

## Opdrachten bij workshop ‘Fitten in Python’

### Opdracht 1

De positie van een auto die met constante snelheid beweegt wordt gemeten als functie van de tijd. De data staat in het bestand ‘Auto.csv’.

- a. Maak een grafiek van de positie van de auto versus de tijd.

Als de auto inderdaad met constante snelheid beweegt, dan geldt voor de positie als functie van de tijd:

$$x_{auto}(t) = x_0 + vt$$

Om in Python te kunnen fitten moeten we een Python functie maken van deze vergelijking.

- b. Schrijf een Python functie **PosAuto(t, x0, v)** die als input de parameters **x0** en **v** en een array **t** van tijdswaarden neemt, en als output een array van posities geeft.
- c. Check dat de bovenstaande Python functie correct werkt door deze functie in dezelfde grafiek als de auto-data te plotten. Kies daarbij de parameters **x0** en **v** zó dat de Python functie al enigszins lijkt op de data.
- d. Gebruik nu de Python functie **curve\_fit** om de optimale parameters **x0** en **v** te vinden. Check dat je alles correct hebt gedaan door de optimale fit in dezelfde grafiek als de auto-data te plotten.

### Opdracht 2

De spanning over een condensator wordt gemeten terwijl de condensator wordt opgeladen. Deze metingen staan in ‘Condensator.csv’. De theoretische uitdrukking voor de tijdsafhankelijke spanning is:

$$V(t) = V_0(1 - e^{-t/\tau})$$

- a. Maak een grafiek van de metingen.
- b. Schrijf een Python functie voor de theoretische uitdrukking en plot deze in dezelfde grafiek als de meetdata. Kies de vrije parameter weer zó dat de theoretische kromme de data ongeveer beschrijft.
- c. Gebruik **curve\_fit** om de optimale fitparameters te vinden.
- d. Plot de fit in dezelfde grafiek als de data.

### Opdracht 3

Het spectrum van een diodelaser, zoals de laser die jullie in dit project gebruiken, wordt redelijk goed beschreven door een zogenaamde Gauss kromme:

$$I(\lambda) = I_0 e^{-(\lambda - \lambda_0)^2 / \sigma^2}$$

In het bestand 'LaserSpec.csv' staat de data van een experimenteel gemeten spectrum.

- Maak een grafiek van het experimentele spectrum.
- Schrijf een Python functie voor de theoretische uitdrukking (Gauss kromme) en plot deze in dezelfde grafiek als de meetdata. Kies de vrije parameters weer zó dat de theoretische kromme de data ongeveer beschrijft.
- Gebruik **curve\_fit** om de optimale fitparameters te vinden.
- Plot de fit in dezelfde grafiek als de data. Hoe zie je aan deze grafiek dat het laserspectrum niet perfect wordt beschreven door de Gauss kromme?

Systematische afwijkingen tussen experimentele data en een fitvergelijking kun je eenvoudig visualiseren door een plot van de residuen (d.w.z. het verschil tussen de experimentele data en de fitvergelijking) te maken.

- Maak een grafiek van de residuen. Leg uit hoe je aan deze plot ziet dat de fitvergelijking geen perfecte beschrijving vormt van de data.

### Opdracht 4 🐍🐍

In deze opdracht gaan we kijken hoe goed het laserspectrum uit opdracht 3 te beschrijven is door een som van twee Gauss krommen.

- Gebruik het stappenplan uit opdracht 1 t/m 3 om de data uit 'LaserSpec.csv' te fitten aan een som van twee Gauss krommen.