

## Metody Probabilistyczne i Statystyka - Lista 1

### Wojciech Typer

Kody źródłowe zostały napisane w języku Rust. Wykresy zostały wygenerowane w Pythonie poprzez bibliotekę Matplotlib. Użyto generatora liczb pseudolosowych Mersenne Twister. Każdy kod źródłowy generuje dane liczbowe, a następnie zapisuje je do pliku tekstowego, z którego następnie generowany jest wykres.

Na przedstawionych wykresach przerywana pomarańczowa linia oznacza dokładny wynik całki, czerwone punkty oznaczają średnią uzyskaną z  $k$  powtórzeń dla danej ilości generowanych punktów  $n$ .

Odczytując wykresy możemy zauważyć tendencję, że wraz ze wzrostem ilości generowanych punktów  $n$  oraz niezależnych powtórzeń  $k$  rośnie dokładność przybliżenia zadanych całek, bądź liczby  $\pi$ .

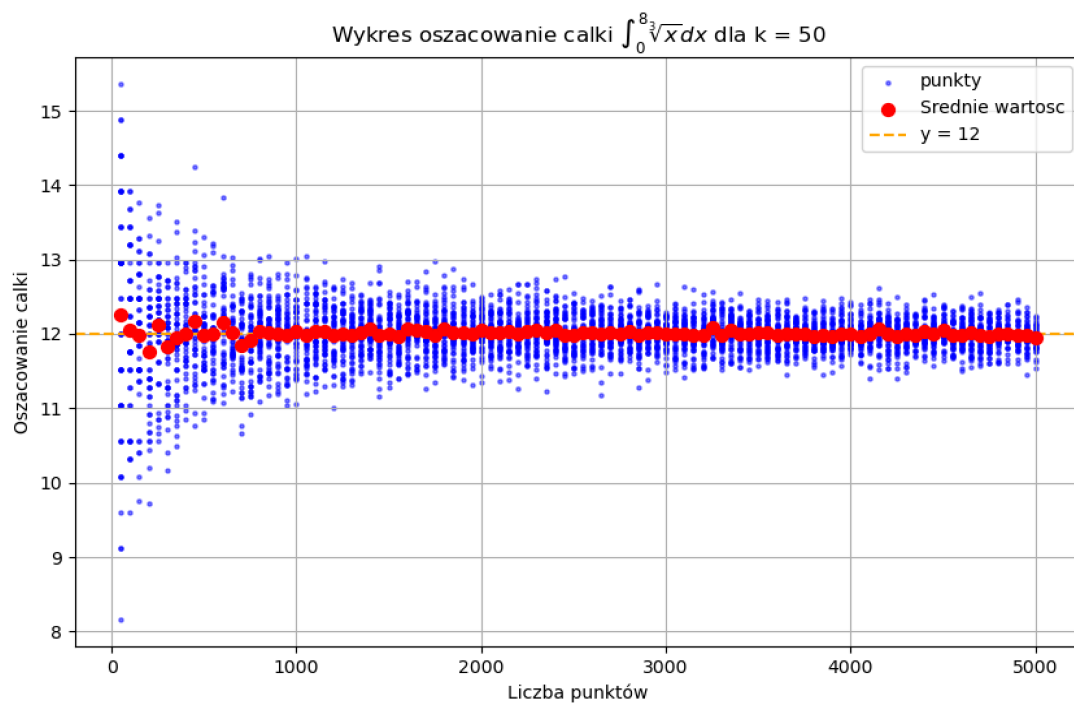
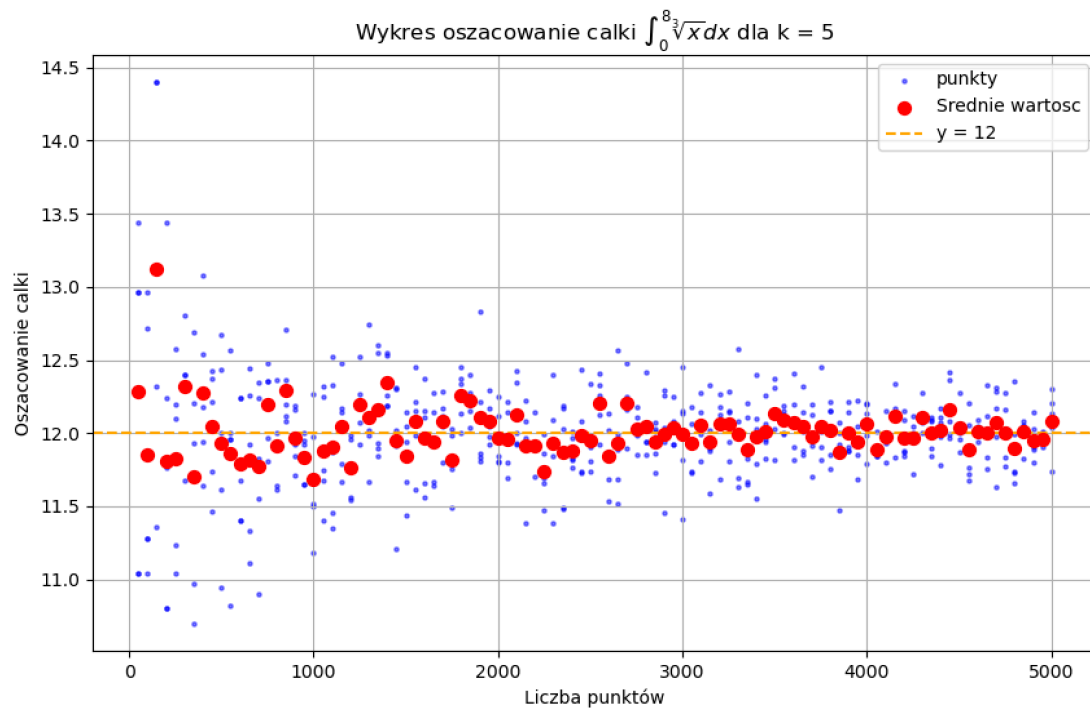
Dzieje się tak, ponieważ większa ilość próbek pozwala na lepsze pokrycie obszaru całkowania.

Z wykresów możemy odczytać, że przy  $k = 5$  wyniki bardziej się rozpraszają, a ich zmienność (fluktuacja) jest większa. Natomiast dla  $k = 50$  krzywe są bardziej "wygładzone" i bliższe wartościom oczekiwanym.

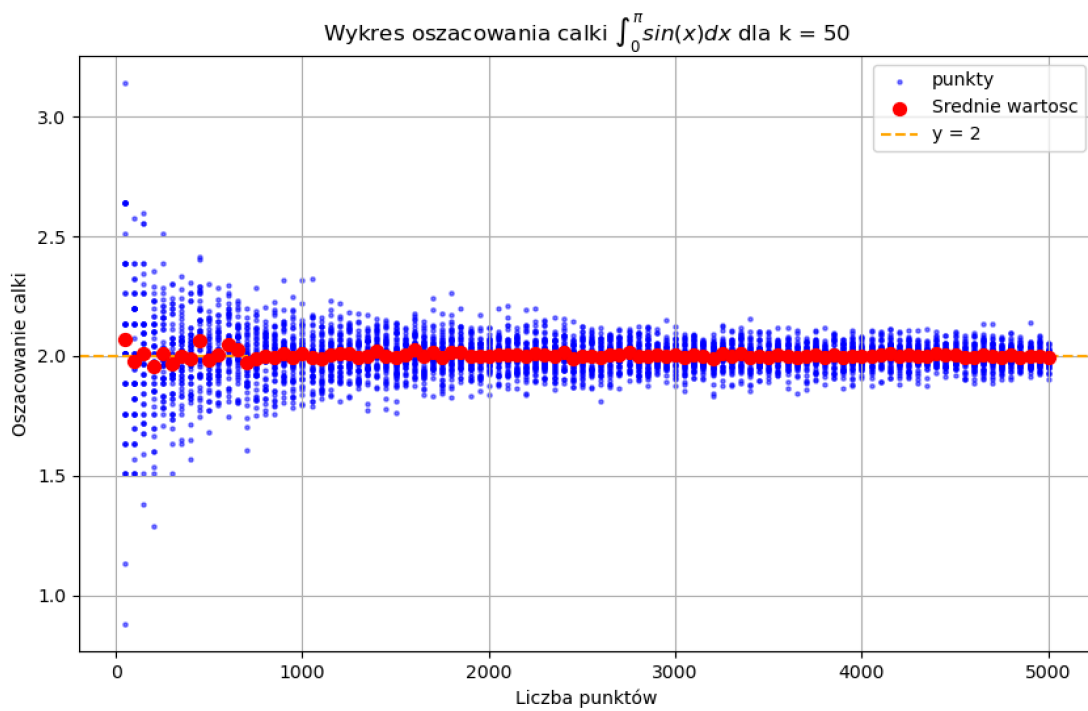
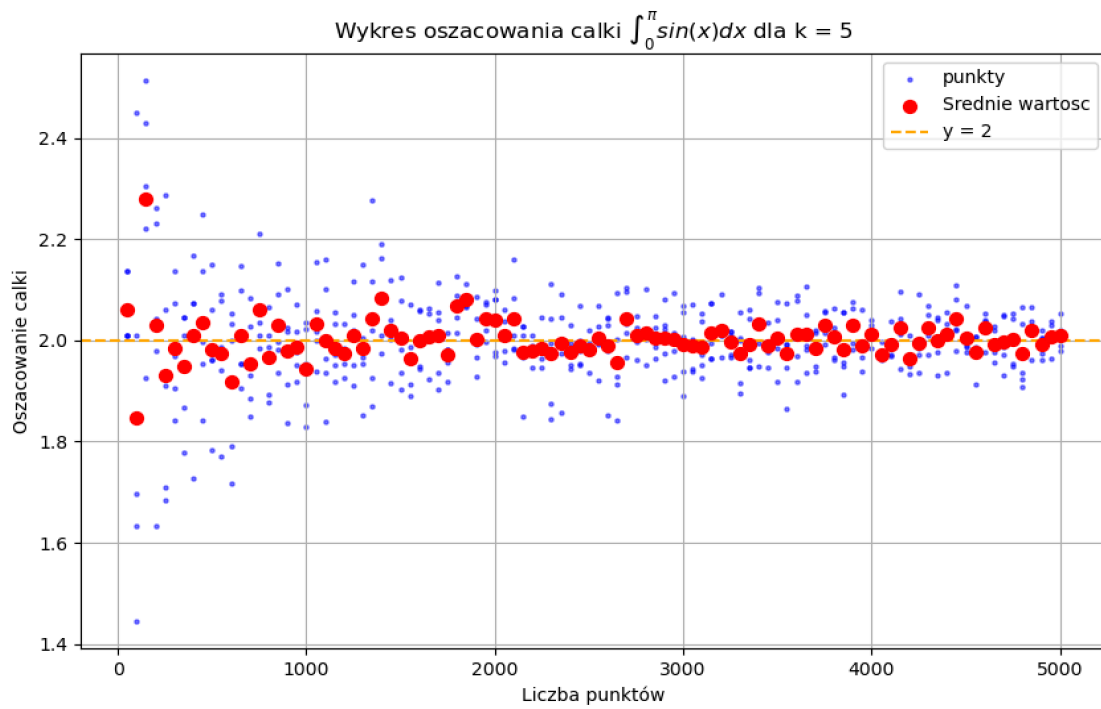
Warto również zwrócić uwagę na długość czasu wykonania programu - im większa ilość  $k$  niezależnych powtórzeń, tym wymagana długość czasu na ukończenie programu rośnie.

Trzeba o tym pamiętać pisząc programy bardziej wymagające pod względem mocy obliczeniowej, zachowując przy tym balans pomiędzy dokładnością oszacowań i długością wykonania programu.

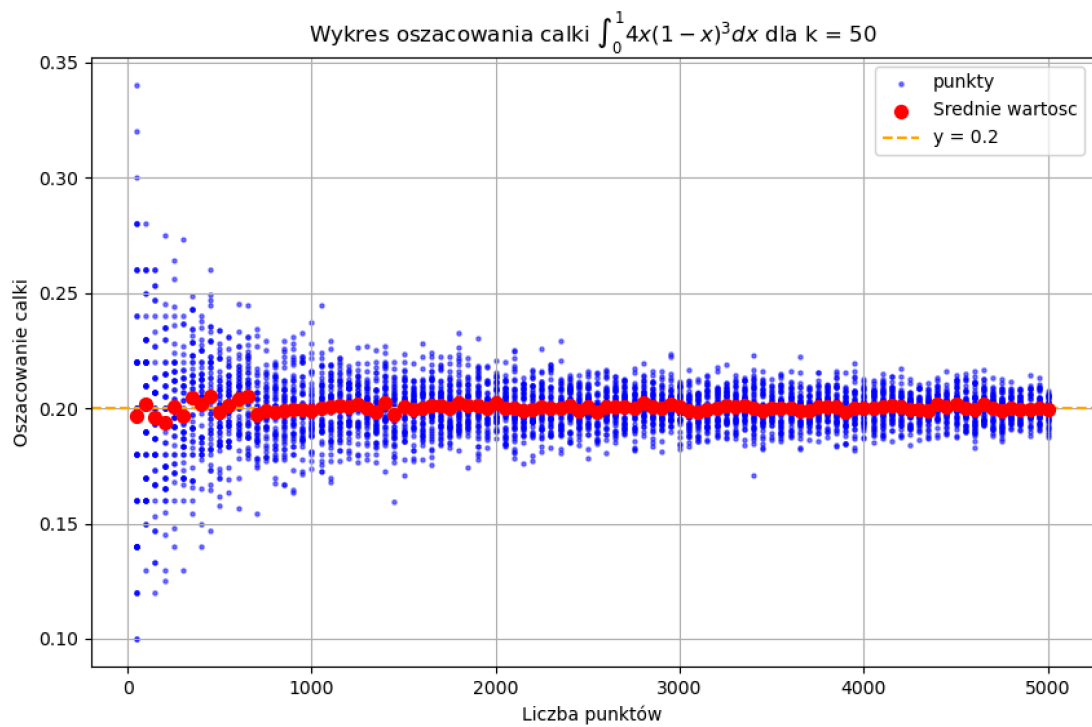
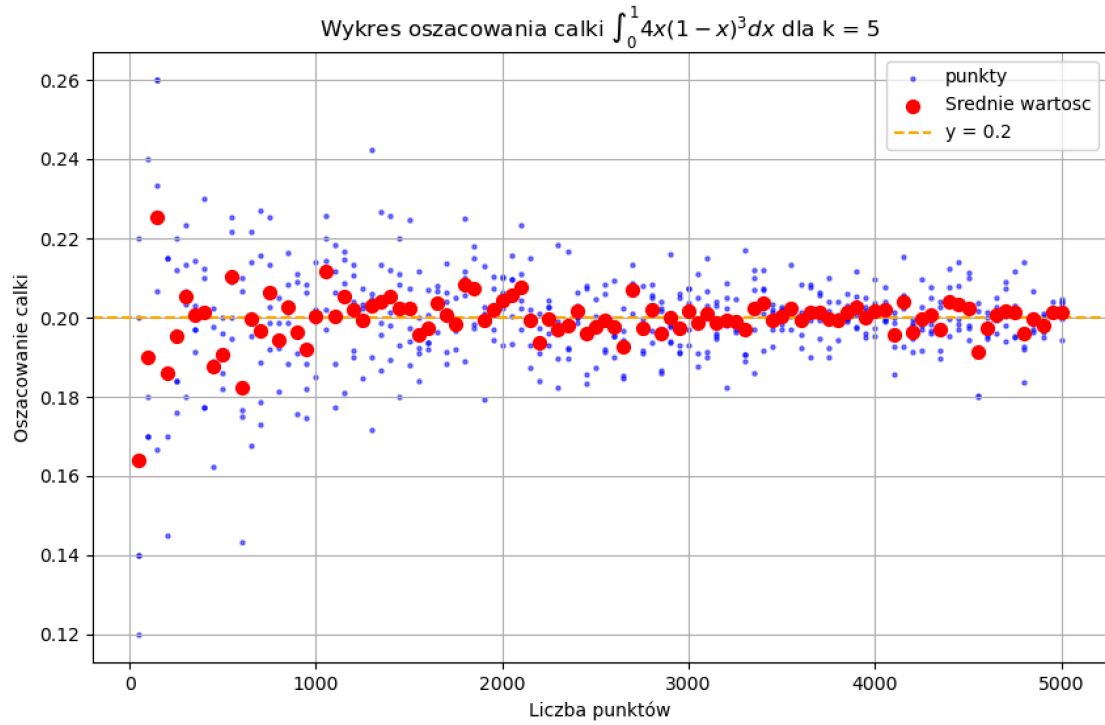
$$\int_0^8 \sqrt[3]{x} dx$$



$$\bullet \int_0^{\pi} \sin(x) dx$$

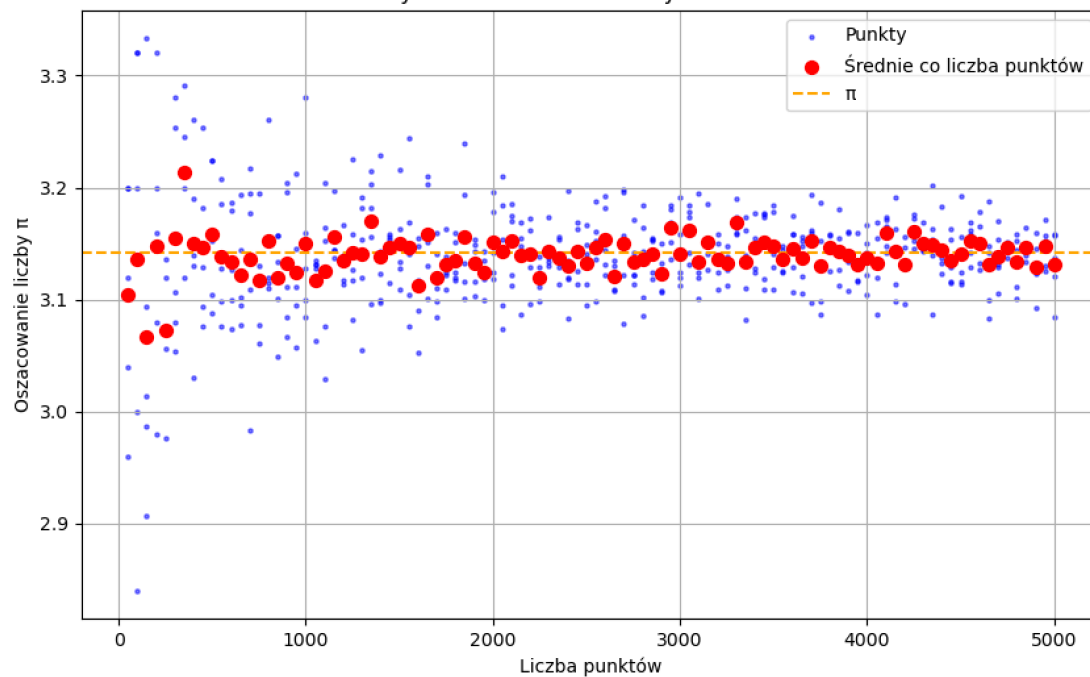


$$\bullet \int_0^1 4x(1-x)^3 dx$$



$\pi$

Wykres oszacowania liczby  $\pi$  dla  $k = 5$



Wykres oszacowania liczby  $\pi$  dla  $k = 50$

