

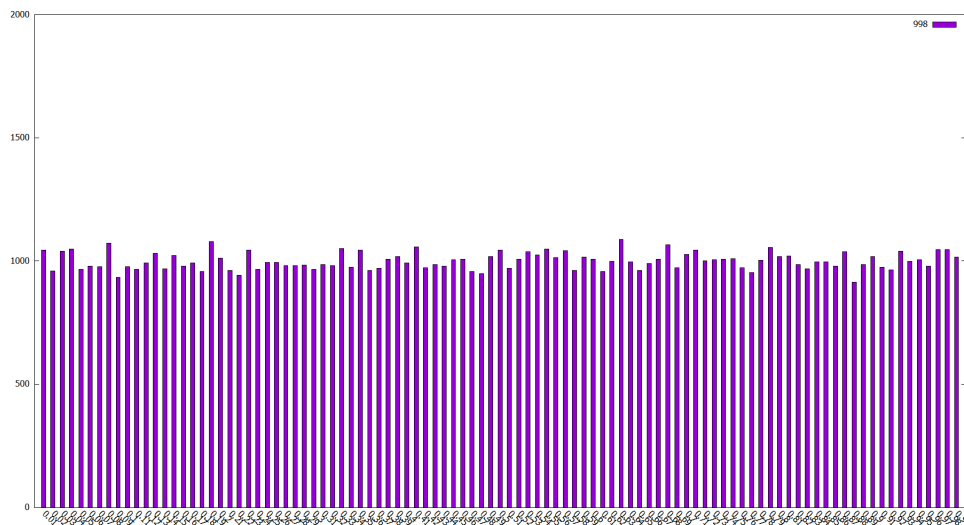
Spacer losowy  
Modelowanie komputerowe  
Lista 2

Łukasz Chmielowski (307713)

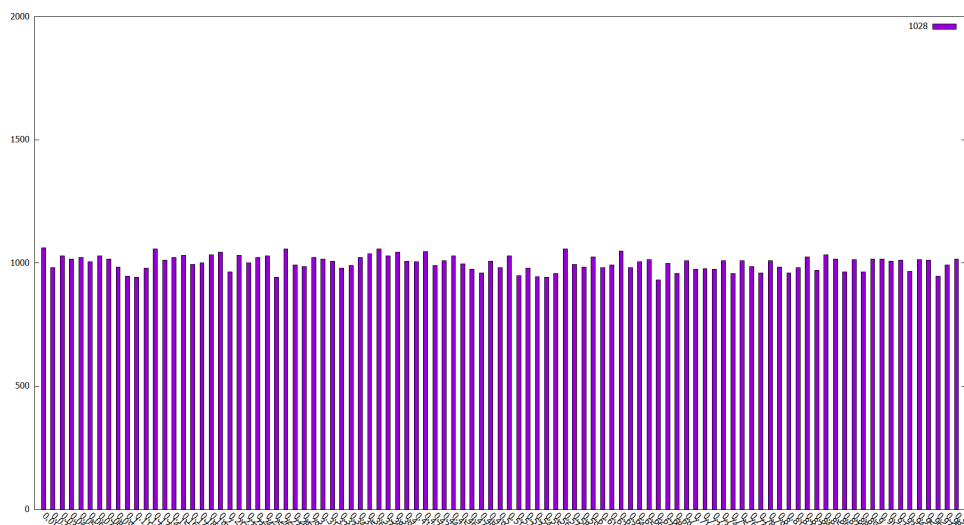
2023

# 1 Zadanie pierwsze

Na początku mieliśmy sprawdzić jakość generatorów liczb losowych *rand()* oraz Mersenne-Twister. W tym celu napisałem program sprawdzający oba te generatory (kod został dołączony do sprawozdania w pliku *spacer.cpp*). Program wykonuje 100'000 symulacji dla obu generatorów, do których dałem takie samo ziarno na początku, które wynosiło 1234. Po przeprowadzeniu symulacji, zliczyłem wystąpienie każdej liczby z przedziału 0.01 - 0.99 z krokiem 0.01. Oba rozkłady są przedstawione na wykresach 1 oraz 2.



Rysunek 1: Rozkład występowania liczb w algorytmie *rand()*.



Rysunek 2: Rozkład występowania liczb w algorytmie Mersenne-Twister.

Następnie wyliczyłem błąd korzystając ze wzoru:

$$s = \sqrt{\frac{\sum_{i=1}^k (x_i - \Delta x)^2}{n}}$$

$$e = \frac{s}{\sqrt{n}}$$

Gdzie:

$s$  jest odchyleniem,

$e$  jest błędem.

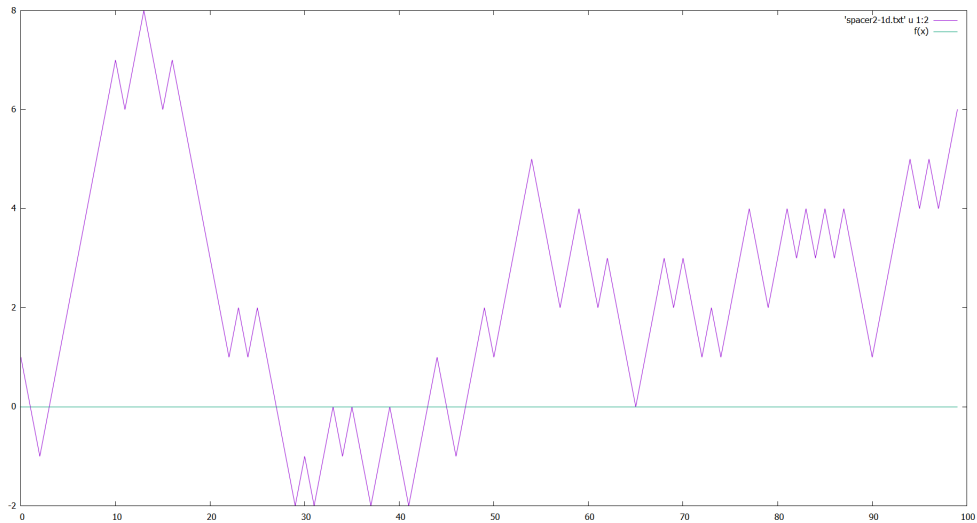
Po wyliczeniach otrzymałem następujące wyniki:

	<i>rand()</i>	Mernenne-Twister
Błąd	0.0009129343605814678	0.0009137774223452954
Odchylenie	0.288695193366688	0.2889617929048774

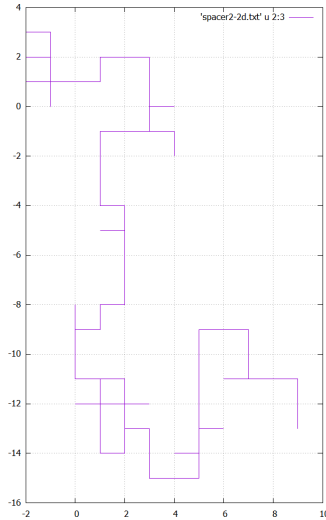
Możemy zauważyć, że niepewności obu tych generatorów są do siebie bardzo zbliżone.

## 2 Zadanie drugie

W drugim zadaniu mieliśmy za zadanie narysować trajektorie spaceru losowego w jednym oraz dwóch wymiarach. Kod generujący te spacery został zawarty w pliku *spacer2.cpp*. Po wykonaniu symulacji otrzymałem następujące rezultaty:



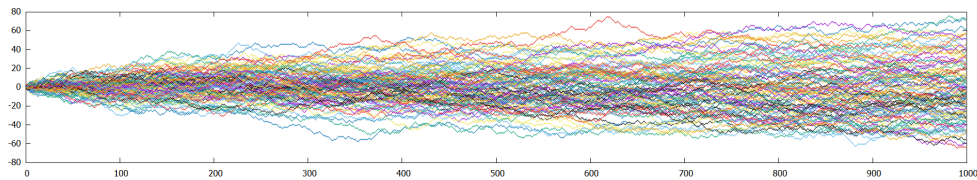
Rysunek 3: Spacer losowy w jednym wymiarze.



Rysunek 4: Spacer losowy w dwóch wymiarach.

### 3 Zadanie trzecie

W zadaniu trzecim napisałem kod (w pliku *spacer3.cpp*) wykonujący 1000 kroków spaceru losowego w jednym wymiarze. Spacerów wykonałem 100. Położenia spacerów prezentują się następująco:



Rysunek 5: 100 spacerów losowych w jednym wymiarze po 1000 krokach.

Następnie porównałem, czy na końcu większe prawdopodobieństwo jest, że dziecko znajdzie w odległości 1 czy 30 jednostek od punktu startowego. W tym celu sprawdziłem ile spacerów skończyło się w odległości 30 oraz -30. Dla obu tych wartości był tylko jeden. Z kolei dla odległości 1 oraz -1 nie było żadnego. Jest to logiczne, ponieważ po parzystej licz-

bie krótków niemożliwym jest znaleźć się w odległości nieparzystej liczby jednostek. Czyli większe prawdopodobieństwo jest znalezienia się w odległości 30 jednostek.

## 4 Zadanie czwarte

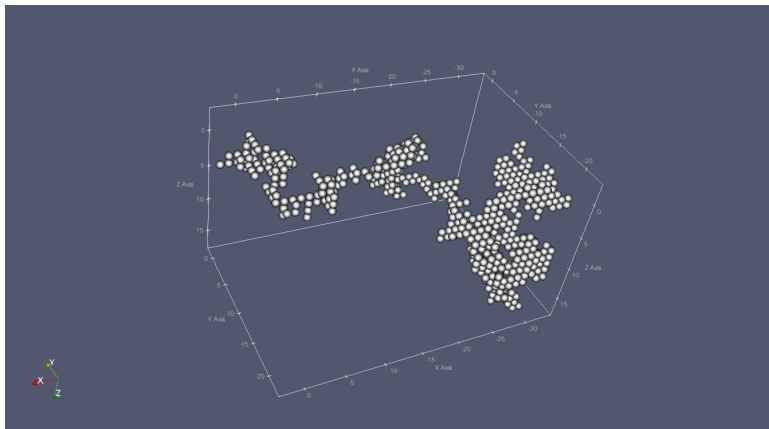
W zadaniu czwartym należało sprawdzić prawdopodobieństwo, że dziecko samo wróci do domu. Wiemy, że dziecko może wrócić tylko w krokach parzystych, a więc oznaczmy liczbę kroków jako  $2n$ . Powołując się na dokument [1], prawdopodobieństwo, że dziecko wróci na start możemy opisać wzorem:

$$p = \binom{2n}{n} (pq)^n \quad (1)$$

gdzie:  $p$  jest prawdopodobieństwem kroku do przodu,  $q$  jest prawdopodobieństwem kroku do tyłu.

## 5 Zadanie piąte

W zadaniu piątym należało wygenerować i narysować trajektorię spaceru losowego w trzech wymiarach. Kod, którego użyłem do wygenerowania tego spaceru został dołączony do sprawozdania (plik o nazwie *spacer5.cpp*). Po wykonaniu symulacji składającej się ze 100 kroków, użyłem programu *ParaView* do narysowania spaceru:



Rysunek 6: 100 spacerów losowych w jednym wymiarze po 1000 krokach.

## Literatura

[1] Random Walk

[https://rinterested.github.io/statistics/random\\_walk.html](https://rinterested.github.io/statistics/random_walk.html)