Generatory liczb pseudolosowych o zadanym rozkładzie w jednym wymiarze

Metody Monte Carlo w fizyce
Nanoinżynieria materiałów

Łukasz Ruba

Akademia Górniczo-Hutnicza
im. Stanisława Staszica w Krakowie

5 marca 2024

Spis treści

Cel ćwiczenia

Celem ćwiczenia było skonstruowanie trzech generatorów liczb losowych przy użyciu algorytmów rozkładu złożonego, łańcucha Markowa oraz metody eliminacji. Uzyskane wyniki zobrazowano w postaci histogramów oraz policzono dla każdego test χ^2.

1 Wstęp teoretyczny

Generatory liczb losowych można podzielić na dwie kategorie - generatory fizyczne oraz komputerowe. Największą zaletą fizycznych generatorów jest możliwość uzyskania prawdziwej losowości np. badając zmienność prądu występującą w rezystorze w odpowiedzi na fluktuacje termiczne. Jednocześnie trudnym jest utrzymać optymalne warunku pracy generatora fizycznego tak, aby mieć pewność że czynniki zewnętrzne takie jak wilgoć nie wpłyną drastycznie na przeprowadzany pomiar. Największym jednak problemem generatorów fizycznych jest ich niewystarczająca szybkość w generowaniu kolejnych liczb. Żeby obejść te problemy stosuje się generatory komputerowe, które choć nie dają prawdziwie losowych wyników, pozwalają nam osiągnąć zadowalające rezultaty w optymalnym czasie.

Jakość generatora oraz to, do losowania jakich ciągów liczb się nadaje określają następujące parametry [1]:

- **okres generatora**, czyli długość ciągu znaków w którym nie występują powtórzenia tychże znaków,
- **ziarnistość ciągu**, czyli jaka jest średnia odległość między najbliższymi sąsiadami,
- **własności statystyczne** takie jak współczynnik autokorelacji, wartość średnia czy wariancja.

Istnieje wiele algorytmów pozwalających skonstruować generatory liczb losowych. W tym sprawozdaniu skupiono się na popularnych trzech opierających się na rozkładzie złożonym, łańcuchach Markowa oraz metodzie eliminacji.

Algorytm rozkładu złożonego bazuje na możliwości rozłożenia dystrybuanty zadanego rozkładu, zgodnie z wzorem:

$$F(x) = \sum_{i=1}^{N} g_i H_i(x), \qquad g_i \in R, \qquad H_i(x) : R \to R, \tag{1}$$

gdzie g_i to dystrybuanta rozkładu dyskretnego, a $H_i(x)$ dystrybuanty rozkładów podlegających superpozycji.

Znajdując funkcje odwrotne H_i^{-1} możemy otrzymać liczbę losową $X = x$.

W algorytmie łańcucha Markowa generuje się ciąg $\{X_0, X_1, X_2, ...\}$, gdzie związek pomiędzy kolejnymi elementami X_i i X_{i+1} określa się na podstawie prawdopodobieństwa przejścia, które spełnia warunek *detailed balance*:

$$T(X_{i+1}|X_i) = T(X_i|X_{i+1}) = \frac{1}{2\Delta}, \qquad \Delta \in [0, 1], \tag{2}$$

i prawdopodobieństwa akceptacji nowego stanu:

$$p_{acc} = min\left\{\frac{T(X_i|X_{i+1})f(X_{i+1})}{T(X_{i+1}|X_i)f(X_i)}, 1\right\}. \tag{3}$$

Chcąc wyznaczyć nowy element X_{i+1} stosuje się algorytm Metropolisa:

$$X_{i+1} = \begin{cases} x = X_i + (2U_1 - 1)\Delta, & \text{gdy } x \in [0, 1] \wedge p_{acc} \geq U_2 \\ X_i, & \text{w przeciwnym wypadku,} \end{cases} \tag{4}$$

gdzie U_1 i U_2 to zmienne losowe z przedziału [0,1] oraz $f(x)$ to funkcja gęstości prawdopodobieństwa.

W metodzie eliminacji wybieramy zmienną losową x w oparciu o funkcję $g(x)$ dla której posiadamy generator G, która ogranicza funkcję gęstości prawdopodobieństwa $f(x)$ od góry i spełniony jest warunek:

$$x = \begin{cases} U_1, & \text{gdy } f(U_1) \geq g(U_2) \\ \text{losujemy nowe wartości } U_1 \text{ i } U_2, & \text{w przeciwnym wypadku.} \end{cases} \tag{5}$$

2 Metodyka

Każdy z wymienionych algorytmów został zaimplementowany w język *C++* i wygenerowano nimi ciągi liczb pseudolosowych o mocy 10^6. Przyjęto funkcję gęstości prawdopodobieństwa:

$$f(x) = \frac{4}{5}(1 + x - x^3), \qquad x \in [0, 1], \tag{6}$$

oraz jej dystrybuantę:

$$F(x) = \frac{4}{5}\left(x + \frac{x^2}{2} - \frac{x^4}{4}\right). \tag{7}$$

Warunek uznania zmiennej losowej za poprawną dla rozkładu złożonego został wyrażony jako:

$$X = \begin{cases} U_2, & \text{gdy } g_1 \geq U_1 \\ \sqrt{1 - \sqrt{1 - U_2}} & g_1 < U_1, \end{cases} \tag{8}$$

gdzie g_1 wynosiło $\frac{4}{5}$.

Algorytm łańcucha Markowa został zrealizowany zgodnie z równaniami [3] i [4]. Policzono ciągi liczb dla dwóch wartości Δ równych kolejno 0.5 oraz 0.05.

Generator na podstawie metody eliminacji opierał się na równaniu [5], przy założeniu generatora G:

$$G = 1.15 \cdot U, \qquad U \in [0, 1]. \tag{9}$$

Następnie przy pomocy języka programowania *Python* policzono dla każdego ciągu wartość testu χ^2 ze wzoru:

$$\chi^2 = \sum_{i=1}^{k} \frac{(n_i - p_i N)^2}{p_i N}, \tag{10}$$

gdzie wartość prawdopodobieństwa p_i obliczono jako:

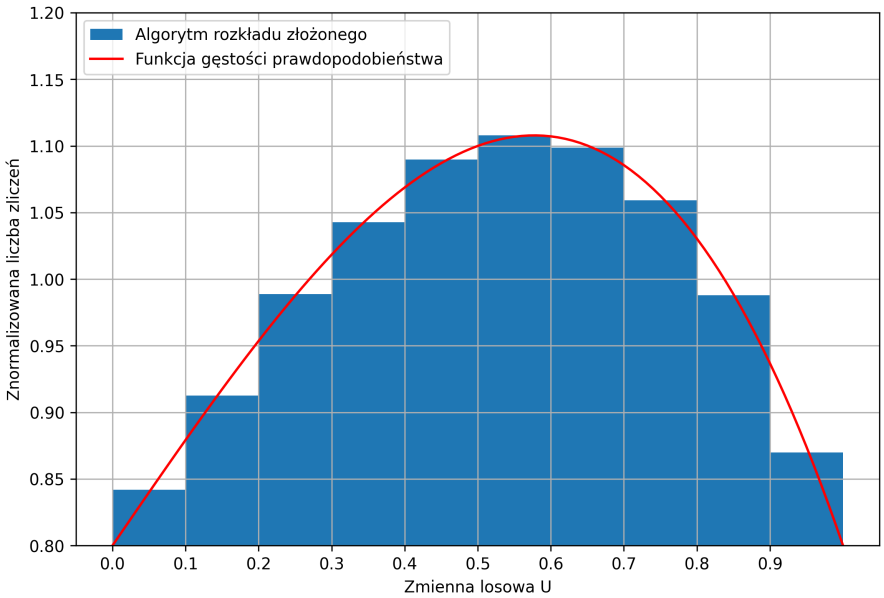
$$p_i = F((i + 1) \cdot 0.1) - F(i \cdot 0.1), \tag{11}$$

a następnie uzyskane wyniki porównano z wartościami tablicowymi.

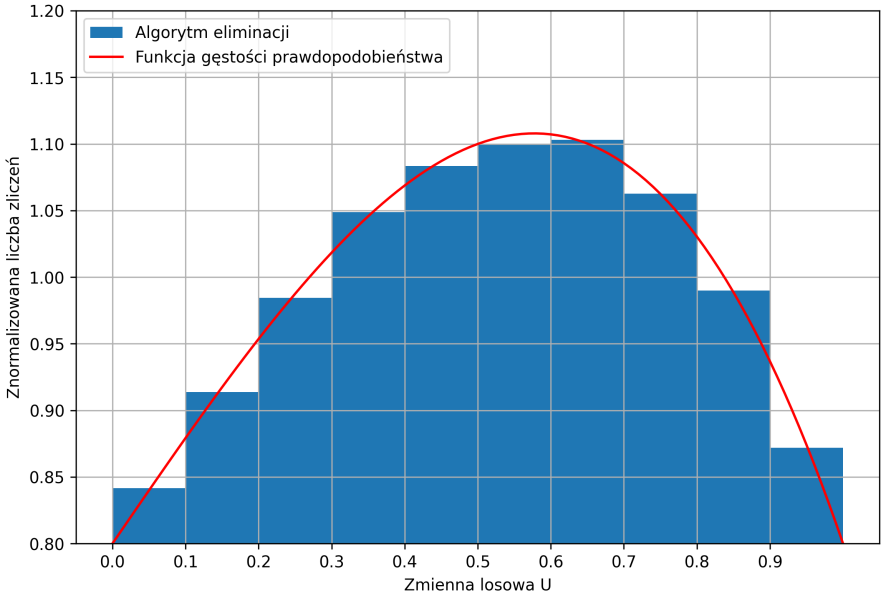
Wszystkie wykresy utworzono przy pomocy biblioteki *matplotlib* w *Python*.

3 Wyniki

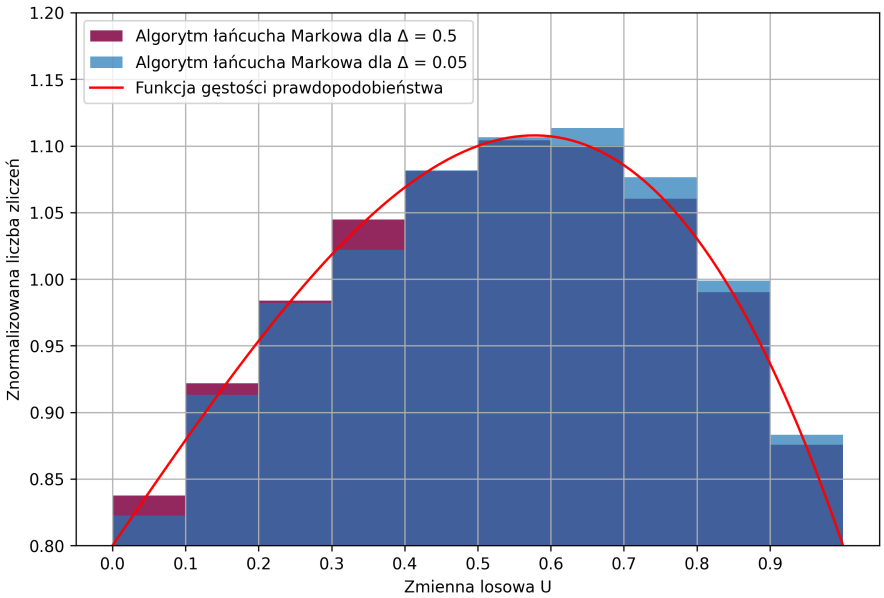
Na Rysunkach 1, 2 i 3 przedstawiono znormalizowane histogramy zliczeń występowania danej wartości U z przedziału [0,1]. Jak można zauważyć kształt każdego z histogramów pokrywa się dość dobrze z kształtem funkcji gęstości prawdopodobieństwa.



Rysunek 1: Histogram znormalizowanej ilości zliczeń zmiennej losowej U $\in$ [0, 1] dla generatora liczb losowych skonstruowanego na podstawie algorytmu rozkładu złożonego. Na wykres naniesiono czerwoną linią kształt bazowej funkcji gęstości prawdopodobieństwa.



Rysunek 2: Histogram znormalizowanej ilości zliczeń zmiennej losowej U $\in$ [0, 1] dla generatora liczb losowych skonstruowanego na podstawie algorytmu eliminacji. Na wykres naniesiono czerwoną linią kształt bazowej funkcji gęstości prawdopodobieństwa.



Rysunek 3: Zestawienie histogramów znormalizowanej ilości zliczeń zmiennej losowej U $\in$ [0, 1] dla generatora liczb losowych skonstruowanego na podstawie algorytmu łańcucha Markowa i zadanych parametrach Δ równych kolejno 0.5 oraz 0.05. Na wykres naniesiono czerwoną linią kształt bazowej funkcji gęstości prawdopodobieństwa.

Chcąc lepiej oszacować dopasowanie generatorów do funkcji gęstości prawdopodobieństwa, aniżeli jedynie graficznie, wykonano test χ^2 dla każdego wygenerowanego ciągu. Wyniki wraz z wartością teoretyczną ([2]) zestawiono w Tabeli 1.

Tabela 1: Wyniki testu χ^2 dla każdego z wygenerowanych ciągów wraz z wartością tablicową [2].

Wartości χ^2 dla 9 stopni swobody				
Wartość teoretyczna	Rozkład złożony	Metoda eliminacji	Łańcuch Markowa dla $\Delta = 0.5$	Łańcuch Markowa dla $\Delta = 0.05$
16.919	6.163	9.997	9.093	162.151

Dla rozkładu złożonego, metody eliminacji oraz łańcucha Markowa spełniającego warunek $\Delta = 0.5$, test χ^2 wypadł pozytywnie. W takim przypadku nie możemy odrzucić hipotezy H_0 mówiącej, że wygenerowane ciągi liczb pseudolosowych mają rozkład $F(x)$.

Możemy za to odrzucić hipotezę H_0 w przypadku ciągu wygenerowanego przez algorytm łańcucha Markowa dla $\Delta = 0.05$. Mimo podobieństwa rozkładu słupków histogramów (Rys. 3) między oboma ciągami liczb, widać, że ten niespełniający hipotezy przesunięty jest w stronę większych liczb. Może to oznaczać, że autokorelacja dla $\Delta = 0.05$ jest zbyt duża.

4 Wnioski

Zaprezentowane wyniki pokazały, że istnieją co najmniej trzy możliwe generatory liczb pseudolosowych pozwalające losować zmienne zgodnie z funkcją gęstości prawdopodobieństwa [6]. Test χ^2 został spełniony przez algorytmy: rozkładu złożonego (który wypadł najlepiej), łańcucha Markowa dla $\Delta = 0.5$ oraz metody eliminacji. Wyniki te pokazują, że istnieje wiele możliwości konstruowania generatorów, które będą spełniały stawiane przed nimi wymagania. Istotnym jest odpowiednie zgłębienie rozpatrywanego problemu i rozmyślne dobranie algorytmu i jego parametrów. Po mimo tego, że algorytm łańcucha Markowa dał drugi najlepszy rezultat w teście χ^2, pokazano, że zły dobór parametru Δ drastycznie zmienia jakość pracy tego generatora.

Literatura

[1] Tomasz Chwiej, dr hab. inż., wykład pt. "*Generatory liczb pseudolosowych*", AGH, Kraków, 2024.

[2] Chi-Square Distribution Table, University of Arizona, https://www.math.arizona.edu/~jwatkins/chi-square-table.pdf