Programowanie 1R – Ćwiczenia 11
Analiza danych pomiarowych (biblioteki `scipy` i `pandas`)

Mikołaj Marcinkowski

26 maja 2026

Zadania możesz wykonywać jako oddzielne skrypty `.py` lub w notatniku jupytera `.ipynb`, chyba że treść zadania wskazuje inaczej.

Zadanie 1. Układ RC

W złączonych plikach `uklad_RC.csv` i `uklad_RC.xlsx` (oba zawierają te same dane) zawarte są dane z pomiaru napięcia na kondensatorze ładowanym przez opornik. Dane zawarte są w trzech kolumnach: czas pomiaru (wyrażony w sekundach), napięcie (wyrażone w woltach), niepewność napięcia. Korzystając a bilioteki `pandas`, załaduj dane pomiarowe. Następnie, wykorzystując funkcję `curve_fit` z biblioteki `scipy.optimize`, dopasuj do tych danych krzywą teoretyczną:

$$U(t) = U_0\Big(1 - e^{-(t-t_0)/\tau}\Big)\,, \tag{1}$$

gdzie t to czas, U to napięcie, a U_0, τ i t_0 to parametry modelu. τ to charakterystyczny czas ładowania kondensatora i zależy od jego pojemności i oporu opornika.

Narysuj na wykresie punkty pomiarowe z niepewnościami oraz dopasowaną krzywą.

Zadanie 2. Plamka Airy'go

Gdy przepuścimy wiązkę światła przez okrągły otwór, utworzy ono w wyniku dyfrakcji obraz zwany Plamką Airy'go, którego natężenie promieniowania I zadane jest wzorem [1]:

$$I(x) = I_0\left(\frac{2J_1(x)}{x}\right)^2\,, \tag{2}$$

gdzie I_0 to maksymalne natężenie, x to unormowana odległość od środka plamki (unormowana odległością ekranu od otworu, długością fali, średnicą otworu itp.), a J_1 to funkcja Bessela pierwszego rodzaju rzędu 1.

W złączonych plikach `Airy.csv` i `Airy.xlsx` (oba zawierają te same dane) zawarte są (spreparowane) dane z pomiaru jasności. Dane są zawarte w trzech kolumnach: odległość od środka (jednostki dowolne), intensywność (jednostki dowolne), niepewność intensywności. **W szczególności odległość nie jest unormowana, jak x.** Dopasuj do tych punktów model teoretyczny. Narysuj na wykresie punkty pomiarowe z niepewnościami oraz dopasowaną krzywą.

Biblioteka `scipy.special` zawiera wiele gotowych funkcji specjalnych, kompatybilnych z `numpy`'owymi arrayami.

Zadanie 3. Szybka transformacja Fouriera (FFT)

Dyskretna transformacja Fouriera y_k ciągu x_n ($n \in \{0, 1, \ldots, N\}$) zadana jest wzorem [2]:

$$y_k = \sum_{n=0}^{N-1} x_n e^{-2i\pi kn/N}\,, \tag{3}$$

i jest zdyskretyzowanym odpowiednikiem transformacji Fouriera. *Szybka transformacja Fouriera (FFT)* jest algorytmem wykorzystującym różne symetrie powyższego wzoru, by szybko obliczyć ją dla wielu częstotliwości k/N jednocześnie.

W załączonym pliku `gitara.vaw` zawarte jest dwusekundowe nagranie dźwięku struny gitary z częstotliwością próbkowania 44100 Hz. Korzystając z biblioteki `scipy.fft`, znajdź częstotliwości modów drgań tej struny. Do wczytania pliku możesz wykorzystać funkcję `scipy.io.wavfile.read`. Przy analizie widma Fourierowskiego może się też przydać biblioteka `scipy.signal`, zwłaszcza funkcja `find_peaks`.

Narysuj wykres widma tego dźwięku.

Zadanie 4. Widmo α

W złączonych plikach `histogram_alfa.csv` i `histogram_alfa.xlsx` (oba zawierają te same dane) zawarte są dane z pomiaru widma energetycznego cząstek α emitowanych przez pewną próbkę, w postaci histogramu. Dane te zawarte są w trzech kolumnach: dolny próg binu (wyrażony w keV), górny próg binu (wyrażony w keV), liczba zliczeń.

Do zawartych linii widmowych dopasuj krzywe Gaussa i znajdź ich środki oraz szerokości. Narysuj wykres słupkowy z histogramem oraz naniesionymi krzywymi Gaussa. Do narysowania słupków możesz skorzystać z `ax.bar` z biblioteki `matplotlib`.

Bibliografia

[1] M. Born i E. Wolf, *Principles of optics*, en, 7 wyd. Cambridge, England: Cambridge University Press, grud. 2019.

[2] William H. Press, B. P. Flannery, S. A. Teukolsky i W. T. Vetterling, *Numerical recipes in FORTRAN 77: Volume 1 of FORTRAN numerical recipes volume 1*, 2 wyd. Cambridge, England: Cambridge University Press, wrz. 1992.