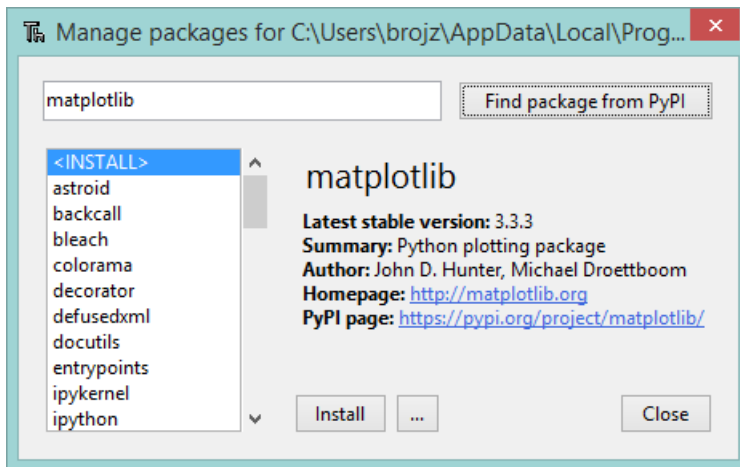


Opdrachten week 3 les 3 – Plotten van data

Voor de programmeertaal Python zijn vele uitbreidingsmodules beschikbaar. In de vorige lessen heb je al kennis gemaakt met de modules `math`, `cmath` en `statistics`. In deze les zul je de module `matplotlib.pyplot` leren gebruiken om gegevens in een nette grafiek te plotten.

De module `matplotlib.pyplot` is niet standaard beschikbaar in Thonny, maar moet nog geïnstalleerd worden.

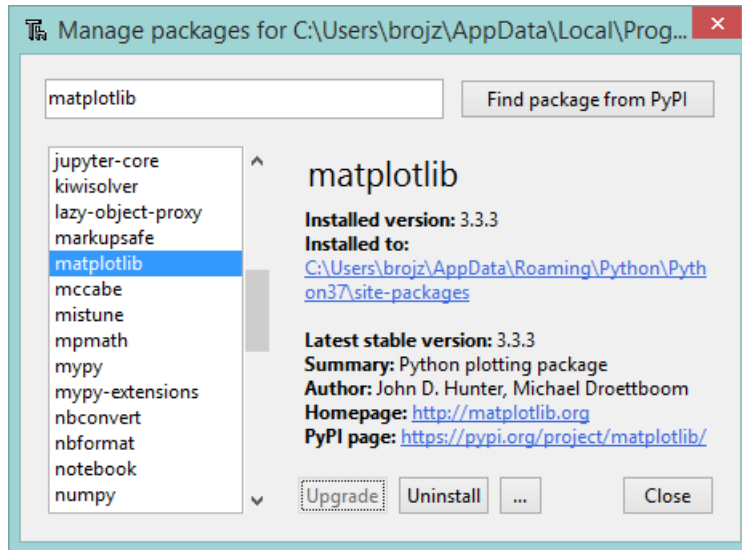
3.3.1 Start Thonny en kies de menuoptie `Tools >> Manage packages...`. Type de naam van de package¹ die je wilt installeren in: `matplotlib` en klik op de ‘Search’-knop. Als het goed is, verschijnt nu een ‘Install’-knop, zie [figuur 1](#).



Figuur 1: Het installeren van de package `matplotlib`.

Klik op de ‘Install’-knop. Thonny installeert ook alle packages waar `matplotlib` afhankelijk van is, zoals `numpy`. Als alles goed is gegaan, ziet het installatiewindow eruit zoals weergegeven in [figuur 2](#).

¹ Een package is een verzameling van een of meer modules. De package `matplotlib` bevat onder andere de module `pyplot`.



Figuur 2: Het installeren van de package matplotlib is gelukt.

Tot nu toe heb je alle benodigde informatie kunnen vinden in het boek². De module `matplotlib.pyplot` wordt echter niet besproken in het boek. Een eenvoudige handleiding voor beginners kun je online vinden: http://www.bogotobogo.com/python/files/python_matplotlib/PythonPlottingBeginnersGuide.pdf.

Lees nu de bovengenoemde handleiding tot en met paragraaf 3.2.5.

In [listing 1](#) is een programma gegeven dat twee perioden van een sinus plot. Op regel 1 wordt de module `numpy` geïmporteerd. De constructie `import numpy as np` zorgt ervoor dat je de module `numpy` kunt gebruiken met behulp van de naam `np`. Als je bijvoorbeeld de constante π wilt gebruiken, die in de module `numpy` gedefinieerd is als `pi`, dan kun je dat doen door de naam `np.pi` te gebruiken. Op regel 2 wordt de module `matplotlib.pyplot` geïmporteerd onder de naam `plt`.

Op regel 3 wordt een array³ genaamd `x` aangemaakt die bestaat uit 200 elementen. De array `x` wordt gevuld met getallen die op gelijke afstand van elkaar liggen beginnend bij 0 en

² Allen B. Downey. *Think Python: How to Think Like a Computer Scientist*. 2de ed. Green Tea Press, 2016. ISBN: 978-1-4919-3936-9. URL: <http://greenteapress.com/wp/think-python-2e/>.

³ Het datatype `array` is gedefinieerd in de module `numpy` en is vergelijkbaar met het in de vorige les behandelde datatype `list`. Het belangrijkste verschil is dat alle elementen van een array van hetzelfde type moeten zijn. Indien gewenst kun je meer informatie vinden over de module `numpy` en het datatype `array` op <https://docs.scipy.org/doc/numpy-dev/user/quickstart.html>.

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 x = np.linspace(0, 4 * np.pi, 200)
5 y = np.sin(x)
6 plt.plot(x, y)
7 plt.show()
```

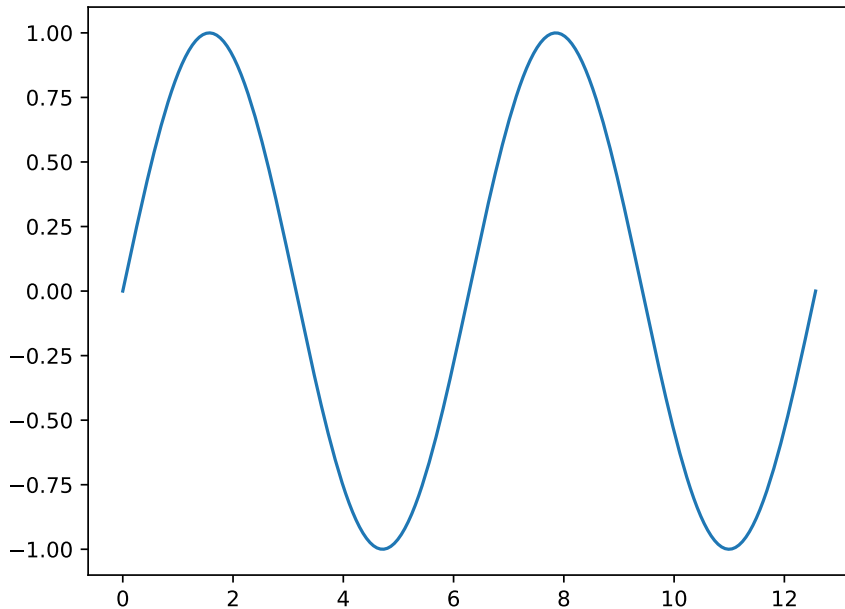
Listing 1: Python programma dat een sinus plot. Zie [plot_sin.py](#).

eindigend bij 4π . Dit zijn de x-waarden van de punten die later geplot zullen worden. Op regel 5 wordt een array genaamd y aangemaakt en gevuld met de sin van de waarden uit array x. De functie `np.sin` berekent de sinuswaarde van alle elementen uit de array x en slaat deze op in de array y. Array y bevat nu de y-waarden van de punten die later geplot zullen worden. In regel 6 worden alle punten (x, y) die opgeslagen zijn in de arrays x en y geplot in een grafiek. De punten worden met een lijn met elkaar verbonden. In regel 7 wordt deze grafiek op het scherm getoond.

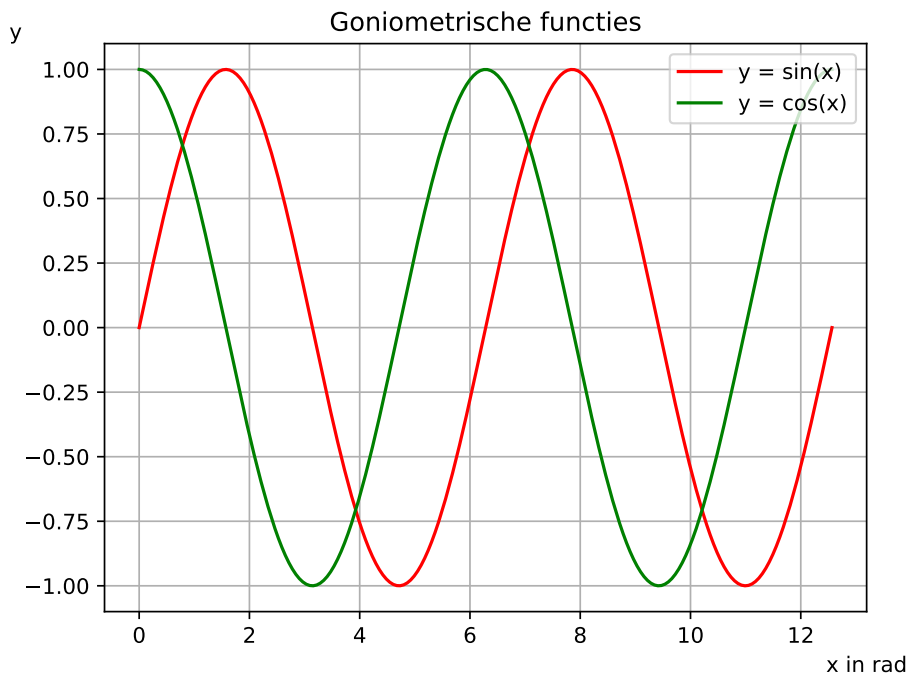
3.3.2 Open het programma [plot_sin.py](#) gegeven in [listing 1](#) in Thonny. Run het programma. Als het goed is, verschijnt de in [figuur 3](#) gegeven grafiek. Wat gebeurt er als je slechts 20 in plaats van 200 datapunten gebruikt om twee perioden van de sinus te plotten?

3.3.3 De grafiek die getoond wordt in [figuur 3](#) is erg kaal. Breid de code gegeven in [listing 1](#) uit zodat zowel een sinus als een cosinus wordt geprint. Zorg voor een titel boven de grafiek, een tekst bij de x- en y-as, een grid en een legenda. Probeer de uitvoer gegeven in [figuur 4](#) zo goed mogelijk te benaderen. Let daarbij op de kleur van de lijnen, de positie van de tekst bij de x- en y-as en de positie van de legenda. De benodigde informatie kun je gedeeltelijk vinden in de bovengenoemde handleiding. Let op: de manier waarop een legenda wordt aangemaakt in paragraaf 3.2.6 van de handleiding werkt niet bij de huidige versie van `matplotlib.pyplot`. De juiste manier om een legenda aan te maken en andere details kun je online vinden: <https://matplotlib.org/stable/api/index.html>.

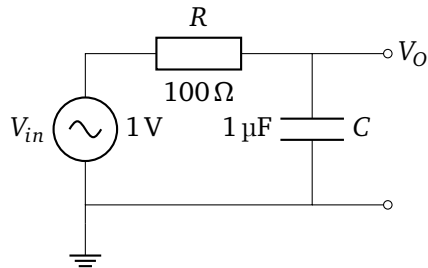
In het vorige kwartaal heb je bij lab 6 van de cursus ELE20 diverse overdrachtsfuncties van filters getekend. Je hebt onder andere metingen uitgevoerd aan het RC-filter gegeven in [schakeling 1](#).



Figuur 3: De uitvoer van het programma gegeven in [listing 1](#).



Figuur 4: De gewenste uitvoer bij [opdracht 3.3.3](#).

**Schakeling 1:** RC-filter

Stel dat je de waarden van V_O hebt gemeten die zijn gegeven in [tabel 1](#). Bij het practicum ELE20 moesten deze meetresultaten weergegeven worden in de grafiek die gegeven is in [figuur 5](#).

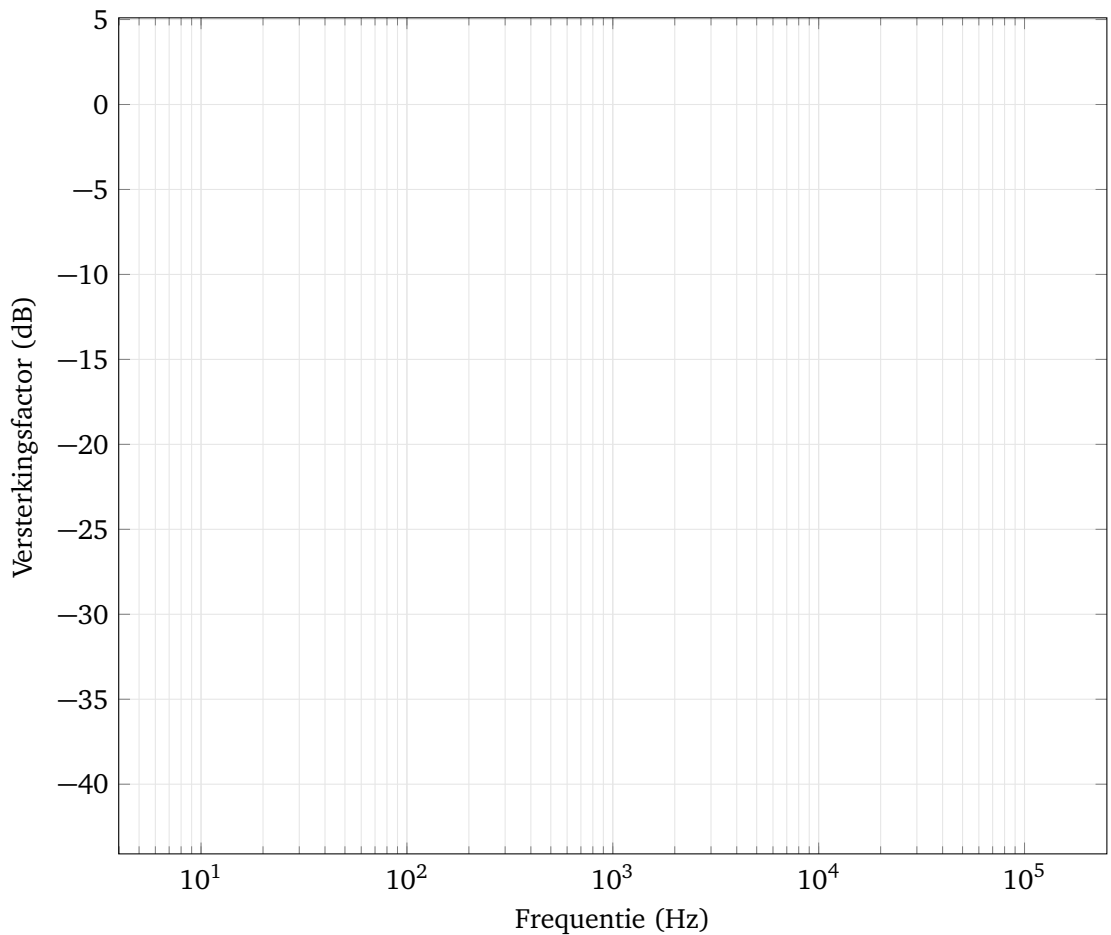
Tabel 1: Metingen aan een RC spanningsdeler

Frequentie	Spanning V_O
10 Hz	0,99 V
100 Hz	0,98 V
1 kHz	0,87 V
1,59 kHz	0,68 V
10 kHz	0,14 V
100 kHz	0,02 V

3.3.4 Schrijf een Python programma dat de meetresultaten gegeven in [tabel 1](#) tekent in een grafiek zoals gegeven in [figuur 5](#). Maak daarbij gebruik van [vergelijking \(1\)](#).

$$\text{Versterkingsfactor (dB)} = 20 \cdot {}^{10}\log \frac{V_O}{V_{in}} \quad (1)$$

Het gewenste resultaat is gegeven in [figuur 6](#).

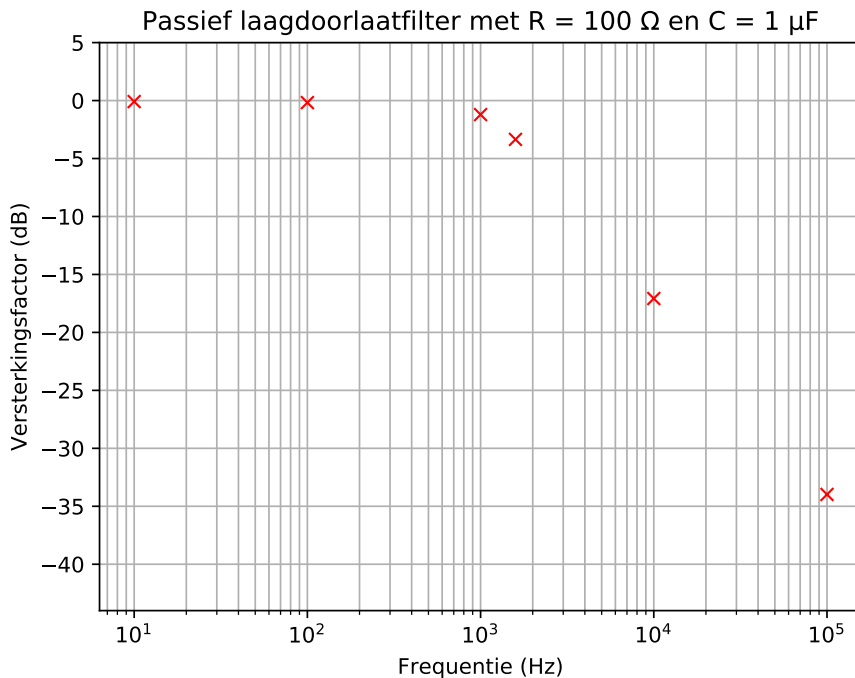


Figuur 5: Grafiek die ingevuld moest worden bij ELE20.

3.3.5 De versterkingsfactor van het filter gegeven in [schakeling 1](#) kan, bij een bepaalde frequentie f , ook worden berekend met behulp van [vergelijkingen \(1\)](#) tot en met [\(3\)](#).

$$\frac{V_O}{V_{in}} = |H(j\omega)| = \left| \frac{\frac{1}{j\omega C}}{R + \frac{1}{j\omega C}} \right| \quad (2)$$

$$\omega = 2\pi f \quad (3)$$



Figuur 6: De gewenste uitvoer bij opdracht 3.3.4.

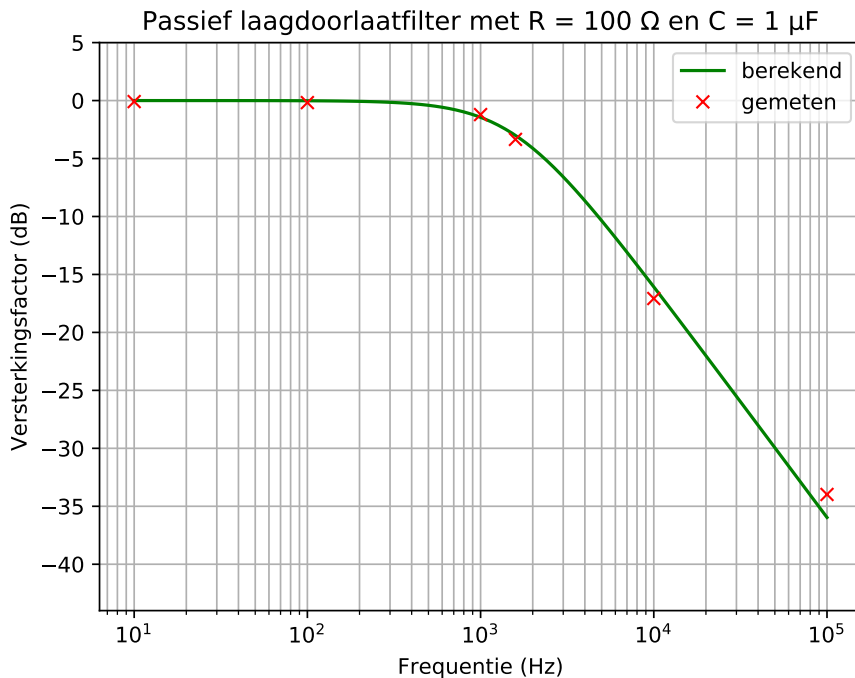
Breid het programma dat je bij opdracht 3.3.4 hebt geschreven uit, zodat ook de berekende waarde van de versterkingsfactor van het filter wordt weergegeven voor alle frequenties tussen 10 Hz en 100 kHz. Het gewenste resultaat is gegeven in figuur 7. Tip: om de x-waarden aan te maken die je kunt gebruiken om de y-waarden te berekenen kun je de functie `logspace`⁴ uit de module `numpy` gebruiken.

Een bodediagram⁵ bestaat uit twee boven elkaar geplaatste grafieken met een logaritmische frequentie-as: de versterkingsfactor in dB en de faseverschuiving in graden als functie van de frequentie. Een bodediagram kan onder andere gebruikt worden bij het ontwerpen en analyseren van filters. Als de overdrachtsfunctie $H(j\omega)$ van het filter bekend is, dan kan de versterkingsfactor in dB als functie van de frequentie berekend worden met vergelijkingen (4), (5) en (7). De faseverschuiving in graden kan berekend worden met vergelijkingen (6) en (7).

$$\text{Versterkingsfactor (dB)} = 20 \cdot \log |H(j\omega)| \quad (4)$$

⁴ Zie: <https://docs.scipy.org/doc/numpy-1.13.0/reference/generated/numpy.logspace.html>

⁵ Zie eventueel <https://nl.wikipedia.org/wiki/Bodediagram>.



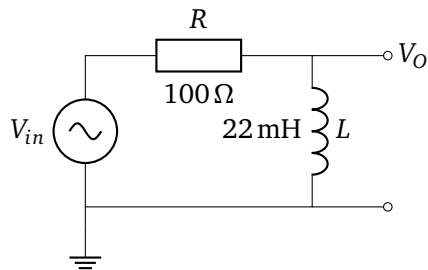
Figuur 7: De gewenste uitvoer bij [opdracht 3.3.5](#).

$$|H(j\omega)| = \sqrt{(\operatorname{Re}H(j\omega))^2 + (\operatorname{Im}H(j\omega))^2} \quad (5)$$

$$\text{faseverschuiving (graden)} = \arctan\left(\frac{\operatorname{Im}H(j\omega)}{\operatorname{Re}H(j\omega)}\right) \cdot \frac{180}{\pi} \quad (6)$$

$$\omega = 2\pi f \quad (7)$$

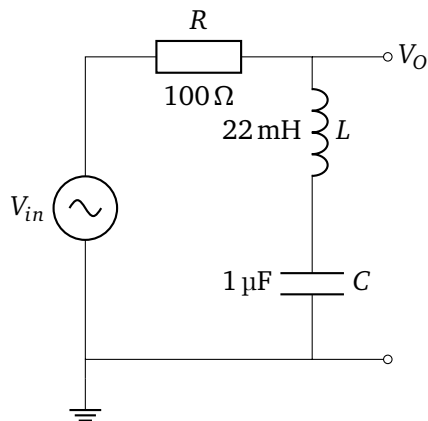
3.3.6 Schrijf een functie genaamd `plot_bodediagram` die een bodediagram plot van een overdrachtsfunctie $H(j\omega)$. Deze overdrachtsfunctie moet als argument aan de functie `plot_bodediagram` worden meegegeven. Als je het bodediagram wilt plotten van het filter dat gegeven is in [schakeling 2](#) waarvan de overdrachtsfunctie gegeven is in [vergelijking \(8\)](#), dan moet dat kunnen met de in [listing 2](#) gegeven code. De gewenste uitvoer is gegeven in [figuur 8](#). Als je het bodediagram wilt plotten van het filter dat gegeven is in [schakeling 3](#) waarvan de overdrachtsfunctie gegeven is in [vergelijking \(9\)](#), dan moet dat kunnen met de in [listing 3](#) gegeven code. De gewenste uitvoer is gegeven in [figuur 9](#).

**Schakeling 2:** RL filter

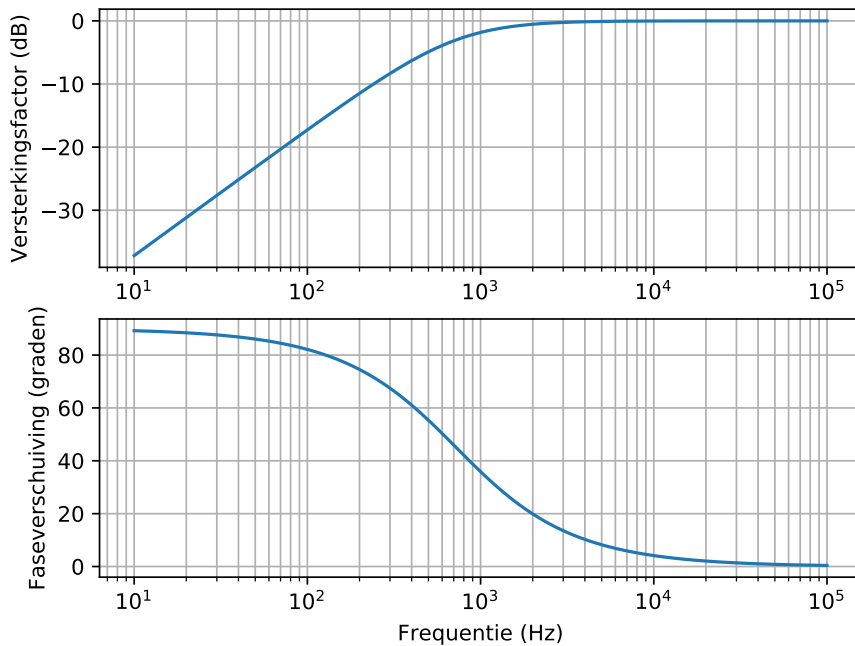
$$H(j\omega) = \frac{j\omega L}{R + j\omega L} \quad (8)$$

```
def H1(w):
    return 1j*w*22e-3 / (1e2 + 1j*w*22e-3)
plot_bodediagram(H1)
```

Listing 2: Python code om het bodediagram van het in [schakeling 2](#) gegeven filter te plotten. Zie [plot_bodediagram.py](#).

**Schakeling 3:** RLC-filter

Bodediagram



Figuur 8: De gewenste uitvoer van de code uit [listing 2](#).

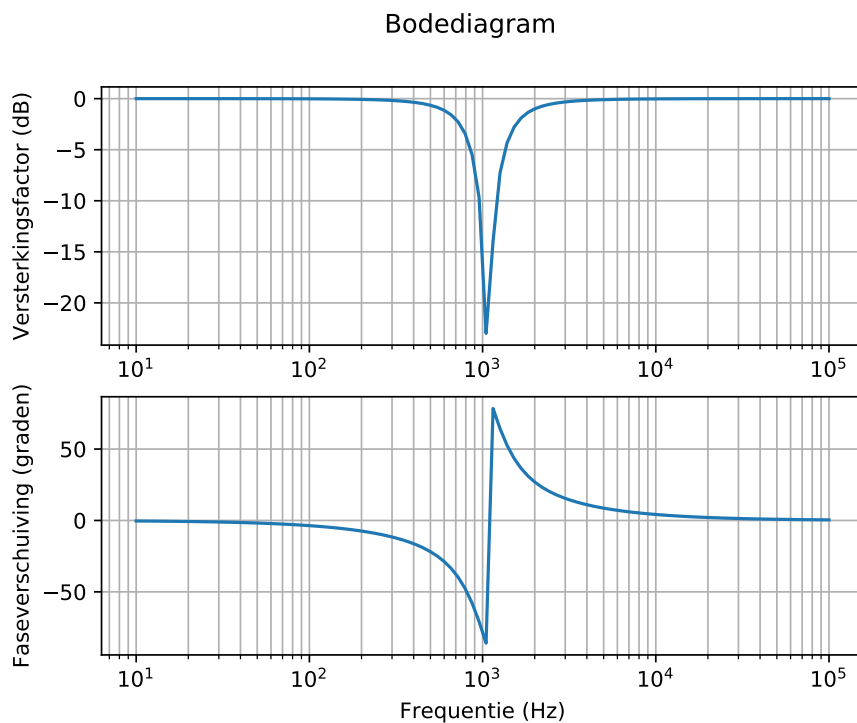
$$H(j\omega) = \frac{j\omega L + \frac{1}{j\omega C}}{R + j\omega L + \frac{1}{j\omega C}} \quad (9)$$

def H2(w) :

```

    return (1/(1j*w*1e-6) + 1j*w*22e-3) / (100 + 1/(1j*w*1e-6) + ↵
    ↵ 1j*w*22e-3)
plot_bodediagram(H2)
```

Listing 3: Python code om het bodediagram van het in [schakeling 3](#) gegeven filter te plotten. Zie [plot_bodediagram.py](#).



Figuur 9: De gewenste uitvoer van de code uit [listing 3](#).