

Opdrachten Sessie 14 (individueel)

Deze opdrachten worden individueel gemaakt en ingeleverd op Canvas. Bij het maken van de opdrachten is het toegestaan de documentatie behorende bij het vak en eerder gemaakte opdrachten te gebruiken. **Het is NIET toegestaan internetbronnen te raadplegen.** Elke opdracht levert één .py file op die ingeleverd wordt in Canvas. Er zijn in totaal 7 opdrachten. Het totaal aantal te behalen punten is 100. Je moet minimaal 60 punten scoren voor een VOLDAAAN.

1 (5p) lever in als: opdracht1.py

Definieer een 2D numpy array \mathbf{z} met 2 kolommen:

- de eerste kolom bestaat uit getallen in het interval $[0, 10]$ met stappen van 0.5.
- de tweede kolom bestaat uit de waarden van de eerste kolom afgerond op gehele getallen.

2 (10p) lever in als: opdracht2.py

Gegeven is de volgende list:

```
s = [1, 3, 5, 6, -1, 3, -5, 6, 7]
```

Schrijf een script die:

- a) door middel van *slicing* van \mathbf{s} de list \mathbf{t} definieert met als waarde:

```
[1, 6, -5]
```

- b) door middel van *list comprehensions* uit list \mathbf{s} een list \mathbf{u} definieert met alle elementen van \mathbf{s} die kleiner dan 0 zijn:

```
[-1, -5]
```

3 (10p) lever in als: opdracht3.py

The logistic map wordt gegeven door de volgende relatie:

$$\begin{aligned} x_n &= 0.5 && \text{if } n = 0 \\ x_{n+1} &= rx_n(1 - x_n) && \text{if } n > 0 \end{aligned}$$

waarbij gekozen is voor een start waarde van $x_0 = 0.5$.

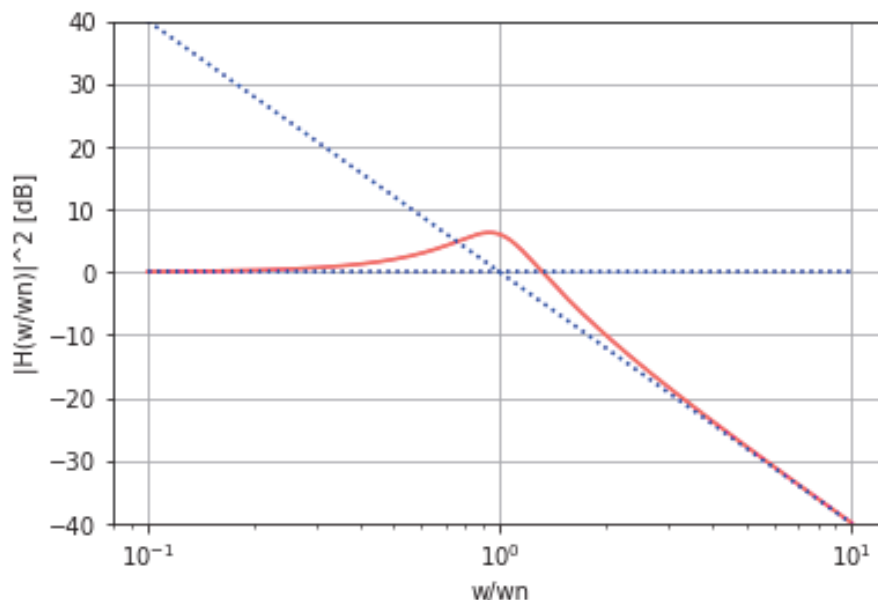
- a) Schrijf een functie `logistic_map(r, n)` die x_n berekent voor n iteraties.
- b) Voeg code toe die x_n voor $r = 3.4$ en $n = 0, 1, 2, 3, \dots, 9, 10$ bepaalt en de resultaten opslaat in een list `xn`

4 (15p) lever in als: opdracht4.py

Gegeven is onderstaande figuur van de amplitude van een overdrachtsfunctie:

$$|H(\omega)| = \frac{k}{\sqrt{(1 - (\omega/\omega_n)^2)^2 + 4\zeta^2(\omega/\omega_n)^2}}$$

, met $k = 1$, $\omega_n = 1$ rad/s, en $\zeta = 0.25$



Schrijf een script die deze figuur *exact* reproduceert. Let op: op de y -as staat $|H|^2$ in decibel (dus $20 \log_{10}(H)$)

5 (20p) lever in als: opdracht5.py

Schrijf een functie `smallest(z, n)` die gegeven een list z en positieve integer n de n kleinste *unieke* waarden van de list bepaalt en een list returned met die waarden. Indien er geen n unieke waarden te vinden zijn bestaat de gereturnde list uit het maximaal aantal te bepalen unieke waarden. Dus bijvoorbeeld:

IPython Console

```
In [1]: z = [4, 2, 9, 678, 45, -45, -23, 34, 34, 0, -1, -1]
In [2]: smallest(z, 4)
Out [2]: [-45, -23, -1, 0]
```

In dit voorbeeld is duidelijk dat het om unieke waarden gaat: de 4 kleinste waarden zijn dus niet `[-45, -23, -1, -1]`.

6 (20p) lever in als: opdracht6.py

Gebruik voor deze opdracht de file `sample.csv` en `blanco.csv` (zie canvas). Beide files bestaan uit één kolom met getallen. De eerste file bestaat uit gemeten lichtintensiteiten en de tweede file bestaat uit bijbehorende blanco meting. Om de metingen verder te verwerken is het nodig om hieruit de absorbance te bepalen volgens:

$$\text{absorbance} = -\log_{10}\left(\frac{\text{sample}}{\text{blanco}}\right)$$

Schrijf een script die de data van de twee files leest en een nieuwe file maakt `absorbance.csv` met één kolom met daarin de waarden van de absorbance volgens bovenstaande uitdrukking.

7 (20p) lever in als: opdracht7.py

Gegeven is een numpy array `w` met unieke integers én een integer `product`. De taak is om twee getallen te vinden in de array zodat het product van die twee getallen gelijk is aan de waarde van `product`. Bijvoorbeeld als `w = [1, 2, 3, 4, 5]` en `product = 12` dan is het antwoord het getallenpaar 3 en 4 omdat $3 \times 4 = 12$. Indien er meerdere getalparen mogelijk zijn dan is het antwoord dat getalpaar dat de laagste som van de indices in de array heeft. Dus bijvoorbeeld als `w = [1, 2, 3, 4, 5, 6]` en `product = 12` dan zijn er twee getalparen: 3 en 4 én 2 en 6. De som van de indices zijn bij het eerste paar gelijk aan $2+3 = 5$ en bij het tweede paar $1+5 = 6$. In dit geval is het antwoord dus getalpaar 3 en 4.

Schrijf een functie `two_product_problem(w, product)` die bovenstaande probleem oplost en het antwoord (getalpaar) returned als een tuple van de twee getallen. Indien geen getalpaar bestaat die voldoet, dan returned de functie de boolean `False`