Opgave 0: Basics

1. Werk je op je eigen apparaat? Zorg dat `Anaconda` geïnstalleerd is. Download van: `https://www.anaconda.com/`.

   Op de labcomputers is ook een versie van `Anaconda` geïnstalleerd.

2. Download vervolgens het bestand `*****.zip` van Brightspace en pak de `.zip` uit op een handige locatie.

3. Open `Spyder` en open het bestand `convert_slice_fit.py`

   Spyder is de standaard `IDE` van `Anaconda`

Opgave 1a: Een bestand inladen

Het bestand `convert_slice_fit.py` bevat code om een een spanningsverloop om te zetten in een weerstandsverloop, om dat vervolgens om te zetten in de transducer-grootheid: luchtvochtigheid[1]. Vervolgens wordt er een fit van *drie parameters* uitgevoerd. Dit alles ga je nodig hebben bij de practica van EN1.

1. Voer het bestand uit in `Spyder`. Er wordt nu een fit gedaan van een bestand genaamd `gaussian_data.csv` in de map '`data`'. Je krijgt de gevonden fitparameters terug, de fouten daarop, en een figuur waarin de gefitte data te zien is. Is deze fit goed?

   Krijg je een foutmelding? Vraag een studentassistent.

2. Bekijk de 'legenda' (grote comment) bovenaan het bestand en kijk eens door de code heen. De 'legenda' legt in grote lijnen uit wat de code doet en waar je die kunt aanpassen. Merk op dat we *niet* vragen om nieuwe code te schrijven!

3. Pas de code aan zodat nu het bestand `exponential_data.csv` in de map '`data`' wordt geladen. De plek waarop dat kan is aangegeven in de code. Voer de code uit en bekijk het resultaat. Is deze fit goed? Bedenk wat er verbeterd kan worden.

   De regels die met een `#` beginnen zijn 'comments' en worden niet uitgevoerd. Hetzelfde voor tekst tussen drie aanhalingstekens: `'''...'''`.

[1] De formule voor omzetting naar luchtvochtigheid en bijbehorende fout ga je in een latere opgave kloppend maken. Neem voor nu aan dat de omzetting juist verloopt

Opgave 1b: Een fitfunctie en `p0` kiezen

De code voert (zoals gedownload) een fit met de functie `gaussian_model(x, a, b, c)` uit. Maar in dit geval is dat niet het beste model.

1. Pas de code aan zodat de code een exponentiele functie fit. Deze is voorgeschreven en heet '`exponential_model(x, a, b, c)`'. De plek om dit aan te passen is aangegeven in de code. Voer de aangepaste code uit.

Het fitprogramma werkt door verschillende waardes voor de parameters `a,b,c` uit te proberen en te kijken welke de beste fit geven. Voor de opgaven hierboven hadden we de juiste beginwaardes ingesteld zodat de fits goed gingen. Als je de verkeerde beginwaardes meegeeft, dan kan `scipy.optimize` geen goede fit vinden.

2. Pas de code aan zodat het bestand `sine_data.csv` wordt gefit. Pas de fitfunctie (model) aan naar `sine_model`. Laat de beginwaardes voor de fitparameters hetzelfde. Is de fit goed?

De functie `sine_model` is van de vorm:

$$a \cdot \sin(b \cdot (x - c)) \tag{1}$$

3. Verbeter de fit door realistische fit-beginwaarden voor a, b, c te kiezen. Gebruik de figuur die in `Spyder` te zien is. Wat zijn de periode ($= 2\pi/b$), amplitude (a) en verschuiving (c) van de sinusoide? [2]

   Sinusvormige signalen zijn notoir moeilijk te fitten. Daarom wordt bij periodieke signalen vaak naar toppen gezocht, of een `FFT` gebruikt. Meer daarover in het tweede jaar bij het vak PE1.

Opgave 1c: van weerstand naar grootheid

Bij veel transducers schaalt de te bepalen grootheid bij lange na niet lineair met de weerstand. Dan is het nodig om de gemeten data (vaak spanning) om te zetten in weerstand, om die weerstand vervolgens weer om te zetten in de grootheid van de sensor. Databestand `exponential_data_unconverted.csv` bevat een exponentieel verval in de grootheid luchtvochtigheid, maar de berekende weerstand is nog niet juist omgerekend naar luchtvochtigheid. Bij opgave 2 oefen je met het opstellen van een ijkfunctie. Gegeven is de ijkfunctie van de luchtvochtigheid $H(R(t))$:

$$H(R(t)) = e^{0.5 \cdot R(t)} \tag{2}$$

1. Laad het bestand `exponential_data_unconverted.csv`, zet de data via de ijkformule om in luchtvochtigheid over tijd en voer een exponentiele fit

[2] In de code is arbitrair een 1:1 verband tussen de spanning en de weerstand gebruikt om dit makkelijker te maken. De foutberekening is ook arbitrair en

uit. Pas de initiele gokparameters aan, en schat zelf een waarde voor a op basis van de grafiek. Gebruik voor de exponenent de numpy-functie `np.exp(...)` (`bekijk eventueel de documentatie`). Deze omzetting correnspondeert met stap (2) in de illustratie hieronder:

$$V(t) \xrightarrow{1.} R(t) \xrightarrow{2.} H(t)$$

2. Schaal de assen logaritmisch door de optie `plt.semilogy()` aan te zetten. Dit kan je doen door de hashtag (`#`) weg te halen. Is het verband exponentieel? Waar kan je dit aan zien? Zie ook in de handleiding EN de sectie 'Niet-lineair Plotten'.

Opgave 1d: data *slicen*

Het kan soms voorkomen dat niet alle data van toepassing is voor je analyse, bijvoorbeeld wanneer een meetapparaat al aan staat voordat je opstelling werkt. In dat geval kan je een gedeelte wegsnijden 'slicen'. Doe dat nooit zonder goede reden!

1. Open in `Spyder` nu ook het python-bestand `quick_dataplot.py` en pas de code aan zodat `noise_data.csv` wordt geplot. Het bestand `noise_data.csv` bevat een stuk ruis en een stuk data. Zoek uit bij welk datapunt de data begint. Welke vorm heeft de data? Hoe zal de data eruit zien als deze is omgerekend in luchtvochtigheid?

2. Ga in `Spyder` terug naar het originele bestand `convert_slice_fit.py` en pas de slice aan zodat alleen de data wordt meegenomen. In het bestand staat aangegeven waar dat kan.

   Gebruik dezelfde formule voor het omzetten van weerstand naar luchtvochtigheid als eerst.

Opgave 2: Standaardafwijking

Soms is het handig om de meetfout te bepalen door te kijken naar de ruis op een nulmeting. De standaardafwijking, die je eerder met de hand berekende, kan ook berekend worden door `Python`. Neem aan dat bij een constant vooronderstelde luchtvochtigheid een nulmeting is gedaan.

1. Open het bestand `standard_deviation.py` in spyder. Pas de code aan zodat Het bestand `humidity_noise.csv` wordt geladen. Wat is het gemmiddelde van de data en wat is de standaardafwijking?

2. Zet het berekende gemiddelde en de berekende standaardafwijking om in de grootheid H met de eerder gegeven formule voor H. Gebruik voor de

fout de formule uit de handleiding EN: (Sectie Meetfouten doorrekenen in N Variabelen)

$$\sigma_p = \sqrt{\left(\frac{\partial p}{\partial x_1}\sigma_1\right)^2 + \left(\frac{\partial p}{\partial x_2}\sigma_2\right)^2 + \ldots + \left(\frac{\partial p}{\partial x_N}\sigma_N\right)^2} \tag{3}$$

Ga ervan uit dat de geschreven fout in R klopt. Bereken dus: $\sigma_R \to \sigma_H$

3. Ga in `Spyder` terug naar het originele bestand `convert_slice_fit.py`. Bouw de formule voor de fout in luchtvochtigheid in die je hierboven berekend hebt in en test de code

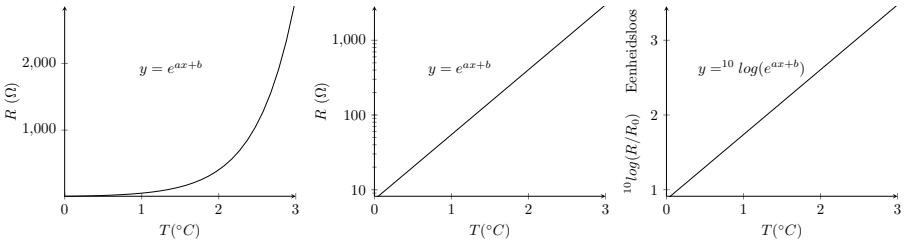
4. De fout in de spanning was ingesteld op: `voltage_error = 0.02 * voltage`. Pas deze aan zodat je de standaardafwijking die je in (1) hebt gevonden.

   Truc: verander de code naar `0.00 * voltage + (...)` Zo behoudt je data de juiste vorm.

5. Zet de comment bij het `plt.savefig` commando uit en sla de figuur op als pdf, met een naam naar keuze.

Opgave 3: Van ijkgrafiek naar formule

IJkgrafieken van transducers worden vaak gegeven in (dubbel)logaritmische grafieken. In deze opgave kan je alvast oefenen wat je later nodig hebt voor het omzetten van een ijkgrafiek. Voor de twee rechter grafieken is een ^{10}log gebruikt.



Figuur 1: De functie $R(T) = e^{aT+b}$ geplot met lineaire y-as (L) en logaritmisch *geschaalde* y-as (M). In de rechtergrafiek (R) is de functie $R(T)$ eenheidsloos gemaakt en is het 10-logaritme genomen

Merk het verschil op tussen de twee rechter grafieken: de ene grafiek heeft logaritmisch geschaalde *assen*, de ander is eenheidsloos gemaakt door te delen door R_0 (bijvoorbeeld $R_0 = 1\Omega$) en vervolgens het 10-logaritme te nemen.

1. Stel een lijn op van de vorm $y = mx + c$ door de meest rechter figuur. Gebruik hiervoor twee punten. Deze lijn geeft nu het verband tussen de y-as: $\left(^{10}\log[\ R(T)/R_0\ ]\right)$, en de x-as: T

Het getal c is eenheidsloos en m heeft eenheden $1/(^\circ C)$

2. Schrijf de gevonden relatie om zodat je een relatie $R(T)$ krijgt

3. Stel uit de middelste figuur een functie van de vorm $y = e^{\tilde{m}x+\tilde{c}}$ zodat je een relatie $R(T)$ krijgt. Vergelijk met het resultaat hierboven.

4. Leg uit waarom voor een verband van de vorm $Q = e^{\lambda x+\mu}$ op lineare en log-geschaalde assen geldt dat:

$$\lambda = \frac{\ln(y_2/y_0) - ln(y_1/y_0)}{x_2 - x_1} \equiv \frac{\Delta ln(y/y_0)}{\Delta x} \tag{4}$$

   Ter inspiratie kan je de 'Cursus Experimentele Natuurkunde', sectie 'Niet-lineair Plotten' gebruiken.

Opgave 4: extra's (lastiger)

1. Bekijk het bestand `timestamp_generator.py`. Zoek de documentatie op van `np.savetxt`. Schrijf code die de data in de gewenste grootheid opslaat met een timestamp. Doe hetzelfde met een plot

2. Verklaar waarom het fitten van sinusoiden zo moeilijk is.

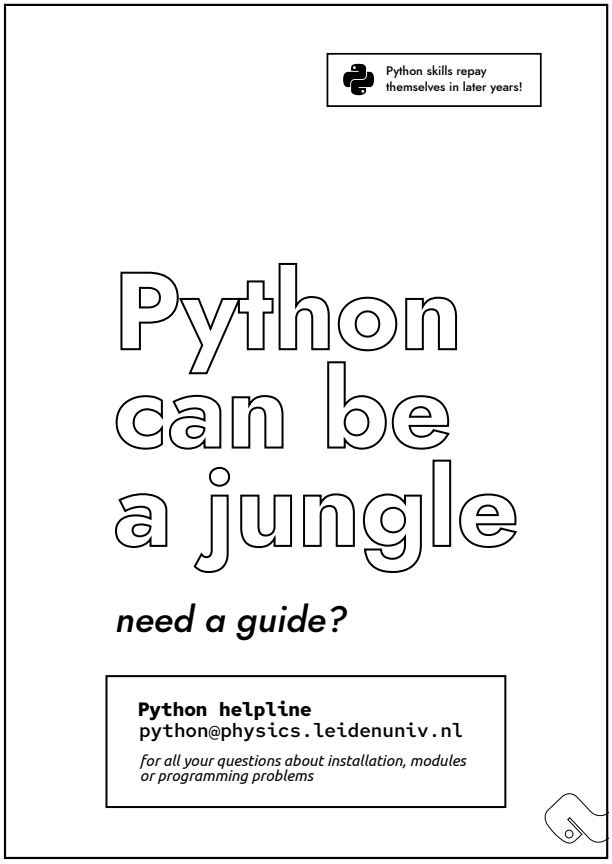
3. Schrijf code die automatisch alle relevante databestanden in een map inleest en er een fit op uitvoert.

4. Gegeven is de gemmiddelde-functie:

$$\bar{x}(x_1, ..., x_n) = \sum_{i=1}^{n} \frac{x_i}{n} \tag{5}$$

   Laat $x_1, ..., x_n$ datapunten, allen met dezelfde fout σ. Dus $x_i \pm \sigma$. Bereken de fout met 3 op het gemmiddelde $\bar{x}$

5. Bekijk de documentatie van scipy.stats.ttest_ind. Bepaal of de volgende twee lijsten met getallen significant verschillen, gebruikmakend van $p = 0.95$. `a = [19, 11, 8, 7, 17, 9, 2, 5]`, `b = [10, 1, 11, 17, 8, 14, 5, 2]`


Python skills repay
themselves in later years!
Python
can be
a jungle
need a guide?
Python helpline
python@physics.leidenuniv.nl
for all your questions about installation, modules
or programming problems