

Addendum Syllabus 'Programmeren voor Natuur- en Sterrenkunde 2' 2016/17: Opdracht 10

May 12, 2017

Dit document bevat de 'standaard'-opdracht van opdracht 10 van de cursus 'Programmeren voor Natuur- en Sterrenkunde 2', cursusjaar 2016/2017 en is een addendum op de syllabus, waarvan de volledige versie te vinden is op <https://numnat.mprog.nl>.

Het is de bedoeling dat je of deze opdracht maakt, of één van de drie suggesties, of zelf een opdracht verzint en voorlegt aan de docent. Lees de volledige instructies in de syllabus.

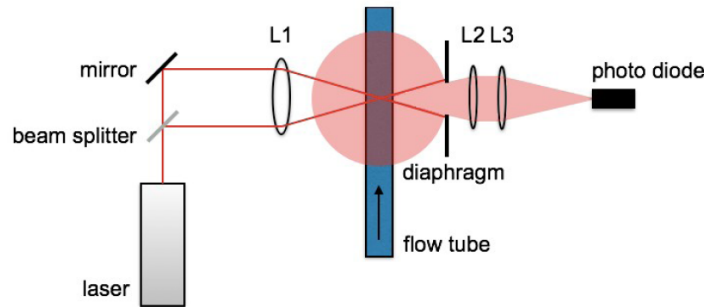


Figure 1: LDA-opstelling. Deeltjes die door het kruispunt van de twee coherente laserbundels gaan verstrooien het licht (met $\lambda = 632,8 \text{ nm}$) met een zwevingsfrequentie f die door de photodiode wordt gedetecteerd. De snelheid v van de deeltjes is gelijk aan $v = f \frac{\lambda}{2 \sin \Theta}$

1 Fitten van Data

1.1 Inleiding

Deze opdracht betreft het verwerken (analyseren) van gegevens (data) uit een experiment. De gegevens die gebruikt zullen worden zijn afkomstig van het practicum-experiment *Laser-Doppler Anemometrie*. Eerst een introductie (of herinnering) aan het experiment en vervolgens de opdrachten.

1.2 Laser-Doppler Anemometrie

De experimentele opstelling is schematisch weergegeven in figuur 1.

Via de LDA-methode worden de frequenties bepaald van het verstrooide licht op een aantal posities in een ronde buis en daarmee de snelheid van het stromende water op die posities. Als een deeltje door het kruispunt van de twee laserbundels stroomt, dan verschijnt er in het verstrooide licht een burst (zie figuur 1). Als de stroming laminair is, dan is de verwachting dat het stromingsprofiel een parabool is met snelheden gelijk aan nul op de positie van de wanden. Er wordt een snelheidsprofiel bepaald dwars op het midden van de buis. Dus langs een middellijn. De buis kan verschoven worden in de x-richting. Er zijn op 16 posities metingen gedaan tussen $x = 5,5 \text{ mm}$ (vlak bij de ene wand) tot $x = 21 \text{ mm}$ (een paar mm van de andere wand). Op elke posities is de frequentie van een ongeveer 1000 deeltjes (bursts) gemeten. Er zijn 16 files beschikbaar die voor elk van deze posities de gemeten waarden bevatten. Het eerste element van de kolom geeft de x-positie van de buis weer.

1.3 Histogram

Meetgegevens kunnen op vele verschillende manieren worden weergegeven. Herhaalde metingen van een grootte zullen niet altijd dezelfde waarden geven. Ze zijn verdeeld volgens een bepaalde kansverdeling. Om dit weer te geven wordt vaak het

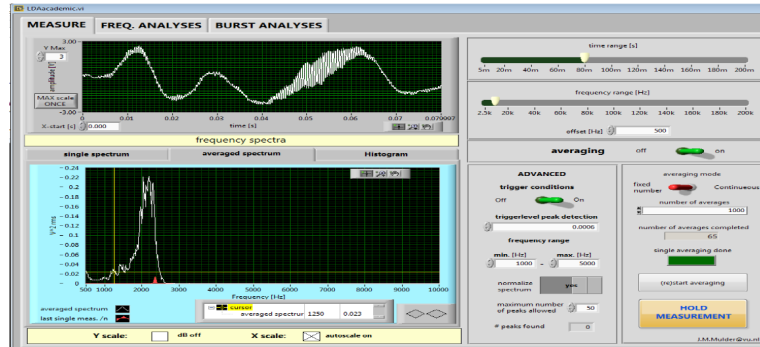


Figure 2: Burst in het verstrooide licht. Een deeltje stroomt door het kruispunt van de twee laserbundels en het oscillerend licht verstrooit met een frequentie evenredig met de snelheid.

bereik van de gemeten grootte (x) onderverdeeld in 'klassen', die elk een interval in dat bereik voorstellen, bijvoorbeeld ($a \leq x < b$, $b \leq x < c$, $c \leq x < d$). Een histogram is een diagram met kolommen met een oppervlak die proportioneel is aan de (relatieve) frequenties van deze klassen. Histogrammen kunnen op verschillende manieren worden genormaliseerd. Ze (dat wil zeggen, elke kolom) kunnen bijvoorbeeld door het totaal aantal meting worden gedeeld zodat het totale oppervlakte 1 is. Een histogram kan ook omgezet worden in een waarschijnlijkheidsdichtheid door vervolgens elke kolom door de breedte (van de klasse) te delen.

Het pakket *matplotlib* kan je helpen met het plotten van histogrammen, bijvoorbeeld met de functie `matplotlib.pyplot.hist(x, bins=None, range=None, normed=False, weights=None, cumulative=False, bottom=None, hist-type='bar', align='mid', orientation='vertical', rwidth=None, log=False, color=None, label=None, stacked=False, hold=None, data=None, **kwargs)`. Let vooral op de argumenten, *bins* waarmee je het aantal klassen langs de x-as kan instellen en *range* waarmee je het bereik kan instellen. Het argument *normed* kan helpen om het histogram te normaliseren.

1.4 Verfris je kennis: De χ^2 als compatibiliteitsmaat

In module 3 van *Programmeren voor Natuur- en Sterrenkunde* heb je kennisgemaakt met de χ^2 . Verfris je kennis van deze module, te vinden op <http://prognsmprog.nl> (onder 'Archive'). Doe de opdrachten niet!

1.5 Bibliotheek voor het fitten van functies

Het bepalen van de vrije-parameters van je model aan de hand van je meet-gegevens heet 'fitten'. Hier heb je ook al kennis mee gemaakt in module 3 van *Programmeren voor Natuur- en Sterrenkunde*. Verschillende python bibliotheken hebben methoden beschikbaar voor het fitten van functies. Als voorbeeld kan je naar het *SciPy* pakket kijken. Documentatie van dit pakket kan onder andere via <https://scipy.org>

gevonden worden.

Je kan natuurlijk ook je eigen fit procedure schrijven die bijvoorbeeld een polynoom volgens de 'kleinste-kwadraten' (least-squares) methode fit, of misschien zelfs met via 'maximum-likelihood' methode.

1.6 De gegevens

De meetgegevens kunnen op de website <http://numnat.mprog.nl> gevonden worden onder het kopje 'Materiaal'. In het zip-bestand bevinden zich 17 files. Elk van de files geeft de metingen op een enkele positie. De positie, in mm, is gegeven in de naam van het bestand. Zo staan in bestand 'data_12.txt' de metingen op positie $x=12$ mm. **Let op:** de eerste regel van het de gegevens bestanden bevatten ook de positie!

1.7 De opdracht

In de opdracht is het de bedoeling om de meetgegevens te reduceren en te analyseren om uitspraken te doen over de meetgegevens en het stromings profiel. Centraal in deze opdracht staan de begrippen **Validiteit** ('hoe goed is mijn model?') en **betrouwbaarheid** ('kan ik op mijn meetgegevens aan?'). Om je door het proces heen te helpen, is de opdracht in een aantal onderdelen opgedeeld.

Let op, dit is een vrije opdracht, dat betekent dat veel zelfstandigheid en initiatief wordt gevraagd.

1.7.1 Onderdeel 1

Maak een programma dat de gegevens bestanden inleest en voor elk van de posities een histogram maakt die de verdeling van de metingen goed weergeeft. Bereken voor elk van de posities het gemiddelde en de standaard deviatie.

1.7.2 Onderdeel 2

Teken in elk van de verdelingen uit onderdeel 1 een normaal verdeling met de gevonden gemiddelde en standaard deviatie. Bereken voor elke verdeling de gereduceerde χ^2 . Dat laatste getal is gelijk aan χ^2/NDOF , waarbij NDOF staat voor het aantal vrijheids graden (Number of degrees of freedom), wat gelijk is aan het aantal datapunten (bins in je histogram!) minus het aantal fit-parameters. Gebruik als fout op het aantal punten in een bin (N) de waarde \sqrt{N} .

1.7.3 Onderdeel 3

Plot de frequentie als functie van positie. Onderzoek of het stromingsprofiel een parabool is. Schrijf daarvoor een fitprocedure voor een parabool aan de datapunten en reken de gereduceerde χ^2 uit. Plot deze. **N.B.**, : Het is bij deze opdracht zeker

toegestaan bibliotheek functies zoals bijvoorbeeld bovend vermeld te gebruiken. Het helemaal zelf schrijven van een fitprocedure kan zeer uitdagend zijn.

1.7.4 Onderdeel 4

Bereken residuen tussen de ge-fitte parabool uit 3 en de data-punten. Plot deze als functie van de positie.

1.7.5 Onderdeel 5

Bediscussieer kort de gevonden resultaten. Doe in ieder geval uitspraken over de validiteit van het model en de betrouwbaarheid van de meetgegevens zoals boven genoemd. Je kan bijvoorbeeld ook proberen een andere functie aan het stromings profiel fitten, of de datapunten op een andere manier reduceren.