

Zadanie projektowe: Czastki, histogramy i korelacje

Detektory czastek bardzo czesto rejestruja ich moment przybycia do detektora z doskonała precyzja. Moment przybycia czastki jest nastepnie zapisywany cyfrowo przy pomocy urzadzeń zwanych time-to-digital converter albo time-tagger.

Dla ustalenia uwagi, zajmijmy sie fotonami które dla energii rzędu 1 eV moga być rejestrowane fotodiodami lawinowymi (APD) albo detektorami z nadprzewodzących nanodrutów (SNSPD).

Napisz klase `Photons` przechowujaca tagi czasowe zarejestrowanych przez detektor fotonów. Klasa powinna zawierac nastepujace metody:

- konstruktor `__init__` budujacy obiekt klasy na podstawie listy tagów czasowych
- metode klasy `mean` pozwalajaca obliczyc srednia liczbe zarejestrowanych fotonów na interwał czasowy `dt`. Interwał czasowy powinien być dodatkowym argumentem funkcji.
- metode klasy `rate` która zwraca liste liczb fotonów n zarejestrowanych w kolejnych interwałach czasowych. Ponownie jak poprzednio interwał czasowy powinien być dodatkowym argumentem funkcji.
- metode klasy `g2` obliczajaca dla danej sekwencji funkcje korelacji drugiego rzędu, zdefiniowana jako srednia zmiennej losowej $n(n-1)$ podzielona przez kwadrat wartosci sredniej n
- metode `hist` generujaca histogram zliczeń, tzn. wykres pokazujacy w ilu oknach czasowych zarejestrowano k fotonów. Funkcja powinna przyjmowac rozmiar okna czasowego `dt` jako dodatkowy argument. Do narysowania histogramu użyj `matplotlib`.
- operator `__add__` pozwalajacy sekwencyjnie polaczyc dwa obiekty klasy `Photons`
- operator `__getitem__` pozwalajacy wybrac k -ty foton z listy zwracajac jego wartosc (czas) albo wybrac fragment `slice` (np. `[4 :: 10]`) zwracajac zredukowany obiekt klasy `Photons`.

Napisz program który laduje przykladowe dane z pliku [link](#). Dane zawieraja 500 time-tagów (fotonów) w całkowitym pomiarze trwajacym 10000 jednostek czasowych (dla ustalenia uwagi - nanosekund). Dla danych oblicz srednia liczbe fotonów na okno o dlugosci `dt = 100 ns`, funkcje korelacji, wyrysuj histogram i wynik funkcji `rate`. Zaprezentuj dzialanie operatorów `__add__` i `__getitem__` na przykladowych sekwencjach.

Możesz też wygenerowac swoje własne dane, np. używajac funkcji `np.random.randint` z `numpy` i sortujac uzyskane wyniki przy pomocy `np.sort`.

Pytania i rozwiazania prosze kierowac na mparniak@fuw.edu.pl.