

Innleveringsoppgaver uke 43

Et enkelt, dobbelt plot

Velg to funksjoner og plot de innenfor de samme x-verdiene.

Lever fila som `f"uke43_{fornavn}_dobbelplot.py` og den resulterende grafen som `f"uke43_{fornavn}_dobbelplot.p`

Starten på utforsking av framoverdifferansen

Velg én av funksjonene fra forrige oppgave, denne skal du nå gjøre følgende med:

1. Finn funksjonens deriverte *analytisk*, og opprett denne som en egen funksjon
2. Lag et plot som sammenligner den analytiske løsningen med den numeriske tilnærmingen du finner ved å bruke *framoverdifferanse*-metoden fra sist uke.

Et eksempel på plotting

Om du har tre lister med verdier som skal plottes, en med x-verdier og to med y-verdier, kan du bruke en fremgangsmåte à la følgende for å få til dette.

Inntil videre kan du bruke den følgende fremgangsmåten ukritisk. Bruk hjelpesystemet aktivt om du ønsker å finne andre ting du kan endre med plottene dine, tilstrebe ikke å google problemer du eventuelt støter på.

```
import matplotlib.pyplot as plt

def f(x):
    return x ** 2 - 5

def g(x):
    return x ** 3 + 13

# Vi lager oss en liste [-10, -9, ..., 0, ... 9, 10]
x_verdier = list(range(-10, 11))

# Vi oppretter tomme lister som skal holde y-verdiene våre
y_verdier1 = []
y_verdier2 = []

# Vi legger funksjonsverdiene til y-verdilistene
for x in x_verdier:
    y_verdier1.append(f(x))
    y_verdier2.append(g(x))

# Vi oppretter et tomt plot
fig, ax = plt.subplots()
```

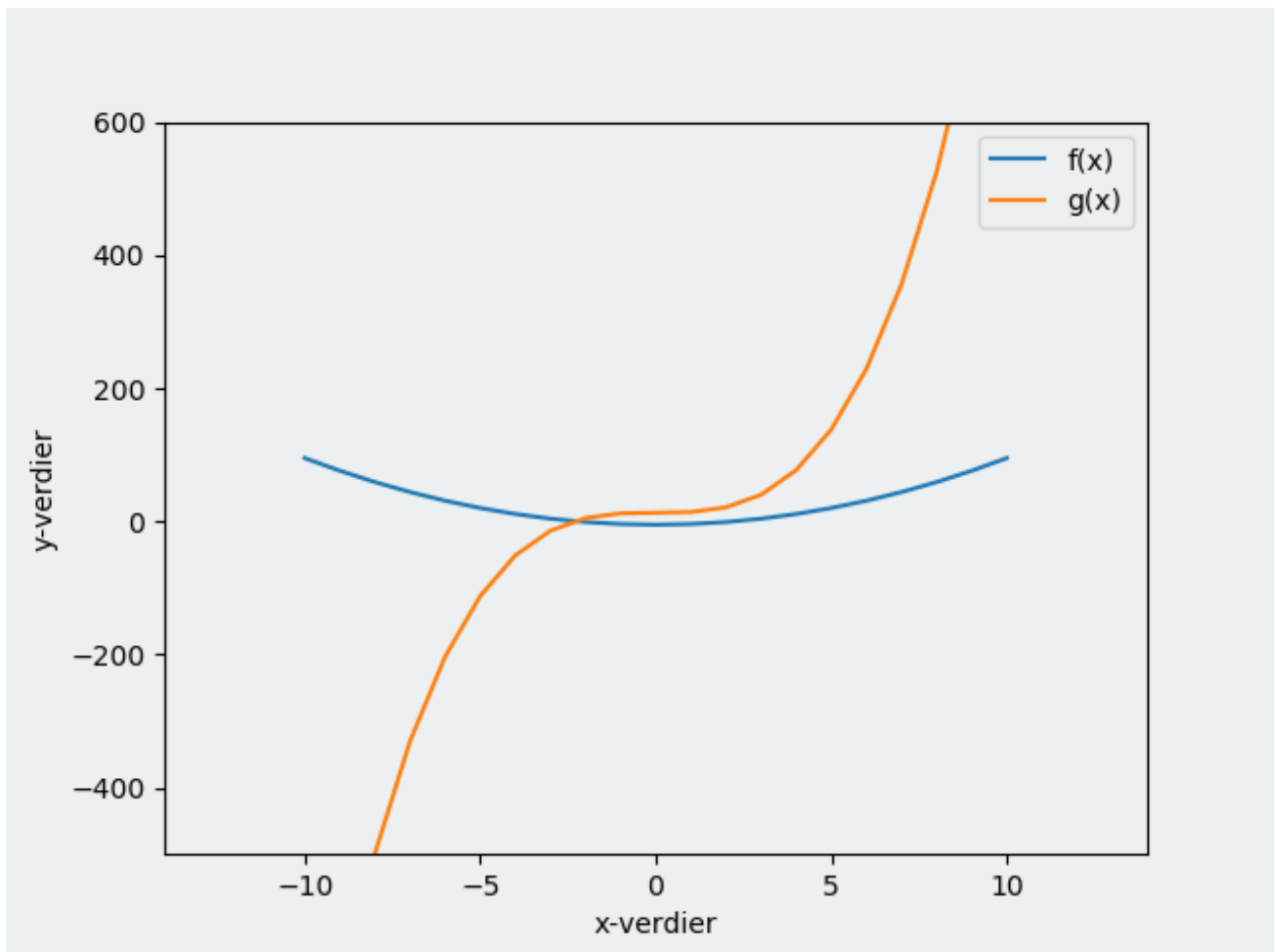
```
# Vi lager oss to linjer, en for hver liste av y-verdier. Legg merke til kommaet
linje1, = ax.plot(x_verdier, y_verdier1)
linje2, = ax.plot(x_verdier, y_verdier2)

ax.set_xlabel('x-verdier')
ax.set_ylabel('y-verdier')
ax.set_xlim(-14, 14)
ax.set_ylim(-500, 600)

# Vi setter navn på linjene våre
ax.legend([linje1, linje2], ['f(x)', 'g(x)'])

fig.savefig('eksempelplot.png')
fig.show()
```

Som produserer følgende plot:



Figur 1: Et enkelt, dobbelt plot