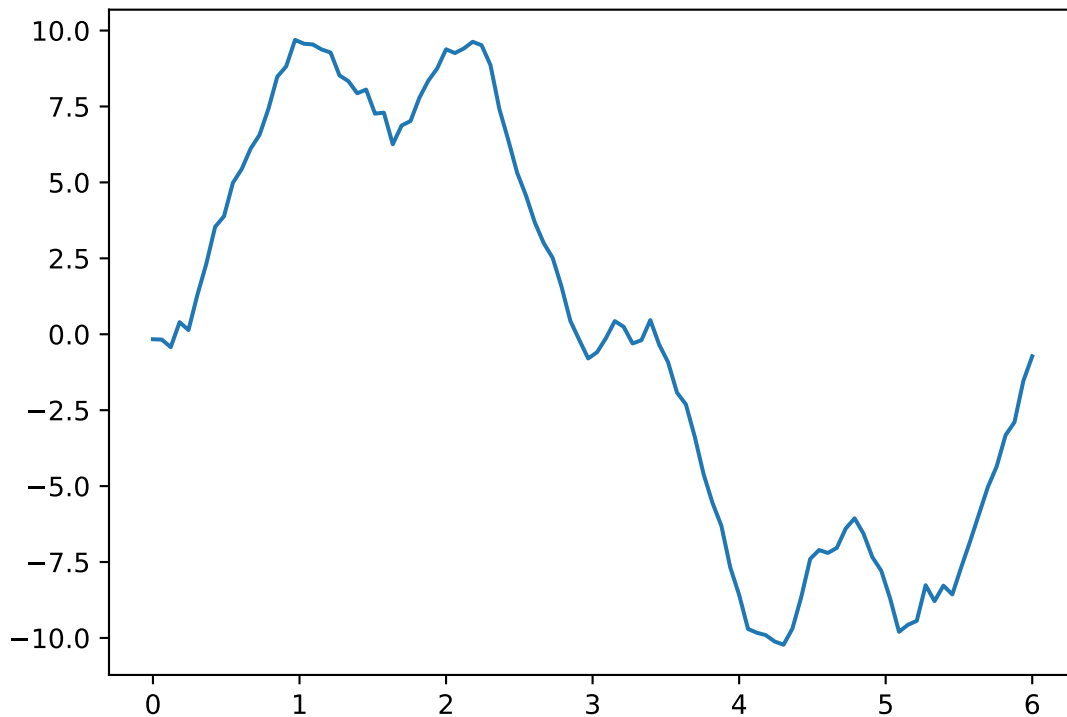
Givet en vektor  $v$  og et heltal offset vil funktionen `np.diag(v, offset)` danne en matrix med  $v$  langs en skrå linje hvor start punktet er forskudt fra det øverste venstre hjørne med offset. Brug `np.diag`, eventuelt kombineret med `np.ones()`, tre gange til at konstruere matrixen  $A$ , der har formen

```
for n in range(1,101):
    v = 1/3 * np.ones((1, n))
    A1 = np.diag(v[0], -1)
    A2 = np.diag(v[0], 0)
    A3 = np.diag(v[0], 1)
    A = A1[0:n, 0:n] + A2[0:n, 0:n] + A3[0:n, 0:n]
```

Foroven har jeg forsøgt a kontruer den ønskede matrice.

d) Plot  $A_{y\_støj}$ . Her skal vi se om den har en form der mindere mere om  $y$  end på  $y_{støj}$ .

```
A_y_støj = A @ y_støj
fig, ax = plt.subplots()
ax.plot(x, A_y_støj)
plt.show()
```



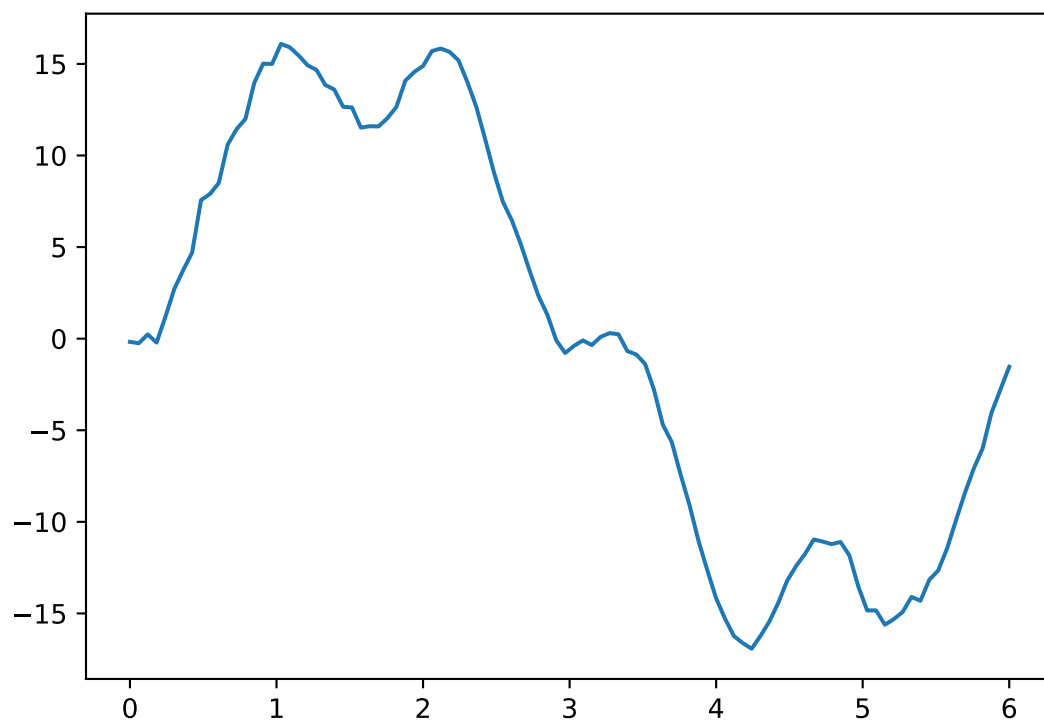
Der er noget som gør galt i min plot. Jeg tror det skyldes at jeg ikke får splitte min matrice A op i en enkelt array, så nu ser den som en nested funktion.

Dog med antagelse af jeg fik et plot, som minder mere om det oprindelige, så vil det ligne mere da vi inkludere flere multiplikation af nul, som gør at vi fjerne noget støj.

e) Ændre på vægtning i A og lave en matrix B, som er bedre end A.

```
for n in range(1,101):
    v = 1/3 * np.ones((1, n))
    B1 = np.diag(v[0], 1)
    B12 = np.diag(v[0], 2)
    B2 = np.diag(v[0], 0)
    B3 = np.diag(v[0], -1)
    B123 = np.diag(v[0], -2)
    B = B1[0:n, 0:n] + B12[0:n, 0:n] + B2[0:n, 0:n] + B3[0:n, 0:n] + B123[0:n, 0:n]

B_y_støj = B @ y_støj
fig, ax = plt.subplots()
ax.plot(x, B_y_støj)
plt.show()
```



I ovenstående B matrice har jeg tilføjet en vægtning på diagonalen. Ud fra ovenstående, så får jeg et pænerer resultatet end forrige opgave. Det giver fint mening da vi har en højere vægtning af støjen vi har tilføjet og vi får ikke et perfekt resultat, men noget som er bedre end før.